## Contents

1.0 Introduction
   - Modeling and drafting
   - Modeling and rendering
   - Modeling and design
   - Keep in touch
      
1.1 The versions of form•Z and their differences
1.2 About this manual and its style
1.3 The features of form•Z
   - Modeling objects and drafting elements
   - Projects, imager sets, and windows
   - Interface
   - Modeling
   - Plotting, rendering, and animations
   - Drafting
   - Other features
1.4 form•Z and the 3D standards
1.5 Installation and memory management
   - Macintosh OS X
   - Windows
1.6 Keyboard commands
1.7 Requesting information from the system 1 - 21
1.8 The Help environment 1 - 23
   Introductory Help... 1 - 23
   Keyboard Help... 1 - 24
   Modeling Tools... 1 - 25
   Drafting Tools... 1 - 26
   Menus... 1 - 27
   Window Tools... 1 - 28
   Manual... 1 - 29
1.9 Customizing the tool bars 1 - 30
1.10 Saving your selections in the Preferences dialog 1 - 35
## Contents

### 2.0 Introduction
2 - 1

### 2.1 The graphics windows and dialogs
2 - 5
  2.1.1 Associated windows
  2 - 6
  2.1.2 Manipulating and activating windows
  2 - 7
  2.1.3 Controlling the graphics of modeling and drafting windows
  2 - 8
  2.1.4 Communications between the modeling and drafting windows
  2 - 9
  2.1.5 Dialogs
  2 - 10
  2.1.6 Preview dialogs and their viewing tools
  2 - 14

### 2.2 The window tools
2 - 16
  2.2.1 Switching the reference planes
  2 - 17
    - XY Reference Plane
    2 - 18
    - YZ Reference Plane
    2 - 18
    - ZX Reference Plane
    2 - 18
    - Arbitrary Reference Plane
    2 - 18
  2.2.2 Locking to the perpendicular direction
  2 - 19
    - Perpendicular Switch
    2 - 19
  2.2.3 Reference Planes
  2 - 20
    Defining arbitrary planes
    2 - 20
    - Define Arbitrary Plane
    2 - 20
    - Define Perpendicular Plane
    2 - 21
    Moving reference planes
    2 - 22
    - Move Plane
    2 - 22
    - Move Plane Origin
    2 - 22
2.2.4 Snapping to grid points

2.2.5 Directional snapping

2.2.6 Object snapping

Drawing in 3D space with snaps
2.2.7 Zooming and panning
- Zoom In/Out about Point (Magnifying Glass)
- Hand
- Zoom In by Frame
- Zoom Out by Frame
- Zoom In Incrementally
- Zoom Out Incrementally
- Go to Previous
- Fit All
- Reset

2.2.8 Viewing
- Set View
- Navigate View
- Walkthrough
- Match Perspective View

2.2.9 The memory box and zoom percentage menu

2.3 Palettes
2.3.1 Palette features
2.3.2 Grouping and placing entities into groups
2.3.3 Palette dialogs and their commands
2.3.4 The contextual palette menu
2.3.5 Transferring palette items

2.4 Coordinates and Prompts palettes
2.4.1 The Coordinates palette and types of coordinates
2.4.2 The Prompts palette
2.4.3 Indirect and direct numeric input
   - Indirect numeric input
   - Direct numeric input through the Prompts palette
   - Numeric input through the Coordinates palette
2.5  The Tool Options palette
   2.6  Color, surface styles, and their palettes
      2.6.1  The Colors palette
      2.6.2  The Surface Styles palette
      2.6.3  Project colors and transferring colors or surface styles
2.7  Layers and their palette
      2.7.1  The Layers palette
      2.7.2  The Layers dialog
2.8  Objects and their palette
      2.8.1  The Objects palette
      2.8.2  The Objects dialog
2.9  Views and their palette
      2.9.1  The Views palette
      2.9.2  The Views dialog
2.10  Lights and their palette
      2.10.1  The Lights palette
      2.10.2  The Lights dialog
      2.10.3  Editing the parameters of a light
      2.10.4  Editing the parameters of groups of lights
2.11  Scenes and their palette
      2.11.1  The Scenes palette
      2.11.2  The Scenes dialog
      2.11.3  The Scene Options dialog
2.12  Symbols, their palette and dialog
      2.12.1  The Symbols palette
      2.12.2  The Symbol Libraries dialog
2.13 More modeling palettes 2 -105
  2.13.1 The Planes palette and dialog 2 -106
  2.13.2 Profiles and their palette 2 -108
  2.13.3 The Status Of Objects palette 2 -109

2.14 More drafting palettes 2 -110
  2.14.1 The Line Styles and Weights palettes 2 -111
  2.14.2 The Hatch Patterns palette 2 -112
## 3.0 Introduction

3.0.1 Types of menu items
3.0.2 Overview of menu items
3.0.3 Contextual Menus
3.0.4 Dialog box conventions

## 3.1 The File menu

3.1.1 Files and naming conventions
3.1.2 Creating and opening native files
   - New [Model]
   - New [Draft]
   - New Imager Set...
   - New Script
   - Open...
   - Open Recent

3.1.3 Closing and saving native files
   - Close
   - Save
   - Save As...
   - Save A Copy As...
   - Revert to Saved

3.1.4 Importing and exporting non-native files
   - Import...
   - Import Recent
   - Export...
3.1 The File menu

3.1.5 Viewing image files

3.1.6 Printing

3.1.7 Terminating the session

3.2 The Edit menu

3.2.1 Undoing and redoing

3.2.2 Cutting, pasting, and duplicating

3.2.3 The form•Z Clipboard

3.2.4 Selecting, deselecting, and clearing
Select By...  3 - 33
Select By...  in modeling  3 - 34
Select By...  in drafting  3 - 40
Select By...  for symbols  3 - 42
Select By...  for model text  3 - 43
Select By...  for draft text  3 - 49
Deselect  3 - 52
Clear  3 - 52
Clear All Ghosted  3 - 52

3.2.5 Key shortcuts  3 - 53
3.2.6 Direct view navigation  3 - 62
Interactive  3 - 63
Keyed  3 - 64
3.2.7 Saving and recalling preferences  3 - 65
Preferences...  3 - 65
  System: General  3 - 66
  System: Language  3 - 67
  System: Interface  3 - 68
  Preview: Dialogs  3 - 68
  System: Scratch Disk  3 - 71
  System: Recent Files  3 - 72
  System: Updates  3 - 73
  Project: General  3 - 75
  Project: Auto Save  3 - 79
  Project: Fonts  3 - 81
  Project: File Search Paths  3 - 83
  Project: Undo  3 - 84
  Project: Warnings  3 - 85
  Project: Modeling: Animation  3 - 86
  Project: Modeling: Model Type  3 - 87
  Project: Modeling: Radiosity  3 - 88
  Project: Modeling: Textures  3 - 89
  Project: Drafting: General  3 - 90
  Script: General  3 - 90

Where are the preference files stored and how are they called  3 - 91
3.3 The **Window** menu

3.3.1 Opening and closing windows

- **New Model Window** 3 - 93
- **New Draft Window** 3 - 93
- **Tile Windows**
  - **Open** 3 - 94
  - **Close** 3 - 94
  - **Arrange** 3 - 94
  - **Align And Scale Views**
    - **Close** 3 - 95
    - **Close All** 3 - 95

3.3.2 Window frames

- **Window Frames** 3 - 97
- **Window Frame Options** 3 - 97
- **Align And Scale Frames** 3 - 98

3.3.3 The extended cursor

- **Extended Cursor** 3 - 99
- **Extended Cursor Options...** 3 - 99

3.3.4 Auto scrolling

3.3.5 Axes and grids

- **Show Plane Axes** 3 - 101
- **Show World Axes** 3 - 101
- **Show Grid** 3 - 101
- **Window Setup...** 3 - 101

3.3.6 Rulers

- **Show Rulers** 3 - 103

3.3.7 Snapping options

- **Snap Options...** 3 - 106

3.3.8 Importing and placing underlays

- **Underlay...** 3 - 107

3.3.9 The window list

3.4 The **Heights** menu

- **Graphic/Keyed** 3 - 111
- **20'-0" ... -20'-0"** 3 - 111
3.5 The **Views** menu

3.5.1 Setting the viewing angles

\[ z = 30^\circ \quad x = 60^\circ \quad \ldots \quad z = 60^\circ \quad x = 30^\circ \]

*Custom View Angles...*

\[ x = 30^\circ \quad y = 60^\circ \quad \ldots \quad x = 45^\circ \quad y = 45^\circ \]

*Inclination* = 30° \ldots *Inclination* = 150°

3.5.2 Selecting an orthographic projection

*Plane Projection*

3.5.3 Types of 3D views and their parameters

*Axonometric*

*Isometric*

*Oblique*

*Perspective*

*Panoramic*

3.5.4 View parameters and animations

*View Parameters...*

*Clip Hither/Yon*

*Reset Hither/Yon*

*Reset View Angle*

*Reset View Spin*

*Save View*

*Views...*

3.5.5 Setting sun positions

*Sun Position...*

Selecting and editing geographical locations

3.5.6 The cone of vision

*Edit Cone of Vision*

3.5.7 The drafting view menu

*Custom Rotation*

*View Parameters*
3.6 The Display menu

3.6.1 Setting the display scale

\[ \frac{1}{8}'' = 1'\text{–}0'' \text{ Scale} \]
\[ \sqrt{\frac{1}{16}}'' = 1'\text{–}0'' \text{ Scale} \]
\[ \frac{1}{32}'' = 1'\text{–}0'' \text{ Scale} \]
Custom Display Scale...

3.6.2 Interactive displays

Wire Frame*
Interactive Shaded*

3.6.3 Quick paint, hidden line, and surface rendering

Quick Paint*
Hidden Line*
Show Transparent Objects: As for Quick Paint*
Surface Render*

3.6.4 Shaded rendering

Shaded Render*

3.6.5 Doodle rendering

Doodle*
Effect Parameters

3.6.6 Project level rendering options

3.6.7 Lights and shadows

3.6.8 Network rendering

Network Render...
The Browse for Server dialog
The Server Settings dialog
Network Render Status...
The Job Information dialog
The form•Z Render Server and its window
The Client Information dialog
The User Settings dialog
The Server Configuration dialog
The form•Z Render Client and its window
The Client Settings dialog
The Server Settings dialog
The Browse for Server dialog
### 3.6.9 Display and rendering options

**Display Options...**

### 3.6.10 QuickTime VR movies

- Saving a cylindrical panoramic movie
- Saving a spherical panoramic movie
- Saving an object movie
- Sizing QuickTime VR movies
- Viewing QuickTime VR movies
- Compressing QuickTime VR movies

### 3.6.11 Animations, rendering memory, and showing surfaces

### 3.6.12 Setting image parameters

**Image Options...**

### 3.6.13 Plotting in drafting

**Replot**

### 3.7 The Options menu

#### 3.7.1 Project level options

- **Layers...**
- **Project Colors...**
- **Symbol Libraries...**
- **Working Units...**
- Options outside of the tabs

#### 3.7.2 Modeling options

- **Animation...**
- **Animation Manager...**
- **Lights...**
- **Macro Transformations...**
- **Objects...**
- **Profiles...**
- **Reference Planes...**
- **Status Of Objects...**
- **Surface Styles...**
- **Attributes...**
- **Information Management...**

#### 3.7.3 Drafting options
3.7.4 User defined attributes
The attributes list
The fields list
An example

3.7.5 Information management
The Records section
The Selection section
The Fields section

3.8 The Palettes menu

3.8.1 Animation palettes
Animation Editor
Animation Score
Animation Time Line

3.8.2 Modeling palettes
Coordinates
Layers
Lights
Modeling Tools
Objects
Planes
Profiles
Prompts
Scenes
Surface Styles
Symbols
Views
Window Tools

3.8.3 Drafting palettes
Colors
Coordinates
Drafting Tools
Hatch Patterns 3 - 239
Layers 3 - 239
Line Styles 3 - 240
Line Weights 3 - 240
Prompts 3 - 240
Symbols 3 - 240
Views 3 - 240
Window Tools 3 - 240
3.8.4 Dialog based palettes 3 - 241
Display Options 3 - 241
Pick Options 3 - 242
Snap Options 3 - 242
Status Of Objects 3 - 243
Tool Options 3 - 243
Window Options 3 - 243
3.8.5 Hiding the palettes and customizing the tool bars 3 - 244
Hide/Show Palettes 3 - 244
Customize Tools... 3 - 244

3.9 The Extensions menu 3 - 245
Extensions Manager... 3 - 246
Run Utility... 3 - 247
Run Recent Utility... 3 - 247
Use Script Debugger... 3 - 247

3.10 The Help menu 3 - 251
3.10.1 Accessing the Internet 3 - 252
form•Z Web Site... 3 - 252
form•Z Web Support... 3 - 252
e-mail Tech Support... 3 - 252
3.10.2 Summary information about the operations of form•Z 3 - 253
General... 3 - 253
3.10.3 Accessing the form•Z Manuals 3 - 254
Manual... 3 - 254
Mac OS form•Z Manual Viewer 3 - 255
The Tool bar 3 - 256
The Menu bar 3 - 257
The outline drawer 3 - 259
The Windows form•Z Manual Viewer 3 - 260

3.10.4 Messages and information 3 - 261
Error Messages... 3 - 261
Project Info... 3 - 261

3.11 The contextual menu 3 - 262

3.12 The form•Z Imager 3 - 264
3.12.1 The Imager set window and list 3 - 265
3.12.2 The Imager operations 3 - 268
3.12.3 The Imager menus
   The File menu 3 - 277
   The Edit menu 3 - 279
   The Windows menu 3 - 279
3.12.4 Imager network rendering 3 - 280

3.13 Importing and exporting object file formats 3 - 281
3.13.1 Common object import options 3 - 282
   Setting the units of the imported entities 3 - 283
   Importing as modeling objects 3 - 283
   Transforming files when importing as modeling objects 3 - 285
   Importing symbols 3 - 286
   Importing text 3 - 288
   Constructing solids from closed polylines 3 - 288
   Importing colors and rendering parameters 3 - 289
   Importing as drafting elements 3 - 290
3.13.2 Common object export options 3 - 291
   Notation and platform types 3 - 291
   Grouping method 3 - 292
   Export method 3 - 293
   Decomposing faces 3 - 296
   Exporting colors and rendering parameters 3 - 297
Exporting drafting data

3.13.3 Importing and exporting texture and surface style parameters
Importing textures
Exporting textures
Tips and warnings

3.13.4 3DGF
Importing 3DGF files
Exporting 3DGF files

3.13.5 3DMF
Importing 3DMF files
Exporting 3DMF files
How are textures handled

3.13.6 3DS
Importing 3DS files
Exporting 3DS files
How are textures handled

3.13.7 Art•lantis
Importing Art•lantis files
Exporting Art•lantis files

3.13.8 Collada

3.13.9 DEM

3.13.10 DXF
Importing DXF files
Exporting DXF files

3.13.11 DWG
Supported entity types
Importing DWG files
Exporting DWG files

3.13.12 FACT
Importing FACT files
Exporting FACT files
How textures are handled

3.13.13 Google Earth

3.13.14 Google SketchUp
3.13.15 IGES
  Importing IGES files  3 - 325
  Exporting IGES files  3 - 329
3.13.16 Lightscape
  Importing Lightscape files  3 - 332
  Exporting Lightscape files  3 - 332
  Exporting lights  3 - 333
  Exporting textures and surface styles  3 - 333
  Exporting symbols and parametric objects  3 - 333
  Exporting radiosity parameters  3 - 333
  Exporting system colors  3 - 333
3.13.17 LightWave
  Importing LightWave files  3 - 334
  Modeling Limitations  3 - 334
  Texture Map Limitations  3 - 334
  Exporting LightWave files  3 - 335
  Limitations  3 - 336
3.13.18 OBJ
  Importing OBJ files  3 - 337
  Exporting OBJ files  3 - 337
  How textures are handled  3 - 337
3.13.19 PLY
  Importing PLY files  3 - 338
  Exporting PLY files  3 - 338
3.13.20 RIB
  Exporting RIB files  3 - 339
  How textures are handled  3 - 339
3.13.21 SAT and ACIS data
  Importing SAT  3 - 340
  Exporting ACIS entities  3 - 341
3.13.22 STL
  Importing STL files  3 - 342
  Exporting STL files  3 - 343
  Requirements and recommendations  3 - 345
3.13.23 VRML
Importing VRML files
Exporting VRML files
How textures are handled
3.13.24 ZPR
Importing ZPR files
Exporting ZPR files

3.14 Importing and exporting image file formats
3.14.1 Common image import options
3.14.2 Common image export options
3.14.3 BMP
Importing BMP files
Exporting BMP files
3.14.4 EPS
Importing EPS files
Exporting EPS files
3.14.5 HPGL
Exporting HPGL files
3.14.6 Illustrator
Importing Illustrator files
Exporting Illustrator files
3.14.7 JPEG
Importing JPEG files
Exporting JPEG files
3.14.8 Metafile
Importing Metafile files
Exporting Metafile files
3.14.9 PICT
Importing PICT files
Exporting PICT files
3.14.10 Piranesi
Exporting Piranesi files
3.14.11 PNG
   Importing PNG files 3 - 368
   Exporting PNG files 3 - 368
3.14.12 QuickTime™ Image
   Importing QuickTime Image files 3 - 369
   Exporting QuickTime Image files 3 - 369
3.14.13 Targa
   Importing Targa files 3 - 370
   Exporting Targa files 3 - 370
3.14.14 TIFF
   Importing TIFF files 3 - 371
   Exporting TIFF files 3 - 371
4 Modeling

Contents

4.0 Introduction 4 - 1
   4.0.1 Types and representations of objects 4 - 2
   4.0.2 Parametric objects and their representations 4 - 5
   4.0.3 Local coordinate systems and centroids of objects 4 - 7
   4.0.4 The modeling tool bar 4 - 9
   4.0.5 Types of tools 4 - 11

4.1 Generating and editing primitives 4 - 18
   4.1.1 Representation and display of analytic primitives 4 - 19
   4.1.2 Drawing cubes 4 - 22
      Cube 4 - 22
   4.1.3 Drawing cones and cylinders 4 - 23
      Cone 4 - 23
      Cylinder 4 - 26
   4.1.4 Drawing spheres 4 - 27
      Sphere 4 - 27
   4.1.5 Drawing tori 4 - 30
      Torus 4 - 30
   4.1.6 Drawing paraboloids 4 - 33
      Paraboloid 4 - 33
   4.1.7 Drawing single hyperboloids 4 - 34
      Single Hyperboloid 4 - 34
4.1.8 Drawing double hyperboloids 4 - 35

**Double Hyperboloid** 4 - 35

4.1.9 Drawing hyperbolic paraboloids 4 - 36

**Hyperbolic Paraboloid** 4 - 36

4.1.10 Editing analytic primitives and quadratic surfaces 4 - 37

- Editing the controls of primitives 4 - 37
- Editing the surface of primitives 4 - 38
- Editing the parameters of primitives 4 - 39

4.1.11 Spherical objects 4 - 40

**Spherical Object** 4 - 40

- Editing spherical objects 4 - 43

4.1.12 Metaballs 4 - 44

**Metaballs** 4 - 45

4.1.13 Stars 4 - 47

**Star** 4 - 47

4.2 Direct object generation 4 - 49

4.2.1 Types of objects 4 - 50

- **2D Surface/Wire** 4 - 50
- **2D Enclosure** 4 - 52
- **3D Extrusion** 4 - 53
- **3D Converged** 4 - 55
- **3D Enclosure** 4 - 56

4.2.2 Drawing polygons 4 - 58

- **Rectangle** 4 - 58
- **Rectangle, 3 Point** 4 - 58
- **Polygon** 4 - 59

4.2.3 Drawing circles and ellipses 4 - 65
Circle, Center and Radius
Circle, Diameter
Circle, 3 Point
Ellipse, Major and Minor Radius
Ellipse, Diameter and Radius

4.2.4 Drawing points, segments, and vector lines

Point
Segment
Vector Line

4.2.5 Drawing curves

Spline, Quadratic Bezier
Spline, Cubic Bezier
B-Spline, Cubic
Sketch Spline, Quadratic Bezier
Stream Line

4.2.6 Drawing arcs

Arc, Clockwise, Endpoint-Last
Arc, Counterclockwise, Endpoint-Last
Arc, Clockwise, Center-Last
Arc, Counterclockwise, Center-Last
Arc, 3-Point, Endpoint-Last
Arc, 3-Point, Midpoint-Last

4.2.7 Drawing continuous lines

4.2.8 Undoing a step at a time when drawing lines

4.2.9 Editing shapes
Editing single point, single segment objects, vector lines, streamlines and polygons 4 - 80
Editing splines and arcs 4 - 81

4.3 Picking and editing 4 - 84

4.3.1 The topological levels 4 - 85
- Point 4 - 85
- Segment 4 - 85
- Outline 4 - 85
- Face 4 - 85
- Object 4 - 86
- Group 4 - 86
- Hole/Volume 4 - 86
- Automatic 4 - 87

4.3.2 The Pick tool and the **Pick Options** dialog 4 - 88
- Pick 4 - 88

4.3.3 Area picking and pick parade 4 - 91
- Area picking 4 - 91
- Pick parade 4 - 92

4.3.4 Prepicking and postpicking 4 - 93

4.3.5 Transforming with the Pick tool 4 - 94
- Move 4 - 94
- Rotate relative to reference plane 4 - 94
- Rotate relative to screen plane 4 - 94
- Scale 4 - 94

4.3.6 Using the nudge keys 4 - 95

4.3.7 Edit tools 4 - 97
- Edit Controls 4 - 98
- Transforming controls 4 - 99
Inserting and deleting controls of splines 4 - 101
Changing the type of a pivot point in cubic Bezier splines 4 - 101

4.4 Insertions
4.4.1 Inserting outlines
   Insert Outline 4 - 107
4.4.2 Inserting faces and volumes
   Insert Face/Volume 4 - 109
4.4.3 Inserting holes
   Insert Hole 4 - 112
4.4.4 Inserting openings
   Insert Opening 4 - 114
4.4.5 Bypassing the face preselection requirement 4 - 116

4.5 Derivative objects
Facetted versus smooth model types 4 - 118
The status of objects settings 4 - 118
4.5.1 Deriving point objects and point clouds
   Point Cloud 4 - 119
4.5.2 Derivative surface objects
   2D Surface 4 - 120
   From each selected entity 4 - 120
   From boundary of surface object 4 - 121
   From closed stitch 4 - 121
   From selected segments 4 - 122
   From selected faces 4 - 122
4.5.3 Derivative 2D enclosures
   2D Enclosure 4 - 123
4.5.4 3D extrusions, convergences, and enclosures 4 - 124
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>3D Extrusion</td>
<td>125</td>
</tr>
<tr>
<td>3.4</td>
<td>3D Converged</td>
<td>125</td>
</tr>
<tr>
<td>3.5</td>
<td>3D Enclosure</td>
<td>126</td>
</tr>
<tr>
<td></td>
<td>3D solids from other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>solid objects</td>
<td>127</td>
</tr>
<tr>
<td>4.5.5</td>
<td>Parallel objects</td>
<td>128</td>
</tr>
<tr>
<td>4.5.6</td>
<td>Frames</td>
<td>134</td>
</tr>
<tr>
<td>4.5.7</td>
<td>Projections of objects</td>
<td>135</td>
</tr>
<tr>
<td>4.5.8</td>
<td>Unfolding objects</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Unfolding facetted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>objects</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>Unfolding smooth</td>
<td></td>
</tr>
<tr>
<td></td>
<td>objects</td>
<td>141</td>
</tr>
<tr>
<td>4.6</td>
<td>Terrain models</td>
<td>145</td>
</tr>
<tr>
<td>4.6.1</td>
<td>The Terrain Model tool</td>
<td>146</td>
</tr>
<tr>
<td></td>
<td>and its dialog</td>
<td></td>
</tr>
<tr>
<td>4.6.2</td>
<td>Terrain modeling type</td>
<td>147</td>
</tr>
<tr>
<td></td>
<td>and options</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mesh models</td>
<td>147</td>
</tr>
<tr>
<td></td>
<td>Stepped models</td>
<td>148</td>
</tr>
<tr>
<td></td>
<td>Triangulated contour</td>
<td>149</td>
</tr>
<tr>
<td></td>
<td>models</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The height options</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Smoothing and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>intersection checks</td>
<td>151</td>
</tr>
<tr>
<td>4.6.3</td>
<td>Terrain modeling</td>
<td>152</td>
</tr>
<tr>
<td></td>
<td>examples</td>
<td></td>
</tr>
<tr>
<td>4.7</td>
<td>Objects of Revolution</td>
<td>156</td>
</tr>
<tr>
<td>4.7.1</td>
<td>Revolved object</td>
<td>157</td>
</tr>
<tr>
<td></td>
<td>Revolved Object</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Generating facetted</td>
<td>159</td>
</tr>
<tr>
<td></td>
<td>revolved objects</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Generating smooth</td>
<td>162</td>
</tr>
<tr>
<td></td>
<td>revolved objects</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Previewing and</td>
<td>163</td>
</tr>
<tr>
<td></td>
<td>editing revolved objects</td>
<td></td>
</tr>
</tbody>
</table>
4.7.2 Helixes

   Helix
   Solid and surface helixes
   Wire helixes about an axis
   Wire helixes along a path

4.7.3 Screws and bolts

   Screw/Bolt
   Detailed Screw/Bolt

4.7.4 Gears

   Gear
   Editing gear objects

4.8 Stairs

   4.8.1 Spiral stairs
   Spiral Stair

   4.8.2 Stairs from paths
   Stair from Path

4.9 Sweeps, skins, lofts, and caps

   4.9.1 Sweeps
   Sweep
   Axial sweep
   Two source sweep
   Two path sweep
   Boundary sweep
   Facetted versus smooth sweeps
   Previewing and editing sweep objects

   4.9.2 Draft sweeps
   Draft Sweep

   4.9.3 Skinning
   Skin
   Executing the Skin operation
Types of the Skin operation  4 - 221
Placement types  4 - 228
Point pairing  4 - 229
Sequence of selection  4 - 230
Generating smoother skinned objects  4 - 232
Facetted versus nurbz objects  4 - 233
4.9.4  Lofting  4 - 234
Loft  4 - 234
4.9.5  Skinned lofting  4 - 238
S-Loft  4 - 238
Previewing and editing s-loft objects  4 - 245
4.9.6  Deriving surfaces from boundaries  4 - 246
Cap  4 - 246

4.10  Meshes and subdivisions  4 - 249
4.10.1  Meshes  4 - 250
Mesh  4 - 250
Examples  4 - 254
4.10.2  Quadratic subdivisions  4 - 255
Q-Subz (Quadratic Subdivisions)  4 - 255
4.10.3  Triangular subdivisions  4 - 259
T-Subz  4 - 259
Previewing and editing T-Subz objects  4 - 264
4.10.4  Reducing the resolution of meshes  4 - 268
Reduce Mesh  4 - 268
4.10.5  Triangulating  4 - 273
Triangulate  4 - 273

4.11  Deformations, displacements, and morphing  4 - 275
4.11.1  Defining profiles  4 - 276
Define Profile  4 - 276
### 4.11.2 Moving meshes
- **Move Mesh**
  - Examples

### 4.11.3 Point Disturbances
- **Disturb**

### 4.11.4 Deforming objects
- **Deform**
  - The Deformations Options dialog
  - General options
  - Editing a deformation object
  - How the deformation tools can be used

### 4.11.5 Bending objects along curves
- **Bend objects along curves**
  - Previewing and editing Bend along Curve objects

### 4.11.6 Morphing
- **Morph**
  - Tips about using morphing

### 4.11.7 Image based displacements
- **Displace**
  - Previewing and positioning displacements
  - Numeric input
  - Graphic input
  - Selecting a displacement map
  - Displacement mapping methods
  - Defining the size and amplitude of a displacement
  - Increasing the resolution of the displaced object
  - Applying a displacement to individual faces
  - Examples

### 4.12 Rounding, beveling, blending, and draft angles
- **Plain Round**
  - Facetted rounding
Smooth rounding

4.12.2 Controlled rounding
  Control Round
  Facetted rounding
  Smooth rounding

4.12.3 Stitch rounding
  Stitch Round
  Facetted stitch rounding
  Smooth stitch rounding

4.12.4 Rounding cases to watch
  The Use Radius versus Use Distance options
  Rounding faceted objects with non-planar faces
  Rounding a faceted stitch with the fillet fitting method
  Applying variable radii to faceted rounding
  Facetted rounding cases to watch

4.12.5 Blending
  Blend

4.12.6 Filleting
  Fillet

4.12.7 Applying draft angles
  Draft Angle

4.13 Smooth splines and c-curves

4.13.1 Types of smooth curves
  NURBS curves
  Tangent curves
  Spline curves

4.13.2 Creating and editing curves
  Creating smooth curves
  Curve
  Editing smooth curves

4.13.3 Reconstructing curves
  Reconstruct Curve
4.13.4 Attaching curves
   • Attach Curve
4.13.5 Blending curves
   • Blend Curve
4.13.6 Merging curves
   • Merge Curve
4.13.7 Extending curves
   • Extend Curve
4.13.8 Breaking curves
   • Break Curve
4.13.9 Curved lines
   • C-Curve
   The curves
   Curve edit mode
   Examples
4.13.10 Formula curve
   • Formula Curve
   Previewing and editing formula curves

4.14 Nurbz objects and c-meshes
4.14.1 Creating nurbz objects
   • Nurbz
   Nurbz from analytic primitives
   Nurbz objects from derivative objects
4.14.2 Editing nurbz
   • Edit Controls
   Geometric transformations
   Inserting and deleting control points while editing a nurbz surface
   • Insert
   • Edit Surface
4.14.3 Operations that preserve or generate NURBS controls

4.14.4 Deriving cross sections and more nurbz
   Nurbz Cross Sections

4.14.5 Reconstructing nurbz surfaces
   Reconstruct Nurbs

4.14.6 Extracting curves
   Extract Curve

4.14.7 Attaching nurbz surfaces
   Attach Nurbs

4.14.8 Blending nurbz surfaces
   Blend Nurbs

4.14.9 Merging nurbz surfaces
   Merge Nurbs

4.14.10 Extending nurbz surfaces
   Extend Nurbs

4.14.11 Splitting nurbz surfaces
   Split Nurbs

4.14.12 Untrimming nurbz surfaces
   Untrim Nurbs

4.14.13 Creating curves on nurbz surfaces
   Curve on Nurbs

4.14.14 C-Meshes
   C-Mesh
   Creating the control net
   Generating the control curves
   Creating the c-mesh
   Types of c-mesh objects
   The c-mesh edit mode
   Examples

4.14.15 Formula surfaces
   Formula Surface
4.15 Patches
4.15.1 Deriving patches
4.15.2 Growing patches
4.15.3 Dividing patches
4.15.4 Attaching patches

Editing patches

4.16 Metaformz
How to derive and evaluate metaformz
4.16.1 Deriving metaformz from other objects
4.16.2 Evaluating metaformz
4.16.3 Editing metaformz
4.16.4 Examples

4.17 Booleans, trim, stitch, and sections
4.17.1 Boolean operations
4.17.2 Trimming, splitting, and stitching

Trim/Split
Trimming
Splitting
Trimming or splitting, stitching, and rounding
Trimming and splitting with lines
Some conditions
Trim and Stitch examples
Constructing the results
Trimming and splitting multiple objects
Using multiple cutters
4.17.3 Lines of intersection

4.17.4 Stitching

4.17.5 Sections of 3D objects

4.17.6 Contours

4.17.7 Generating cages

4.18 Joining, separating, grouping, and ungrouping

4.18.1 Joining volumes

4.18.2 Separating volumes and parts of objects

4.18.3 Grouping and ungrouping
4.19 3D text

4.19.1 Placing text

Text Place

4.19.2 Generation of text

4.19.3 Editing text and its control lines

Text Edit

4.19.4 Font files and utilities

Font files on the Macintosh
Font files on Windows
Font utilities

4.19.5 Searching and replacing text

Text Search and Replace
Replacing the entire text of all selected elements with a new string
Replacing all instances of a search string in the text of all selected elements with a new string
Changing the font, size and/or style within the text of all selected elements
Replacing all instances of a font, size, and/or style within the formatting of the text of all selected elements
Changing the formatting of all instances of a search string within the text of all the selected elements

4.20 3D modeling symbols

4.20.1 Symbol libraries and the Symbols palette

4.20.2 Managing symbol libraries

Manipulating symbol libraries
Viewing symbol libraries
Manipulating symbol definitions

4.20.3 Creating symbol definitions

Symbol Create
Attributes used by a symbol definition 4 - 575

4.20.4 Editing symbol definitions 4 - 576
   Editing the parameters of a symbol definition 4 - 576
   Editing the geometry of a symbol definition 4 - 578

4.20.5 Placing symbol instances 4 - 580
   Symbol Place 4 - 580

4.20.6 Editing symbol instances 4 - 584
   Symbol Edit 4 - 584

4.20.7 Exploding symbol instances 4 - 588
   Symbol Explode 4 - 588

4.20.8 Symbol definitions with nested symbol instances 4 - 589

4.20.9 Working with symbols 4 - 591

4.20.10 Troubleshooting 4 - 595

4.21 Editing 3D lines, attributes, and controls 4 - 597

4.21.1 Breaking a line 4 - 598
   Break Line 4 - 598

4.21.2 Closing an open line 4 - 600
   Close Line 4 - 600

4.21.3 Trimming lines 4 - 601
   Trim Lines 4 - 601

4.21.4 Connecting lines 4 - 605
   Connect Lines 4 - 605

4.21.5 Joining lines 4 - 607
   Join Lines 4 - 607

4.21.6 Filleting and beveling 4 - 610
   Fit Fillet/Bevel Lines 4 - 610

4.21.7 Inserting points and segments 4 - 612
   Insert Point 4 - 612
   Insert Segment 4 - 612

4.21.8 Adjusting topological attributes 4 - 613
   Make First Point 4 - 613
Reverse Direction 4 - 613
Make One/Two Sided 4 - 614
Set/Clear Point Marker 4 - 614
4.21.9 Converting types of objects 4 - 615
Convert 4 - 615
4.21.10 Covering and uncovering wire objects 4 - 619
Cover 4 - 619
4.21.11 Extracting Control Objects 4 - 620
Extract 4 - 620

4.22 Inquiring about objects, their attributes, and quantities 4 - 622
4.22.1 Querying objects 4 - 623
Query 4 - 623
Querying groups 4 - 623
Querying objects 4 - 624
Editing the object axes 4 - 626
Querying points 4 - 628
Querying segments 4 - 628
Querying outlines 4 - 628
Querying faces 4 - 629
4.22.2 Using Query to change parametric objects 4 - 630
Querying arcs, circles, ellipses, splines, and composite curves 4 - 630
Querying c-curves 4 - 630
Querying c-meshes 4 - 631
Querying primitives 4 - 633
4.22.3 Inquiring about attributes and parameters of objects 4 - 635
Query Attributes 4 - 635
Query Parameters 4 - 636
4.22.4 Measuring lengths and distances 4 - 637
Measure 4 - 637
Lengths of shapes 4 - 637
Distances between two points measured interactively 4 - 638
Distances between points and segments 4 - 638
Distances between points and surfaces 4 - 639
Distances between points and the reference plane 4 - 639
Distances between surfaces 4 - 639

4.22.5 Diagnosing and correcting problem conditions in an object 4 - 640

Object Doctor 4 - 640
The General tab 4 - 641
The Facetted tab 4 - 642
The Smooth tab 4 - 644
Actions to be taken 4 - 645

4.22.6 Diagnosing and correcting problem conditions in an project 4 - 647

Project Doctor 4 - 647

4.22.7 Identifying potential problems before exporting to 3D printing 4 - 648

Print Prep 4 - 648

4.23 Geometric transformations 4 - 651

4.23.1 The Self/Copy modifiers 4 - 652

Self 4 - 652
One Copy 4 - 652
Continuous Copy 4 - 653
Repeat Copy 4 - 653
Multi-Copy 4 - 654

4.23.2 The Transformation tools 4 - 655

Translate (Move) 4 - 655
The Smooth Topology Manipulation options 4 - 658

Rotate 4 - 660
Independent Scale 4 - 662
Uniform Scale 4 - 664
Mirror (Reflection) 4 - 665
4 Modeling

4.23 Transform

4.23.3 Transforming and copying holes

4.23.4 Macro transformations

- The Macro Transformations dialog
- Defining macro transformations through the keyboard
- Defining macro transformations graphically

Define Macro Transformation

Executing a macro transformation

- Macro Transformation 1
- Macro Transformation 2
- Macro Transformation 3

Using macros to define other macros

4.23.5 Cloning

- Creating clones
- Transforming clones and tracks
- Working with clones

4.24 Relative geometric transformations

4.24.1 Attachments

- Attach
  - Attaching entities of the same level
  - Point-to-point attachments
  - Segment-to-segment attachments
  - Face-to-face attachments
  - Object-to-object attachments
  - The justification options
  - Attaching to all faces
  - Attaching to the reference plane

4.24.2 Aligning and distributing objects

- Align/Distribute

- Macro Transformation 1
- Macro Transformation 2
- Macro Transformation 3

Using macros to define other macros

- Attaching to all faces
- Attaching to the reference plane
4.24.3 Extending segments

**Extend**

4 - 704

4.24.4 Placing shapes and objects

**Place**

Alignment of the source shape and snapping to points
Placing copies and multiple copies
Orienting the source shape when placing it
Adjusting the source shape when placed on points
Generating a nurbz object after placement
Applying a macro transformation when placing
Examples

4.24.5 Replace

**Replace**

4 - 716

4.25 Assigning and inquiring about attributes

4.25.1 Assigning colors

**Color**

4 - 718

4.25.2 Assigning smooth shade to surfaces

**Smooth Shade**

4 - 719

4.25.3 Assigning shadow attributes and display types

**Render Attributes**

4 - 720

4.25.4 Rendering tools

**Texture Map**

4 - 723

**Decal**

4 - 723

**Render Textures**

4 - 723

**Radiosity Attributes**

4 - 723

**Radiosity Bounding Box**

4 - 723

4.25.5 Changing all the attributes of objects

**Set Attributes**

4 - 724

**Get Attributes**

4 - 724
4.25.6 Copying attributes from one object to other objects 4 - 725

Copy Attributes 4 - 725

4.25.7 Ghosting and unghosting objects 4 - 727

Ghost 4 - 727

Unghost 4 - 727

4.25.8 Placing objects on layers 4 - 728

Set Layer 4 - 728

4.26 Delete operations 4 - 729

4.26.1 Deleting objects, groups, and holes 4 - 730

Delete 4 - 730

4.26.2 Deleting topological entities 4 - 731

Delete Topology 4 - 731

Deleting topology of points 4 - 731
Deleting the topology of segments 4 - 731
Deleting the topology of outlines 4 - 732
Deleting the topology of faces 4 - 732

4.26.3 Deleting geometric entities 4 - 733

Delete Geometry 4 - 733

Deleting the geometry of points 4 - 733
Deleting the geometry of segments 4 - 734
Deleting the geometry of outlines 4 - 735
Deleting the geometry of faces 4 - 735

4.26.4 Applying multiple topology or geometry deletions 4 - 736
# Drafting

## Contents

5.0 Introduction  
5.0.1 Drafting elements and their representation  
5.0.2 The drafting tool bar  
5.0.3 Types of drafting tools  

5.1 Generating delineators  
5.1.1 Types of drafting elements  
5.1.2 Drawing polygonal polylines  
5.1.3 Drawing arcs  

- Shape  
- Enclosure  
- Rectangle  
- 3 Point Rectangle  
- Polygon  
- Circle, Center and Radius  
- Circle, Diameter  
- Circle, 3 Point  
- Ellipse, Major and Minor Radius  
- Ellipse, Diameter and Radius  
- Arc, Clockwise, Endpoint-Last  
- Arc, Counterclockwise, Endpoint-Last
5.1.4 Drawing points and segments
- Draw Point
- Draw Segment

5.1.5 Drawing polylines and splines
- Polyline
- Spline, Quadratic Bezier
- Spline, Cubic Bezier
- B-Spline, Cubic
- Spline Sketch
- Polystream
- Polyarc, Clockwise, Endpoint-Last
- Polyarc, Counterclockwise, Endpoint-Last
- Polyarc, Clockwise, Center-Last
- Polyarc, Counterclockwise, Center-Last
- Polyarc, 3-Point, Endpoint-Last
- Polyarc, 3-Point, Midpoint-Last

5.2 Drawing the area and panes
- Generating and using the area
  - Area
- Generating and using panes
  - Pane

5.3 Picking
- The topological levels
Point 5 - 27
Segment 5 - 27
Element 5 - 28
Compound 5 - 28
Automatic 5 - 29
Area 5 - 29
5.3.2 The Pick tool and Pick Options dialog 5 - 30
Pick 5 - 30
Area picking 5 - 30
Unpicking 5 - 31
Pick parade 5 - 31
Prepicking and postpicking 5 - 31
5.3.3 The area and its picking and clipping functions 5 - 32
5.3.4 The Pick Pane tool 5 - 35
Pick Pane 5 - 35
5.4 Derivative enclosures and parallel elements 5 - 36
5.4.1 Derivative enclosures 5 - 37
Enclosure 5 - 37
5.4.2 Parallel Offset 5 - 38
Parallel Offset 5 - 38
5.5 Boolean and compound operations 5 - 41
5.5.1 Boolean operations 5 - 42
Union 5 - 42
Intersection 5 - 42
Difference 5 - 42
Boolean Split 5 - 42
5.5.2 Compound operation 5 - 43
Compound 5 - 43
5.6 Line editing 5 - 45
5.6.1 Breaking lines 5 - 46
5.6.2 Trimming segments

- Trim Open
- Trim Join
- Trim Fit Fillet
- Trim Bevel

5.6.3 Trimming lines

- Trim Lines
- Trim with Line

5.6.4 Joining

- Join Lines

5.6.5 Closing Open Lines

- Close lines

5.6.6 Connecting segments and lines

- Connect Segments
- Connect Lines

5.6.7 Filleting and beveling

- Fit Fillet
- Bevel

5.6.8 Inserting points

- Insert Point

5.7 Text

5.7.1 Generating and editing text

- Text Place
- Text Edit

5.7.2 Searching and replacing text

- Text Search and Replace

Replacing the entire text of all selected elements with a new string
Replacing all instances of a search string in the text of all selected elements
with a new string 5 - 65
Changing the font, size, and/or style within the text of all selected elements 5 - 65
Replacing all instances of a font, size, and/or style within the formatting of
the text of all selected elements 5 - 65
5.7.3 Displaying text 5 - 67

5.8 2D drafting symbols 5 - 69
5.8.1 Symbol libraries, their palette, and dialog 5 - 70
5.8.2 Creating and editing symbol definitions 5 - 71

Symbol Create 5 - 71
5.8.3 Placing symbol instances 5 - 72
Symbol Place 5 - 72
5.8.4 Editing symbol instances 5 - 73
Symbol Edit 5 - 73

5.9 Exploding drafting elements 5 - 75
Explode 5 - 76

5.10 Dimensions and leader lines 5 - 77
5.10.1 Linear dimensions 5 - 79
Dimension Horizontal/Vertical 5 - 80
Dimension Parallel 5 - 83
5.10.2 Angular dimensions 5 - 85
Dimension Angular 5 - 85
5.10.3 Radius and diameter dimensions 5 - 87
Dimension Radius 5 - 87
Dimension Diameter 5 - 87
5.10.4 Arc dimensions 5 - 91
Dimension Arc 5 - 91
5.10.5 Leader lines 5 - 92
Leader Line 5 - 92
5.10.6 The dimension and leader line attributes
- The Witness Lines tab
- The Terminators tab
- The Dimension Text tab

5.10.7 Changing the attributes of dimensions and leader lines
- Get Dimension Attributes
- Set Dimension Attributes

5.10.8 Moving dimensions
- Reposition Dimension

5.11 Image elements

5.12 Querying and measuring
- 5.12.1 Querying drafting elements
- Query
- 5.12.2 Querying image elements
- 5.12.3 Querying element attributes
- Query Attributes
- 5.12.4 Measuring lengths and distances
- Measure

5.13 Geometric transformations
- 5.13.1 The Self/Copy modifiers
- Self
- One Copy
- Continuous Copy
- Repeat Copy
- Multi-Copy
- 5.13.2 The geometric transformation tools
- Translate (Move)
- Rotate
Independent Scale 5 - 113
Uniform Scale 5 - 113
Reflect (Mirror) 5 - 113

5.13.3 Transforming polylines 5 - 114
5.13.4 Transforming rectangles 5 - 115
5.13.5 Transforming arcs 5 - 117
5.13.6 Transforming text 5 - 118
5.13.7 Transforming dimensions 5 - 119
5.13.8 Transforming leader lines 5 - 121
5.13.9 Transforming hatch patterns 5 - 122
5.13.10 Transforming symbols and image elements 5 - 123

5.14 Relative geometric transformations 5 - 124
5.14.1 Alignment and distribution 5 - 125
Align/Distribute 5 - 125
5.14.2 Extending lines and arcs 5 - 126
Extend Line/Arc 5 - 126
5.14.3 Placing elements on elements 5 - 128
Place 5 - 128

5.15 Element attributes 5 - 130
5.15.1 Setting and assigning line styles 5 - 131
Line Style 5 - 131
5.15.2 Setting and assigning line weights 5 - 135
Line Weight 5 - 135
5.15.3 Setting the color and all attributes 5 - 137
Color 5 - 137
Set All Attributes 5 - 137
5.15.4 Ghosting/unghosting and setting layers 5 - 138
Ghost 5 - 138
Unghost 5 - 138
5.16 Hatching

5.16.1 Selecting the current hatching
5.16.2 Changing the parameters of the vector patterns
5.16.3 Applying hatching

5.17 Delete

5.17.1 Deleting points
5.17.2 Deleting segments
5.17.3 Deleting elements
5.17.4 Deleting with the area

5.18 Importing image elements and modeling views

5.18.1 Importing image elements
5.18.2 Importing modeling views
5.18.3 Using modeling operations for drafting purposes
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>6.0 Introduction</strong></td>
<td>6 - 1</td>
</tr>
<tr>
<td>6.0.1 The rendering levels and their rendering effects</td>
<td>6 - 2</td>
</tr>
<tr>
<td>6.0.2 How the rendering effects are created</td>
<td>6 - 3</td>
</tr>
<tr>
<td>6.0.3 Multiprocessor support</td>
<td>6 - 8</td>
</tr>
<tr>
<td><strong>6.1 Producing a rendered image</strong></td>
<td>6 - 9</td>
</tr>
<tr>
<td>6.1.1 Selecting rendering type and parameters</td>
<td>6 - 10</td>
</tr>
<tr>
<td>Rendering type</td>
<td>6 - 10</td>
</tr>
<tr>
<td>Setting the raytrace parameters</td>
<td>6 - 10</td>
</tr>
<tr>
<td>Setting the image size</td>
<td>6 - 12</td>
</tr>
<tr>
<td>Pre-estimating the memory required to render an image</td>
<td>6 - 12</td>
</tr>
<tr>
<td>Setting project level rendering options</td>
<td>6 - 13</td>
</tr>
<tr>
<td>Saving a rendered image</td>
<td>6 - 14</td>
</tr>
<tr>
<td>Applying the settings to all windows</td>
<td>6 - 14</td>
</tr>
<tr>
<td>6.1.2 The <strong>Shading tab</strong></td>
<td>6 - 15</td>
</tr>
<tr>
<td><strong>Shadows</strong></td>
<td>6 - 15</td>
</tr>
<tr>
<td><strong>Texture Maps</strong></td>
<td>6 - 16</td>
</tr>
<tr>
<td><strong>Reflections</strong></td>
<td>6 - 16</td>
</tr>
<tr>
<td><strong>Transparencies</strong></td>
<td>6 - 16</td>
</tr>
<tr>
<td><strong>Bump Mapping</strong></td>
<td>6 - 17</td>
</tr>
<tr>
<td><strong>Antialiasing</strong></td>
<td>6 - 17</td>
</tr>
<tr>
<td><strong>Super Sampling</strong></td>
<td>6 - 17</td>
</tr>
<tr>
<td>6.1.3 Illumination</td>
<td>6 - 18</td>
</tr>
<tr>
<td>6.1.4 Global illumination and its tab</td>
<td>6 - 22</td>
</tr>
<tr>
<td>What is global illumination</td>
<td>6 - 22</td>
</tr>
<tr>
<td>The <strong>Global Illumination</strong> tab</td>
<td>6 - 23</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.1.5 Ambient occlusion and its tab</td>
<td>25</td>
</tr>
<tr>
<td>The <strong>Ambient Occlusion</strong> tab</td>
<td>26</td>
</tr>
<tr>
<td>Detailed parameters</td>
<td>27</td>
</tr>
<tr>
<td>How to use ambient occlusion</td>
<td>28</td>
</tr>
<tr>
<td>Controlling ambient occlusion on individual surfaces</td>
<td>31</td>
</tr>
<tr>
<td>Ambient occlusion in abstract renderings</td>
<td>33</td>
</tr>
<tr>
<td>Performance and limitations</td>
<td>33</td>
</tr>
<tr>
<td>Tutorial</td>
<td>34</td>
</tr>
<tr>
<td>6.1.6 Final gather and its tab</td>
<td>37</td>
</tr>
<tr>
<td>When to use final gather</td>
<td>37</td>
</tr>
<tr>
<td>The <strong>Final Gather</strong> tab</td>
<td>39</td>
</tr>
<tr>
<td>Improving final gather by using dome lights</td>
<td>42</td>
</tr>
<tr>
<td>Improving final gather by using radiosity</td>
<td>44</td>
</tr>
<tr>
<td>Specular effects with final gather</td>
<td>47</td>
</tr>
<tr>
<td>Quality settings</td>
<td>49</td>
</tr>
<tr>
<td>Detailed parameters</td>
<td>49</td>
</tr>
<tr>
<td>Global Illumination Tutorial</td>
<td>53</td>
</tr>
<tr>
<td>6.1.7 The <strong>Geometry</strong> tab</td>
<td>60</td>
</tr>
<tr>
<td>Decomposing non planar faces</td>
<td>60</td>
</tr>
<tr>
<td>Rendering wire frame objects</td>
<td>61</td>
</tr>
<tr>
<td>6.1.8 The <strong>Scene</strong> tab</td>
<td>63</td>
</tr>
<tr>
<td>Background</td>
<td>64</td>
</tr>
<tr>
<td>Setting the environment</td>
<td>72</td>
</tr>
<tr>
<td>6.1.9 Depth effects</td>
<td>82</td>
</tr>
<tr>
<td><strong>Depth Cue</strong></td>
<td>82</td>
</tr>
<tr>
<td><strong>Fog</strong></td>
<td>84</td>
</tr>
<tr>
<td><strong>Rain</strong></td>
<td>85</td>
</tr>
<tr>
<td><strong>Snow</strong></td>
<td>85</td>
</tr>
<tr>
<td><strong>Snow 3D</strong></td>
<td>85</td>
</tr>
<tr>
<td>6.1.10 Postprocessed rendering effects</td>
<td>87</td>
</tr>
<tr>
<td>Applying the Alpha Channel</td>
<td>88</td>
</tr>
<tr>
<td>Enabling postprocessing</td>
<td>88</td>
</tr>
<tr>
<td>Postprocessing after a rendering is completed</td>
<td>89</td>
</tr>
<tr>
<td>Postprocessing the background</td>
<td>90</td>
</tr>
<tr>
<td>Adjusting the image exposure</td>
<td>91</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.1.11 Loading and manipulating image maps</td>
<td>6 - 102</td>
</tr>
<tr>
<td>6.2 Surface styles</td>
<td>6 - 104</td>
</tr>
<tr>
<td>6.2.1 The Surface Styles palette and dialog</td>
<td>6 - 105</td>
</tr>
<tr>
<td>6.2.2 Editing the parameters of a surface style</td>
<td>6 - 107</td>
</tr>
<tr>
<td>The Simple tab</td>
<td>6 - 108</td>
</tr>
<tr>
<td>Types of shaders</td>
<td>6 - 108</td>
</tr>
<tr>
<td>Previewing the effects of the shaders</td>
<td>6 - 109</td>
</tr>
<tr>
<td>How procedural textures are mapped</td>
<td>6 - 110</td>
</tr>
<tr>
<td>How surface styles are rendered by display types other than RenderZone</td>
<td>6 - 111</td>
</tr>
<tr>
<td>The shaders dialogs</td>
<td>6 - 112</td>
</tr>
<tr>
<td>6.2.3 Color shaders</td>
<td>6 - 115</td>
</tr>
<tr>
<td>Plain color shader</td>
<td>6 - 115</td>
</tr>
<tr>
<td>The background color shader</td>
<td>6 - 116</td>
</tr>
<tr>
<td>Streaks</td>
<td>6 - 117</td>
</tr>
<tr>
<td>Surface Evaluation</td>
<td>6 - 118</td>
</tr>
<tr>
<td>Colors based on wrapped procedural textures</td>
<td>6 - 120</td>
</tr>
<tr>
<td>Colors based on solid procedural textures</td>
<td>6 - 137</td>
</tr>
<tr>
<td>Color image maps</td>
<td>6 - 145</td>
</tr>
<tr>
<td>6.2.4 Reflection/transmission shaders</td>
<td>6 - 148</td>
</tr>
<tr>
<td>Non-reflective shaders</td>
<td>6 - 151</td>
</tr>
<tr>
<td>Ray traced reflections</td>
<td>6 - 156</td>
</tr>
<tr>
<td>Simulated reflections</td>
<td>6 - 165</td>
</tr>
<tr>
<td>Applying reflection and transmission effects with textures</td>
<td>6 - 167</td>
</tr>
<tr>
<td>6.2.5 Transparency shaders</td>
<td>6 - 169</td>
</tr>
<tr>
<td>Independently set transparencies</td>
<td>6 - 171</td>
</tr>
<tr>
<td>Transparency shaders matching the color textures</td>
<td>6 - 175</td>
</tr>
<tr>
<td>Transparencies from images</td>
<td>6 - 176</td>
</tr>
<tr>
<td>6.2.6 Bump shaders</td>
<td>6 - 178</td>
</tr>
<tr>
<td>Bumps based on wrapped procedural textures</td>
<td>6 - 179</td>
</tr>
<tr>
<td>Bumps based on solid procedural textures</td>
<td>6 - 184</td>
</tr>
<tr>
<td>Bumps matching the color textures</td>
<td>6 - 185</td>
</tr>
<tr>
<td>Bumps from images</td>
<td>6 - 186</td>
</tr>
</tbody>
</table>
6.2.7 Copying surface styles and invoking predefined surface styles 6 - 187
6.2.8 LWA predefined materials 6 - 189
   Legal limits of LWA files 6 - 190
   Textures in LWA materials 6 - 191
   LWA files in the **Predefined Materials** dialog 6 - 191
   Saving form•Z files with shaders that have read only or secret parameters 6 - 192
6.2.9 Rendering trees 6 - 193

6.3 Texture map control 6 - 195
6.3.1 The **Default Texture Map Control** tab 6 - 196
6.3.2 The Texture Map tool 6 - 198

   ![Texture Map] 6 - 198

6.3.3 Previewing and positioning textures 6 - 200
   Numeric input 6 - 202
   Graphic input 6 - 203
6.3.4 Texture mapping methods 6 - 204
6.3.5 Defining the size of textures 6 - 209
6.3.6 Transforming textures with objects 6 - 211
6.3.7 Texture groups 6 - 215
   Assigning and editing surface styles 6 - 218
6.3.8 Mapping solid textures 6 - 219
6.3.9 Examples 6 - 222
   Simple mapping of textures 6 - 222
   Using texture groups 6 - 223
   Textures which follow curved surfaces 6 - 225

6.4 Decals 6 - 226
6.4.1 The Decal tool 6 - 227

   ![Decal] 6 - 227

6.4.2 Creating decals 6 - 229
6.4.3 Determining the faces to be affected by a decal 6 - 231
6.4.4 Examples 6 - 232
   Creating a bottle with labels 6 - 232
   Surfaces with partial reflections 6 - 234
   Mixing bumps 6 - 235
6.5 Render textures
   6.5.1 The Render Textures tool

6.6 Lights
   6.6.1 Measuring light
   6.6.2 The Lights palette and dialogs
      Options outside the tabs
   6.6.3 Distant, cone, point, and projector lights
      Distant lights
      Point lights
      Cone lights
      Projector lights
   6.6.4 Making lights glow
   6.6.5 Area and line lights
      Creating an area or line light
   6.6.6 Custom lights
      The custom light representation
      Creating a custom light and setting its parameters
      Setting the intensities of a custom light and their distribution
   6.6.7 Environment lights
      Creating an environment light and setting its parameters
      Setting up a scene with environments
   6.6.8 Specifying the correct light intensity

6.7 form-Z RenderZone Plus tutorial examples
   6.7.1 Bulb
   6.7.2 Landscape
   6.7.3 Bottles
   6.7.4 Magnifying Glass

6.8 Rendering with radiosity
   What is radiosity and how does it work
   Reflections in radiosity
   Usage of radiosity: final gather support, hybrid and pure
6.8.1 The radiosity process and commands

Radiosity Options...

Initialize Radiosity*/Enter Radiosity*

Generate Radiosity Solution*

Exit Radiosity*

6.8.2 The radiosity preferences

6.8.3 The common radiosity parameters

Final Gather Support

Render Direct Illumination

Pure Radiosity

The Common Options tab parameters

The Options tab for Final Gather Support

The Options tab for Render Direct Illumination

The Options tab for Pure Radiosity

The Termination tab

The Preview tab

The Statistics tab

6.8.4 Setting the radiosity attributes

Radiosity Attributes

6.8.5 Specifying the extent of a radiosity rendering

Radiosity Bounding Box

6.8.6 Lights in radiosity

6.8.7 Surface styles in radiosity based renderings

6.8.8 Radiosity in form•Z Imager

6.9 Radiosity tips

6.9.1 How to improve the radiosity solution

The radiosity solution takes too long

Crisp shadows by manual meshing

Crisp shadows by proper choice of light type

Crisp shadows through direct illumination

6.9.2 Troubleshooting

Shadow or light leaks

The radiosity mesh becomes too dense
The image appears over or underexposed 6 - 339
The radiosity based image does not show any effects from bump mapping 6 - 340
The progress bar does not advance during the execution of the radiosity solution 6 - 340
The rendering based on a radiosity solution is all black 6 - 340

6.10 Radiosity tutorials 6 - 341

6.10.1 A radiosity solution using sun light 6 - 342
6.10.2 Simulating atmospheric light 6 - 346
6.10.3 A radiosity solution using interior lighting 6 - 348

Appendix A - 1
# Contents

7.0 Introduction
- 7.0.1 The animation process
  - Keyframe
  - Animate Along Path
  - Animate Entities
  - Animating through the Animation Score palette
- 7.0.2 Animating parameters
  - Numeric control of animated information
- 7.0.3 Fine tuning, rendering, viewing, and exporting animations
  - Rendering an animation
  - Viewing an animation
  - Exporting an animation
- 7.0.4 The animation preferences

7.1 The animation dialogs and palettes
- 7.1.1 Animation options
- 7.1.2 The Animation Time Line palette
- 7.1.3 The Animation Score palette
- 7.1.4 The **Animation Manager** dialog
  - Add tracks
  - Add controllers
- 7.1.5 Animation Controller functions
  - The Bezier controller
  - The Sine controller
  - The Noise controller
7.1.6  The Animation Editor palette 7 - 30
  Graph 7 - 31
  Editing control points and curves 7 - 32
  Cutting, copying, and pasting control points 7 - 33
  Zooming, fitting, and panning the graph 7 - 35
  Aligning control points 7 - 36
  The Key section 7 - 37
  The Track section 7 - 39

7.2  The animation tools 7 - 43
  7.2.1  The Keyframe tool 7 - 44
    Keyframe 7 - 44
    Animating points, segments, outlines, and faces 7 - 47
  7.2.2  The Animate Along Path tool 7 - 48
    Animate Along Path 7 - 48
  7.2.3  The Animate Entities tool 7 - 54
    Animate Entities 7 - 54
  7.2.4  Animation Group tool 7 - 58
    Animation Group 7 - 58
    Creating and querying animation groups 7 - 60
  7.2.5  Animation Deformation 7 - 62
    Animation Deformation 7 - 64
  7.2.6  The Extract Animation tool 7 - 68
    Extract Animation 7 - 68
  7.2.7  The Reverse Animation tool 7 - 70
    Reverse Animation 7 - 70
  7.2.8  Replace Animation 7 - 71
    Replace Animation 7 - 71
    Replacing between different types 7 - 74
  7.2.9  Using common modeling tools on animated entities 7 - 75
    Operating on an animated entity 7 - 75
    Operating on keyframes 7 - 76
    The range of the Transform tool 7 - 77
7.3 Animating color

7.3.1 Animating surface styles
- Animating surface style colors
- Animating the frame time parameter of map based shaders
- Animating numeric shader parameters
- Preview icons of animated surface styles

7.3.2 Animated color options

7.3.3 Animated color example

7.4 Generating and saving animations

7.4.1 Generating animations

7.4.2 Exporting animations

7.5 Tutorials

7.5.1 Swinging door tutorial
- Animation Editor palette
- Adding keyframes

7.5.2 Bouncing ball tutorial
- Advanced motion: squash and stretch

7.5.3 Animating surface styles
- Animating the color of the ball
- Animating the color of the floor
- Rendering the animation

7.5.4 Animating lights and cameras
- Animating the camera
- Animating the cone light

7.5.5 Animating parts of objects
- Keyframing points
- Animating point along a path

7.5.6 Animating deformations or dancing board

7.5.7 A movie within a movie
1.0 Introduction

WELCOME to the world of form•Z. If your decision to purchase the program implies that you already know where it is positioned in the realm of 3D computer graphics and modeling, you may proceed to the next section. However, you may find the contents of this section helpful in determining the ways in which form•Z is expected to contribute to your 3D modeling and design endeavors.

form•Z is a general purpose solid and surface modeler with an extensive set of 2D/3D form manipulating and sculpting capabilities, many of which are unique. form•Z is above all a modeling system that is intended to facilitate and enhance physical design. It also contains a drafting module, animation, and rendering capabilities. However, in the regular version of form•Z, the latter are intended to support the modeling and design tasks rather than transforming form•Z into a drafting, animation, or rendering program. form•Z RenderZone Plus, the version of the program that includes photorealistic rendering radiosity, final gather, and ambient occlusion, is a complete renderer as well as modeler.

form•Z is an effective design tool for architects, landscape architects, urban designers, product and interior designers, engineers, illustrators, animators, and all the design fields that deal with the articulation of three dimensional spaces and forms. The system is particularly responsive to the needs of the conceptual and preliminary stages of design, as it is able to treat the elements of composition as “soft” and easily changeable entities. However, it also offers all the tools required for the production of extensively detailed 3D models and 2D drawings. The developers of form•Z aimed at a true 3D form synthesizer that is responsive to the advanced needs of a mature designer and, at the same time, allows a novice to use it with ease. The result has been a system that offers a fresh approach and helps revise outdated perceptions of CAD and 3D modeling.
Modeling and drafting

Even though **form•Z** also contains a drafting module, the system has been designed in such a manner that modeling is the “king” and drafting is the “servant.” You may find this a radical deviation from other systems currently on the market.

The design philosophy of **form•Z** is that 3D entities and configurations are best created and visualized directly in 3D, rather than through their 2D representational conventions. The system offers extensive tools that facilitate and enhance the generation and articulation of 3D models directly in their 3D world environment. Once a design has been modeled, it may need to be communicated for presentation or construction purposes. Certain aspects of a design are best communicated through three dimensional images. Others require two dimensional representations. For instance, the construction industry still relies on the two dimensional conventions of construction drawings that have been perfected over many centuries.

In response to that need, **form•Z** offers its drafting tools as an integral part of the complete system. Without exiting the system, images of 3D entities can be transported into the drafting module, to be complemented with the drafting representational conventions, which are typically expected by the construction industry. Research has shown that the drafting conventions are not directly extractable from the 3D models. This is partly because of the representational variations used in different fields, and partly because of the restrictions it would impose on the freedom for individualistic expression, as frequently reflected in drafted drawings.

**form•Z** offers all the tools necessary to extract accurate reference drawings from the 3D models. These drawings may be plans, elevations, sections, or 3D axonometric, isometric, oblique, perspective, or panoramic views. In the drafting module, these drawings can be refined by articulating the weights and the styles of the lines, by dimensioning and hatching them in an associative manner, by annotating them, and, in general, by adding whatever symbolic representations are practiced by the different design professions. You can also use a special layout space to finalize your drafting presentation.

But the system is not restricted to a one way path. The reverse is also possible. Drawings created in the drafting module can be transported into the modeling environment, to become the basis for generating 3D models. Actually, **form•Z** can be used exclusively as a 3D modeling system, a drafting system, or both modules can be fully integrated. In the latter case, the freedom to move back and forth between the two environments is completely unrestricted. We have made the assumption that it is natural to start 3D design in 3D space, but we have no intention to tell you how to design. Consequently, **form•Z** allows you to work the way you prefer and no explicit or implicit recommendation is made that you change your style of design.
Modeling and rendering

The regular version of form\-Z offers many methods for displaying images which include z-buffer renderings with optional smooth shaded surfaces, shadows, antialiasing, and transparency. However, the rendering capabilities of the program are intended to support the modeling operations. Models need to be viewed and inspected as they are created. It is this need that the rendering facilities of form\-Z intend to address. Quite a few form\-Z users find this level of rendering sufficient for their final images. Others use form\-Z RenderZone Plus that offer all the advanced rendering and lighting effects that may be desired. These enhanced versions of form\-Z allow you to complete your rendering within the same program, without having to export anything. However, the latter remains an option and, with its support of over twenty file formats, form\-Z complements many of the rendering and animation programs whose modeling capabilities are limited.

Modeling and design

form\-Z is above all a design tool. Modeling a known and existing 3D form is one type of task, while using modeling to discover and create 3D forms that may not yet exist is different. Illustrators and animators may be primarily interested in the first type of use, while architects, product designers, and mechanical engineers frequently have new design problems to solve. In these cases, designers require a modeler that allows them to treat objects as soft entities that are easy to manipulate and change. When the latter capabilities are available, then a 3D modeler is also a true design tool.

form\-Z allows you to interactively and dynamically create and sculpt 3D forms in a variety of versatile ways, which are frequently thought provoking. As it combines many parametric and polygonal personalities, and solid with surface modeling tools within a single work environment, the variety of forms that can be created is virtually unlimited. You can use it to model known shapes and forms, or you can use it in a manner that unfolds compositional possibilities that lead to discoveries of new forms and virtual realities.
Keep in touch

The system you are about to start using has quite a few features and operations that, most likely, you have not experienced before. In addition, you will find some of the features that are commonly available to be uncommonly powerful. In producing this manual, auto-des-sys Inc. resisted the temptation to follow the rather common practice of “bragging” about each detail that is believed to be original or superior. We are confident that our readers and users will have no difficulty discovering and recognizing them. However, whether you do or not is insignificant. What counts is whether form•Z is a system that can satisfy your requirements and maybe even inspire you in your design endeavors.

Since the introduction of form•Z version 1.0 in early 1991, the program has been updated many times. form•Z started as a solid modeler and version 2.0 transformed it into a solid and surface modeler. Subsequent upgrades introduced 3D text, 3D symbols and instances, and many other advanced modeling operations. More recently, version 3.0 expanded the program’s parametric personalities dramatically, made its interface completely customizable, and introduced context sensitive tool options palettes, which make it easier for you, the user, to create the 3D forms you imagine. Versions 4.0 and 5.0 introduced many new nurbs and smooth modeling tools, a new open architecture, a script language, and the first plugins. Version 6.0 offered some extensive animation enhancements and version 6.5 introduced significant rendering improvements.

While many of these updates and upgrades followed our own schedule of development, many others responded to a variety of suggestions offered by our users. We shall be delighted to be able to continue this tradition. Please, call us if you need any help and if you have questions about a feature of the program. The purchase of the program comes with 120 days free technical support and updates. From there on you can continue these services by subscribing to the form•Z Family Plan whose cost we keep as low as possible. Whether you subscribe or not, please, contact us with any suggestions you may have. There is an excellent probability that your suggestion will be incorporated into the program, provided it is consistent with form•Z’s structure and spirit. Training sessions at your location or our offices can also be arranged.

Here is how to reach us:

- Phone: (614) 488 - 8838, 9:00 AM - 6:00 PM (EST), Monday through Friday.
- Fax: (614) 488 - 0848
- e-mail: general: formZ@formZ.com
  technical support: support@formZ.com
  sales: sales@formZ.com
1.1 The versions of form•Z and their differences

Two versions of the program are available: form•Z and form•Z RenderZone Plus. They are all available for the Macintosh and Windows.

The regular version of form•Z contains the full set of modeling tools, as well as drafting, animation, and a range of plotting and rendering procedures, up to smooth shaded images produced by a z-buffer algorithm. These include transparencies, antialiasing, and both hard and soft shadows cast by multiple lights, which are always distant (sun) lights.

form•Z RenderZone Plus is the version that offers photorealistic rendering, including z-buffer and ray tracing. It contains all the capabilities of the regular version of form•Z and, in addition, it offers advanced rendering effects, such as texture mapping, decals, reflections, refractions, environment maps, foreground and background effects, bumps, and more features. It also offers different types of lights, producing both soft and hard shadows. It also includes advanced techniques for Global Illumination including radiosity, final gather, and ambient occlusion. The rendering engine in form•Z RenderZone Plus is based on state of the art shader technology provided by LightWork Design, Ltd. of the United Kingdom.

On both the Macintosh and Windows operating systems, all versions of form•Z are one and the same and behave exactly the same when executing operations. However, a few differences exist in the interface of the program when it runs on a Macintosh and on a Windows machine. These are caused by certain features that are inherently different on the two platforms. These differences are summarized below and are also pointed out throughout this manual.

• On the Macintosh there is one additional menu, the Apple menu (). The About form•Z item appears in the Apple menu on the Macintosh and appears in the Help menu on Windows. This item presents information about the version of form•Z you are running.

• The special keys on the Macintosh and Windows keyboards are different, which affects the key shortcuts available for different operations of form•Z. The option key is available on the Macintosh but not on Windows. The alt key is available on Windows but not on the Macintosh. However, these two keys cannot be used in the same manner since the alt key is reserved for many system operations on Windows. The key equivalents as implemented in form•Z for both the Macintosh and Windows machines are discussed in section 1.6.
• The graphic keyboard layout that is available on the Macintosh is not on Windows. On the Macintosh this is used to show the key equivalents of different operations, when Keyboard... is selected from the Help menu. On Windows, the same Keyboard Help dialog is invoked, and you can still get information on the key commands, but the keyboard graphic is not present.

• Some of the image formats supported on the Macintosh and Windows are different.
  • PICT is available on the Macintosh; on Windows it is available as an option of the QuickTime image format.
  • Metafile is only available on Windows.

• All the dialogs on the Macintosh and Windows look different due to different graphics styles on each platform, but their content is identical. Exceptions to this are four groups of dialogs that are strongly linked to the operating system and their content is significantly different on each platform, even though they perform similar tasks. These are: Color Picker, Folder/Directory Selector, File Save, and File Open.

Also, dialogs invoked by the Page Setup... and Print... menu items will vary depending on the version of the operating system being used, as well as the printers and drivers that are available.

• Windows files do not include a signature as the Macintosh files do. Therefore all file types are recognized by their file extensions only. File extensions are three characters following the “.” after the file name. These conventions are discussed in more detail in section 3.1.1.
1.2 About this manual and its style

The form•Z User’s Manual consists of six chapters bound in four volumes. An additional volume contains the sixth chapter covering the rendering and radiosity features of form•Z RenderZone Plus. Each volume contains an index and table of contents. There is also a Tutorial, which is a separate volume. The full set is also supplemented by an Index/Glossary.

Each chapter is broken into sections, each section into subsections, and hierarchical indices are used, such as 1, 1.1, and 1.1.1, respectively. The first volume of the manuals contains chapters 1, 2, and 3. This first chapter is an introduction and offers an overview of the features of form•Z. It also contains a section on the Help environment. form•Z’s interface and graphics environment are discussed in chapter 2, and its menu commands in chapter 3. The second volume contains chapter 4 (modeling), the third volume contains chapter 5 (drafting). The fourth volume contains chapter 6, which only apply to form•Z RenderZone Plus. The fifth volume contains chapter 7 (animation).

Is it necessary to read this manual, before using form•Z? The answer to this question can be both “yes” and “no.” We expect any user to be able to start using the program and to start creating models without reading a single word of this manual. A major effort has gone into making form•Z as intuitive as possible. Especially users with prior CAD and 3D modeling experience are expected to be able to go a long way, without ever consulting this manual. However, there are also certain details and optional variations that are highly unlikely to be “discovered” without reading about them in this manual. In addition, form•Z contains certain operations that are unique and have no counterparts in other programs on the market. These operations will require some “learning” of the principles involved. Consequently, our recommendation is that even experienced users should read this manual.

The way it is written, this manual intends to serve two purposes: first, to offer a progressive presentation of the features of form•Z, and second, to be used as a reference, whenever information about how to use a certain feature needs to be refreshed. With minor exceptions, the structure of the manual follows the structure of the system and its commands. This makes it easy to start using the system and read the manual in a parallel and gradual fashion.
As the material is presented progressively, we have tried to make the individual sections as complete and self-contained as possible. This has required some material to be discussed more than once. There is always one place where a theme is discussed in full detail. The same theme may also be summarized in other places, when necessary, to make the discussion in that context complete. Cross references have been included throughout, whenever the material in one section relates to the material of another.

There are two ways for finding the place in this manual where a specific term, operation, or theme is discussed: the Tables of Contents and the Index, which complement each other. These are found in each of the individual volumes. The Table of Contents lists all the tool icons and all the menu bar commands of the system in the order they are discussed in the manuals. The menu commands and the names of the tools are also listed in the Index, in alphabetical order. In addition, the Index contains the names of the dialog boxes, and the parametric options. The Help menu can also be used to access on-line summary information about the tools and commands of the system.

As written, the form•Z User’s Manuals cover the operations of the program on both the Macintosh and Windows platforms. Where there are differences, such as in menus, due to different key equivalents, or File Open/Close dialogs both versions are shown. Where the content of menus and dialogs are identical on both platforms they are shown only once.

The style of this manual follows an image oriented format. The menu commands, the names of the dialog boxes, as well as the optional items that are selected from the dialog boxes are in the Chicago font. This is the standard font used for the menus and dialogs by MacOS and works well for highlighting the commands and dialog options even when the manuals are read for usage of form•Z on Windows. The icons that represent tools frequently appear in the text next to the name of the tool. The names of the tools are Capitalized. Bold italics are used when a term or operation is discussed for the first time or when a specific distinction needs to be emphasized.
1.3 The features of form•Z

A session with form•Z typically starts with a modeling window where you can generate primitive or other basic objects that you can then transform and synthesize into composite objects or configurations of objects. You communicate with the system by picking menu bar commands or tools from the tool bars. form•Z has been designed in a manner that maximizes graphic interaction. However, numeric input is also available, whenever more accuracy is required. All operations are dynamically executed and their result is immediately visible on the screen. form•Z is a what you see is what you get system. The following paragraphs offer a summary overview of the features of the program.

Modeling objects and drafting elements

The entities that are manipulated by the modeling module of form•Z are called modeling objects or simply objects (chapter 4). Those manipulated by the drafting module are called drafting elements or simply elements (chapter 5).

The modeling objects that can be created and manipulated in form•Z can be a variety of parametric entities or polygonal objects. Both can be either solid or surface objects. They both exist in the same 3D world and a variety of operations can be applied to initially “flat” surface objects to transform them into 3D entities. In addition, composite surface meshes are typically created as true 3D objects from the beginning. Thus the main difference between the solid and the surface objects is that the former are completely enclosed by a set of bounded surfaces while the latter are not. A variation is the surface solid, which consists of double sided surfaces that completely enclose it. It thus has all the structural characteristics of a solid but contains no volume.

A special variation of a polygonal object is the enclosure, which is a double line (“wall”) entity that can be a surface or a solid object. Enclosures behave as any other object when operations are applied to them and, as they represent entities that contain “space,” they are intended to facilitate the modeling of floor plans and buildings.

Parametric objects in form•Z include the set of analytic primitives, nurbz, Bezier and Coons patches, metaformz, text, and spherical objects. Also available is a hybrid type of an object called controlled mesh (c-mesh), that consists partly of parametric NURBS surfaces and partly of polygonal surfaces, and can be surface or solid objects. Parametric objects are internally stored with the parameters that generated them, which makes it easy to edit them and change their form after the initial creation. Examples of form•Z objects are shown in Figure 1.3.0.1.
A major characteristic of \textit{form-Z} is its ability to apply operations to complete objects as well as to parts of polygonal objects. These parts, called \textbf{topological levels}, are the \textit{point} (or vertex), the \textit{segment} (or edge), the \textit{outline}, the \textit{face}, the complete \textit{object}, and \textit{group} of objects. A special type of a part that can also be manipulated by itself is the \textit{hole}.

There are three main types of drafting elements: \textit{delineators}, \textit{annotations}, and \textit{symbols}. There are also a few special types of drafting elements: The \textit{area} is used for manipulating other elements and does not appear in the final drawing. The \textit{panes} are shapes that specify windows into the drafting world, while in layout space. The \textit{image element} offers the ability to include bit-mapped images in a drafting project. Given the exclusively 2D nature of drafting, all the drafting elements are 2D entities. From the drafting elements, the delineators are used for the representation of the physical entities that need to be graphically described. They are connected sequences of segments, called \textit{polylines}, that correspond to the surface objects of the modeling environment, or \textit{arcs}. The annotations (\textit{text}, \textit{dimensions}, and \textit{leader lines}) are used to attach explanatory information to a drafted drawing. The symbols are constructed from other drafting elements and facilitate repetition. The topological levels of the drafting elements are the \textit{point}, \textit{segment}, and complete \textit{element}. The \textit{area} is also treated as a special type of a topological level.
Projects, imager sets, and windows

form•Z is organized in projects. A project corresponds to a document or file and consists of a modeling and/or a drafting part (section 2.1). When open on the screen, a project may have multiple windows, some modeling and some drafting. The modeling windows are views into the 3D world of modeling and the drafting windows are views into the 2D drafting space. When saved, a project is saved as a single file which retains both the modeling and the drafting environment and their corresponding windows. form•Z also includes an additional type of window, the imager set window, which lists views of projects to be rendered or printed in batch mode by form•Z Imager, a utility included in the program.

Interface

form•Z features a highly interactive graphic interface that allows you to work directly in 3D space, through axonometric, isometric, oblique, or perspective views. You create, edit, and manipulate objects relative to a reference plane which can be one of the three Cartesian planes or any arbitrarily positioned plane you may define (see subsection 2.2.1). Multiple associated windows and frames offer you the option to work in different views and scales simultaneously (subsection 2.1.1). Integration of 2D and 3D modeling provides you the flexibility to work with 2D and 3D entities in the same environment.

You most often instruct form•Z what to do by selecting one of the iconic tools in the tool bars. The modeling and the drafting environments have their own independent tool bars. In each window there is a window tool bar that is used to control the graphics environment. From all the tool bars, groups of tools can be torn off to become floating tool palettes. The modeling and drafting tools can also be completely customized, including rearranging their individual tools and even turning them off, which offers the option to reduce your tool palettes to only the necessary tools (see section 1.8). Most of the tools are affected by options you select from either dialogs or the Tool Options palette. You can invoke the dialogs directly from the tools by double clicking on them. The Tool Options palette displays the options of the currently selected tools and is automatically updated as soon as another tool is activated. The palette is open by default, but can be closed.

While the Tool Options palette has by default a fixed position to the right of the project window, it is a floating palette and can be placed elsewhere. There are more floating palettes dedicated to specific operations. Some of them are opened when you launch the program. The others can be invoked from the Palettes menu.

Virtually unlimited undo/redo operations allow you the freedom to experiment with a design and eliminate the fear of making errors or destroying the work you already completed. You can undo any number of previously executed operations in one step and you can even save the undo records with the project and be able to undo operations you did in previous sessions.
Simultaneously available **prepick** and **postpick** modes allow you to decide which method best fits your needs. You can preselect any number of entities and apply an operation to all of them at once, or you can first select a tool and apply its operation to entities, one at a time (section 4.3).

**Modeling**

You can generate primitive forms, such as cubes, spheres, cones, and tori, by drawing directly on the screen. Variations of primitive shapes are the metaballs and the spherical objects that include the set of Platonic solids, geodesic spheres, and soccer balls. You can also create **surface** (2D) and **solid** (3D) objects directly by setting a modifier to tell the program what type of object you are generating and using another tool to draw shapes (section 4.2). These shapes include **rectangles**, **polygons**, **circles**, and **ellipses**, as well as single **points**, single **segments**, **vector lines**, a variety of **parametric splines**, **stream lines**, and **arcs**. These can be drawn independently or you can freely mix them in a composite drawing. They can be generated as single lines or properly cleaned double (“wall”) lines, called **enclosures**. Any of the 2D surface shapes can become the base for the direct generation of 3D solids, through parallel or point **extrusions** or 3D enclosures can be generated from the 2D enclosures. These are objects that contain space, with or without floors and ceilings.

You can generate **derivative objects** from any surface or solid object, or their parts (section 4.4). In addition to parallel extrusions, extrusions to points (convergences), and enclosures, you can also derive **parallel objects**, **objects of revolution** (lathing), **screws and bolts**, four variations of **sweeps**, multi-source and multi-path **skins**, **lofts**, a variety of **stairs**, **sections of solids**, **3D terrain models** from 2D contour lines and sites, **projections** of objects, and **unfolded** objects.

In addition to the repertoire of parametric objects you can generate in **form•Z**, which includes the **analytic primitives**, **nurbz**, **patches**, and **metaformz**, a hybrid type of objects, called **controlled meshes** (c-meshes), is available. They can be generated using a complete range of mathematical methods that include NURBS, splines, and Bezier (section 4.5). Like the parametric objects, they can be edited and changed interactively. Meshed objects can also be reformed by smoothly pushing or pulling areas of the mesh, and by applying random or wave disturbances, as well as a variety of **deformations**. You can also **trim**, **split**, and **stitch** meshes. Points, edges, or both points and edges of objects can be smoothly **rounded**, and stitch lines at which two intersected surfaces meet can be blended.

You can apply **Boolean** operations, which include **union**, **intersection**, and **difference**, to either 2D surface objects or 3D solids (section 4.6). They offer constructive solid geometry methods for composing primitive shapes into complex forms.
You can execute 2D/3D geometric transformations graphically, which allows you to move, rotate, scale, or mirror either individual entities or a number of entities simultaneously (section 4.22). These operations are executed dynamically in 3D space, which allows you to visualize their effect. You can apply transformations to the entities directly, or to single or multiple copies at repetitive or uneven intervals. The transformations can be applied at any of the topological levels (Point, Segment, Outline, Face, Object, Group, or Hole). You can also record a sequence of transformations as a macro that you can add to the system’s library of macros and subsequently apply it through a tool (subsection 4.22.4).

Form editing and sculpting in 2D/3D allow you the freedom to reshape objects. These operations include insertions and deletions of points, segments, curves, faces, or volumes (sections 4.4 and 4.25, respectively), and you can use them to “sculpt” 3D forms. You can use the Attach tool to position objects relative to other objects, the Align/Distribute tool to line up or evenly distribute objects or elements, the Extend tool to extend segments to surfaces, and the Place tool to place shapes and objects on lines or other objects (section 4.23).

Other advanced modeling features of form-Z include the ability to define 3D symbols, which can be stored in libraries and placed as instances, whenever repetitive modeling is appropriate.

Plotting, rendering, and animations

In the regular version of form-Z, a variety of display options allow you to visualize models at different levels of detail (section 3.6). These include wire frame, interactive shaded, quick paint, hidden line, surface rendering, and shaded rendering, which includes smooth shading, antialiasing, and transparency. Colors, smoothness, and transparency are interactively assigned to objects or their surfaces (section 4.24). Shadows from one light (sun) can be included with quick paint, and surface rendering. Both soft and hard shadows cast from multiple lights can be included with shaded renderings. form-Z RenderZone Plus, in addition to all the rendering options available in the regular version, offers photorealistic rendering with many advanced rendering effects produced by rendering algorithms at a variety of levels, including z-buffer and ray tracing (chapter 6). form-Z RenderZone Plus offers the most accurate simulation of lighting effects. It is based on advanced Global Illumination techniques including final gather, ambient occlusion, and radiosity (chapter 6).

Graphic control of perspective, isometric, oblique, and axonometric views allows you to interactively change the viewing positions (subsection 2.2.8). For additional control, you may manipulate the cone of vision to accurately define the position of the viewer or camera, center of interest, light sources, angle of vision, and the positions of the hither and yon clipping planes (subsection 3.5.6). The viewing parameters can also be set through numeric input (subsection 3.5.3) and the position of the sun light can be determined by selecting the geographic location of a site, a time, and a date (subsection 3.5.5). Panoramic views are also available and can be displayed as static images or may become the basis for QuickTime VR movies.
Objects, surface styles, lights, and cameras can be animated by a variety of methods available within form\-Z. Animated sequences can be rendered with any of the display methods available in form\-Z, which ranges from wire frames to photorealistic renderings. Animations can be played back from within form\-Z or can be conveniently exported.

**Drafting**

The drafting module complements the modeling capabilities of form\-Z (chapter 5). Images of models can be freely transported from the modeling to the drafting module, and vice versa. The drafting capabilities include line, arc, circle, ellipse, polygon drawing and editing, double line (wall) elements, compound elements, Boolean operations, geometric transformations, associative hatching, associative and non-associative dimensions, customized line styles and weights, both outline and bit-mapped text, symbols, and a special layout space that makes it easy to composite drawings for printing and plotting.

**Other features**

form\-Z also features floating point precision, virtually unlimited layers that can be hierarchically structured in both modeling and drafting, some thirty import/export file formats, English and metric units, user defined preferences for setting the default parameters, and on line help.
1.4 form•Z and the 3D standards

The design and implementation of form•Z has to a large extent followed the guidelines recommended by the designers of Macintosh and Windows. However, our implementation has primarily followed the spirit of the desktop paradigm, rather than blindly copying the ways in which other programs have interpreted the guidelines. Given that many of the “standards” were established by simpler word processing or 2D drawing and painting programs, and that they cannot accommodate many of the requirements of 3D modeling, we had no choice but to adapt standards set by other 3D applications outside the Macintosh and Windows worlds, and to develop our own alternative interpretations. Therefore, some of the features of form•Z may at first glance appear to deviate from the standard ways. When it appears so, look harder. form•Z subscribes to a philosophy of choice and the chances are that it has an option to also allow you do things the way you may be more familiar with. Yet, you may be better off insisting and acquainting yourself with the “different” ways of form•Z. You may soon discover that these ways are not after all that different and, to the extent they may be, they make good sense. After you let yourself get used to these different ways, they may actually prove a lot more productive for you. As we know that for nearly a decade form•Z has been heavily imitated by other programs, it may well be that it offers a demonstration of the direction into which the 3D standards are evolving.

The following paragraphs summarize the main areas of the implementation that risk being perceived as deviations, offer a short discussion of the reasons, and point out the alternative options that may be available.

Selection highlight: In form•Z, selected entities are highlighted with color, which by default is red, but can be changed. Other programs use little squares, called handles, on the corners of the bounding boxes of shapes. In form•Z, handle-like markers are used for highlighting the controls of parametric objects, when they are edited.

The main reason for not generalizing the use of handles is that the displays of fairly complex polygonal 3D models would be hardly readable. In addition, the distinctions that the handles imply do not suffice for the full range of topological levels (point, segment, outline, face, volume, group), as they exist in form•Z’s polygonal 3D modeling. Where they work well for marking parametric controls, they are used.
Mouse clicking versus dragging: In form•Z, operations, such as drawing a rectangle or moving an entity, are executed by individual clicks of the mouse. In some other programs the mouse is dragged from the first point to the second and clicking is used from there on, whenever an operation requires more than two input points.

There are at least four major reasons for implementing the click rather than the drag method, for most operations of form•Z:

1. Uniformity and consistency: You do not have to switch from dragging to clicking for the frequent operations that require more than two input points.
2. Mixing operations: As you apply a tool you may need to apply another operation. For example, as you draw a rectangle you may need to zoom in or pan. This is not possible with the drag method since the mouse is occupied and unavailable for selecting another tool.
3. Comfort and precision: Freeing your hand from the requirement to continue pressing the mouse button allows you to both relax your arm, and take your time in determining where your next click should be.
4. Mixing graphic and numeric input: You can enter one point graphically and the next point numerically, which cannot be done when you drag the mouse.

While the click-click is the general way of form•Z, the program also recognizes the cases where click-drag is the natural way and uses it. For example, when you draw a frame for picking or a frame for zooming, or when you edit the controls of parametric objects you click and drag.

Command execution: In form•Z, you can apply a tool and execute an operation by either prepicking or postpicking the objects. Most other programs only offer the prepick method.

Prepicking is typically perceived to be consistent with the “noun-verb” approach of the desk top paradigm. However, there seems to be an argument about what the “noun” really is and some of the established CAD programs appear to follow a different paradigm. In a way, form•Z is avoiding the argument and suggests that you the user should decide how you wish to execute your operations. Admittedly, some operations work better with prepicked objects while others work better with postpicked objects. It should also be noted that, rather frequently in 3D modeling, one object does something to another object, which is a “noun-verb-noun” situation. Such operations are handled by both the prepick and postpick method.
**Moving:** In *form-Z*, entities are typically expected to be moved with the Move tool. Other programs use the Pick tool (Selection Arrow) to move entities, which is also an option in *form-Z*.

In 3D modeling, moving is one of the geometric transformations, which include rotation, scaling (sizing), and reflection (mirror). To treat moving differently would be inconsistent. Thus all the geometric transformation tools have been grouped together and behave the same way. In addition, all geometric transformations can be applied to any of the topological levels, a distinction that can only partially be accommodated when using the Pick tool.

This explanation aside, we understand that many people are accustomed to using the Pick tool for geometric transformations. Therefore, *form-Z* includes an option that allows the Pick tool to be used in this manner.

**Multiple selections:** In *form-Z*, to pick an entity while other entities are already selected, you simply click on the entity. By default, you do not need to press the shift key as other programs require.

In 3D modeling, the selection of more than one entity is the rule, rather than the exception, and there is no reason to burden it. In *form-Z*, use of the shift key is typically reserved for the pick parade function (parading through entities that lie on top of each other) which has a rather major significance in 3D modeling.

Here again, because many users are expected to be accustomed to using the shift key in the multiple selection process, *form-Z* includes it as an option. You can instruct the program to require use of the shift key for multiple selections.
1.5 Installation and memory management

The instructions for the installation of form•Z for both Macintosh and Windows machines, are shipped on a separate sheet, included in the package. Please, follow them to install your system. Note that the installation on each of the two operating systems is different.

**Macintosh OS X:**

form•Z requires a Macintosh running OS X 10.4 or later. form•Z requires a minimum of 512 MB of RAM. We strongly recommend using 1 GB or more RAM when creating models of significant magnitude. The software also requires 1 GB of hard disk space.

Macintosh OS X is inherently a virtual memory based system, which is sometimes interpreted as making the quantity of physical memory available insignificant. This is not true. While programs will run with even small amounts of RAM available to them, they will run quite inefficiently. Thus the rule remains that the more physical memory you have the better and faster the program will run. In other words, the less frequent use you make of the virtual memory that OS X makes available, the better your program will run. OS X automatically manages the virtual memory disk space.

You also need to be aware of the size of the disk space available to the system. We recommend against running form•Z with less than 100 MB. The undo/redo records, which offer you the ability to save undos with your project, are by default written on the disk. If the system cannot find sufficient space to produce these records, the undo/redo functions become unavailable. When this occurs, the system displays an alert box to warn you that these commands are no longer available.
Windows:

**form•Z** for Windows can be run on Intel based PC’s with a minimum Pentium processor. It requires that you use the Microsoft Windows 2000, Windows XP, Windows ME, or Windows Vista.

Note that the Windows operating system is also an inherently virtual memory based system, as OS X is. The discussion above about how to use virtual memory also applies to the Windows machines. If you run out of memory when using **form•Z**, you will either have to add more RAM to your system or increase the size of your Windows page file. This is the area of your hard disk that is used by Windows for virtual memory. The larger the page file is, the more virtual memory you have available. If you are not sure how the size of the page file is set, we recommend that you consult your Windows documentation.

When **form•Z** is running, it can be asked to report the memory available in the **Memory Box**, located on the lower margin of the active window. More details about the memory box can be found in subsection 2.2.9. We recommend against running the system when the available memory is less than 1 MB, especially if you intend to generate objects of a considerable magnitude, to plot a hidden line display, or to generate a rendering of a complex scene. Note that, from a memory management point of view, the magnitude of an object is not its physical size, but the number of points and edges it contains.
1.6 Keyboard commands

form•Z contains two main types of commands: menu commands (or items) and iconic commands (or tools). Both types are executed in the standard manner by selecting them with the mouse. On Windows, menus and menu items can also be selected using the alt and the arrow keys from the keyboard.

Many of the menu items and some of the tools have preset keyboard equivalents assigned to them and can be executed by pressing a combination of keys. These as well as all the other menu items and tools can be assigned customized key equivalents by the user (see section 3.2.7). The key equivalents for the menu items are noted in the menus. For the tools, they are noted in the Help environment (see section 1.8). A complete list of the default form•Z keyboard commands is also contained in the Appendix and on the back of the Quick Reference Guide.

Due to the differences of the keyboards on Macintosh and Windows machines, the key equivalents are different on each platform. The following table summarizes the correspondence between the keys on the two platforms:

<table>
<thead>
<tr>
<th>Macintosh</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>command ((⌘, ⌘))</td>
<td>ctrl</td>
</tr>
<tr>
<td>option</td>
<td>ctrl + shift</td>
</tr>
<tr>
<td>control</td>
<td>ctrl + alt</td>
</tr>
<tr>
<td>control + option</td>
<td>ctrl + shift + alt</td>
</tr>
<tr>
<td>option + shift + &lt;key&gt;</td>
<td>ctrl + alt + &lt;key&gt;</td>
</tr>
<tr>
<td>delete</td>
<td>backspace</td>
</tr>
<tr>
<td>return</td>
<td>enter</td>
</tr>
</tbody>
</table>

Following are the most commonly used key commands:

- The delete or backspace key can be used in the place of the Delete tool to delete a highlighted entity (when there is no text selected in the Prompts or Tool Options palettes).

- The return or enter key can be used in the place of the default button commands of the dialogs, or to complete numeric input.

- The tab key can be used to move the insert cursor to the next text field in a dialog or in the Prompts palette. Shift + tab moves the cursor to the previous field.

- The \((⌘ . (command + period)\) combination of keys on the Macintosh or ctrl + period on Windows can be used to interrupt the execution of a lengthy operation or to cancel a dialog. On both platforms, the esc key can also be used to achieve the same result (assuming he default key assignment for "cancel" has not been changed).
1.7 Requesting information from the system

There are three types of information you may require from the system:

- The first is when `form·Z` beeps and abstains from the operation you attempted.
- The second is when you require information about the currently active project, such as object types and counts, topology types and counts, and remaining free memory.
- The third is when you are uncertain about how to execute a command and you need help.

Whenever the system beeps and cancels the execution of an operation, it also posts a message about why the operation cannot be executed. This message may be presented to you immediately, if the `Show Error Message Immediately` option is selected in the `Project: Warnings` section of the `Preferences` dialog, which is invoked from the same item in the `Edit` menu. When this option is off, the message is entered into a message file. The message file can be read by selecting the `Error Messages...` item from the `Help` menu. Selecting this item opens the `Error Messages` dialog shown in Figure 1.7.0.1, for both the Macintosh and Windows. The window in the center of the dialog contains a list of error messages presented by the system, with the most recent error at the bottom of the list.

The `Error Messages` dialog also contains a `Save` and a `Clear` button command. `Save` allows you to save the list of error messages as an ASCII file for later reference. When selected it invokes the standard dialog for saving, where you can assign a name to the file. The `Clear` button clears all the messages and starts fresh the next time a message is posted.

![Figure 1.7.0.1: The Error Messages dialog on (a) Macintosh and (b) Windows.](image_url)
form\textsuperscript{-}Z also keeps detailed information on the currently active project. You can access this information by selecting the \textbf{Project Info...} item in the \textbf{Help} menu. Selecting this item invokes the \textbf{Project Info} dialog, the three tabs of which are shown in Figure 1.7.0.2.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.7.0.2.png}
\caption{The \textbf{Project Info} dialog, with the (a) Modeling, (b) Drafting, and (c) Project File Info tab selected.}
\end{figure}
1.8 The Help environment

Whenever you are uncertain about how a feature of the system works, you may either come to this manual for a complete explanation or you may invoke some summary discussion. How to access the on line manual is discussed at the end of this section. For a summary explanation you can select the General... or some of the other items in the second group of the Help menu.

The General... item invokes the General Help dialog, shown in Figure 1.8.0.1. This dialog consists of a small window that provides summary information about the currently selected tool. The name and the icon of that tool are also shown above the window. The General Help dialog also contains six command buttons that branch to additional information. These branches can also be accessed through their own items in the Help menu.

![General Help dialog](image)

**Figure 1.8.0.1:** The General Help dialog.

### Introductory Help...

When you select this button or the Introduction... item in the Help menu, the system presents an introductory overview of the features of form·Z, in a scrollable dialog (Figure 1.8.0.2).

![Introductory Help dialog](image)

**Figure 1.8.0.2:** The Introductory Help dialog.
Keyboard Help...

When you select this button or the Keyboard... item in the Help menu, the program invokes the Keyboard Help dialog, which is as shown in Figure 1.8.0.3 on the Macintosh. This dialog contains an image of your keyboard and, when you press a combination of keys, it displays a description of the operation that is programmed for that key, if one exists. The text is displayed for as long as you keep the keys pressed. A Keyboard Help dialog is also available on Windows, however, the keyboard image is not provided by the Windows operating system, so this is missing. Descriptions also appear when you press key combinations.

The default keyboard commands available in form-Z are listed in the Appendix and on the back of the Quick Reference Guide.

Figure 1.8.0.3: The Keyboard Help dialog (a) on the Macintosh and (b) on Windows.
Modeling Tools...

When you select this button or the item with the same name in the **Help** menu, the system presents all the modeling icons under their respective palette names in a scrollable window (Figure 1.8.0.4). The icons are in the order they are set in the **Icon Customization** dialog, which is discussed in section 1.9. If no changes have been made to their order in the tool palette, the default icon locations will be shown.

As the mouse is moved around, the icon under the cursor is highlighted and a summary description of its workings is presented in the **Description** box at the bottom of the dialog. Also its key equivalent is displayed in the **Key Shortcut** box, if an equivalent has been defined. The dialog can also be used to set new key shortcuts or edit existing ones, as follows:

*Figure 1.8.0.4: The Modeling Tools dialog.*

Double clicking on an icon invokes the **Shortcuts** editor shown in Figure 1.8.0.5. It contains buttons that can be used to **Add**, **Edit**, and **Delete** a key shortcut for the selected tool. See section 3.2.7 for details on shortcuts.

*Figure 1.8.0.5: The Shortcuts editor.*
Drafting Tools...

Selecting this button or the item with the same name in the Help menu invokes the Drafting Tools dialog as shown in Figure 1.8.0.6. This dialog box works as the Modeling Tools dialog.

As with the Modeling Tools dialog, double clicking on an icon invokes the Shortcuts editor (Figure 1.8.0.5).

![Drafting Tools dialog]

*Figure 1.8.0.6:* The Drafting Tools dialog.
Menus...

When you select this button or the same item in the Help menu, the system presents the Menu Help dialog box that contains all the menu bar items unfolded and in the order they appear on the screen. This is shown in Figure 1.8.0.7. As the mouse is moved around the dialog, the menu item under the cursor is highlighted and a summary description of its workings is presented in the box at the bottom. There are also buttons that return to the Main Help dialog or exit the Help environment.

As with the Modeling Tools dialog, double clicking on an icon invokes the Shortcuts editor (Figure 1.8.0.5).
**Window Tools...**

When you select this button or the same item in the **Help** menu, the system presents the **Window Tools** dialog that shows all the window icons unfolded in the order they appear in the window tool palette (Figure 1.8.0.8). It works as does the **Modeling Tools** dialog.

As with the **Modeling Tools** dialog, double clicking on an icon invokes the **Shortcuts** editor (Figure 1.8.0.5).

*Figure 1.8.0.8: The **Window Tools** dialog.*
Manual...

Selecting this item from the third group of the Help menu launches the form•Z Manual Viewer with which you can browse the form•Z User's Manual.

The form•Z manuals are available as a PDF document and are accessible from form•Z, the Imager, Render Client, Render Server, and Script Editor applications. Various sections of the manual can be accessed in a number of different ways, the first being through the Manual... item of the Help menu, which opens the manual to the title page.

The second way to access the form•Z manuals is through the Manual... item of the contextual menus available for many of the form•Z palettes. This menu item opens the manual to the section that describes the palette.

Another way to access the form•Z manual is by shift-clicking on a tool icon or a menu item. This method opens the manual to the section that describes the tool or menu item. This same shift-click mechanism also works from the icon and menu help dialogs.

The form•Z manual can also be accessed by shift-clicking on items in the key shortcuts list on the Key Shortcuts Manager dialog. This provides manual access for commands that are only available as key shortcuts.

Many dialogs in form•Z contain a help button ( ). Clicking on this button opens the manual to the section that describes the dialog.

The form•Z Manual Viewer application that displays the manual looks and behaves differently on Windows, and Mac OS X. This is due to the availability of different PDF document viewing technologies on these platforms. Each of these applications is described in some detail in section 3.10.2.
1.9 Customizing the tool bars

form•Z allows to fully customize its tool palettes to your own liking, if you so desire. This refers to its three main tool bars: the modeling, drafting, and windows tool bars. After they have been customized and saved, the palettes are automatically retained from session to session.

Customization of the tool palettes includes the following specific capabilities:

- You can choose to include or exclude any of the tools available in form•Z.
- You can structure the tool palettes to contain any tools in any combination you prefer.
- You can set the tool bar to be displayed as one, two, or as many columns as you desire.
- You can change the appearance of the icons by selecting one of three variations.
To customize the tool palettes, select the **Customize Tools**... item at the bottom of the Palettes menu shown in Figure 1.9.0.1. This invokes the **Icons Customization** dialog shown in Figure 1.9.0.2.

This dialog contains three windows, labeled **Tool Bar**, **Tool Palettes**, and **Tool Set**. Each displays a different aspect of the form·Z icons. It also contains two pop up menus, labeled **Category** and **Icon Style**, and a number of button commands.

**Category**: The item selected from this pop up menu determines which icon set will be displayed in the **Customization** windows. There are three choices: **Drafting Tools**, **Modeling Tools**, and **Window Tools**. Selecting one of these items fills the windows with icons of the respective category. The default is either modeling or drafting tools depending on the active window type at the time the **Customize Tools**... item is selected.

![Figure 1.9.0.1: The Palettes menu.](image)

![Figure 1.9.0.2: The Icons Customization dialog.](image)
**Tool Bar:** This window displays the tool bar of the current category. Behind each displayed icon there is a tool palette, as in the program. The full tool palettes are show in the **Tool Palette** window to the right. Within the tool bar window, tool palettes can be removed (deleted), repositioned within the tool bar, renamed, and new tool palettes can be added. Tool palettes are selected by clicking on their icon. Selected icons are highlighted with a red box that frames them. Pressing *shift* allows the selection of multiple icons.

The following actions can be performed:

- **Removing** a tool palette: Select the icons and press *delete* or the *Remove* button at the bottom of the window.

- **Repositioning** a tool palette: Click and drag the icon to the desired new location. While the icon is dragged, its boundary follows the mouse and the new position is highlighted as the icon boundary tracks across the palette. Positioning the cursor between icons highlights the line between the icons, where the insertion will occur. Once the desired position is indicated, release the mouse and the icon is moved to the new location. The remaining icons shift to adjust for the newly inserted icon. If multiple icons were selected, they all move to the new location. An example is shown in Figure 1.9.0.3.

- **Replacing** a palette: Click and drag the icon onto another icon. The former icon replaces the latter.

- **Renaming** a palette: Double click on its icon to invoke the **Palette Name** dialog shown in Figure 1.9.0.4. Type the desired name and press *OK*. The names of the palette are shown in the **Tool Palettes** window.

The two buttons in the tool bar box work as follows:

- **New...**: Clicking on this button creates a new tool palette. The **Palette Name** dialog (Figure 1.9.0.4) is invoked for naming it. After pressing *OK*, the new tool palette is added at the bottom of the tool bar and appears in both the tool bar and tool palettes windows. It is initially empty and tools can be added in the tool palettes window.

- **Remove**: As already mentioned, clicking on this button while an icon is highlighted, it removes the tool palette from the tool bar. This is the same as pressing the *delete* key. If no icon is selected, this button is dimmed.

**Figure 1.9.0.3:**
Repositioning the Text palette:
(a) before and (b) after the move.

**Figure 1.9.0.4:**
The **Palette Name** dialog.
**Tool Palettes**: This window displays the tool palettes of the current tool bar. In it, icons can be removed (deleted) from a palette, repositioned within a palette, or moved to another palette. Icons are selected as with the Tool Bar window. The following can be performed:

- **Removing** a tool: Select the icon and press the delete key or the Remove button.

- **Repositioning** a tool in the same palette: Click and drag it to the new place in the palette.

- **Moving** a tool to another palette: Click and drag it from one palette to another, to the desired location, as shown in Figure 1.9.0.5.

There are also two options in this window:

**Remove**: As already mentioned, clicking on this button removes the tools highlighted in the Tool Palettes window. This is the same as pressing delete. When no icons are picked, this button is dimmed.

**Show In Alphabetic Order**: When this option is on, the tool palettes are sorted in alphabetic order. When off (default), they appear in the order they are in the tool bar.

**Tool Set**: This window displays all the tools available in form•Z for the selected category. The icons are organized in the same order and groups they are in the Key Shortcuts Manager dialog. Tool icons in this window are selected as in the other windows. The following operations are available:

- **Placing** tools into palettes: Drag an icon from the tool set window into a palette of the tool palettes window (Figure 1.9.0.6(1)).

- **Replacing** a tool: Select an icon in the tool set window and drag it on top of a tool in a palette (Figure 1.9.0.6(2)).

---

**Figure 1.9.0.5**: Repositioning the Spherical Objects tool: (a) its original place and (b) its new position.

**Figure 1.9.0.6**: (1) Inserting and (2) replacing icon. (a) Before and (b) after.
In addition to the already discussed options that affect individual windows of the **Icons Customization** dialog, there are more options that affect the whole dialog, as follows:

**Defaults...**: This button resets the tool layout for all categories to the system defaults. The **Tool Bar** and **Tool Palette** windows in the dialog are refreshed to show the default layout.

**Save...**: Clicking on this button saves the current layout to a file so that it can be used later. After selecting this button, a standard File Save dialog is invoked for selecting the name and location of the file to save. Tool bar menu layout files have the extension “.mnu”.

**Load...**: Clicking on this button loads a previously saved tool bar menu file. A standard File Open dialog is first invoked for selecting the tool bar menu file to load. The **Tool Bar** and **Tool Palette** windows in the dialog are refreshed to show the loaded layout.

**Icon Style...**: This pop up menu determines the appearance of the icons in **form•Z**. There are three options: **White**, **Gray**, and **Color**. They are shown in Figure 1.9.0.7 for the modeling tool bar. **Gray** shows the icons in a style consistent with other versions of MacOS and Windows. **White** displays the traditional “minimalist” **form•Z** icons. **Color** is the default and displays icons that use color 3D graphics and no text. Note that the lack of text is compensated by the help hints that can be invoked by placing the cursor on an icon.

**Figure 1.9.0.7**: The **Icon Style** pop up menu: (a) **White**, (b) **Gray**, and (c) **Color**.
1.10 Saving your selections in the Preferences dialog

You can instruct form•Z to start your session with the options and parametric values you have set in the dialogs in previous sessions, rather than using its own defaults. This is done from the Preferences dialog (Figure 1.10.0.1) that is invoked from Preferences... in the Edit menu.

![Preferences dialog](image)

**Figure 1.10.0.1:** The Preferences dialog.

To apply your own preferences of defaults, you first need to save them. You do this by selecting the Use Preference File option of the System: General category of the Preferences dialog. Then select the Save Preferences... button and enter a name for your preferences file in the Save dialog. The selected name shows under the Use Preference File option. This way you can save any number of preference settings under different names. Saved preferences files can be retrieved by clicking on the Load Preferences... button. Next time you launch form•Z, the settings that were saved under the name appearing in the Current Preferences File field will be applied.

At the left of the Preferences dialog there are a variety of preference categories, which can be customized to your preferred settings. The options of the Preferences dialog are discussed in detail in subsection 3.2.8.
2.0 Introduction

A user typically communicates with form•Z through the mouse and/or the keyboard. Some also use tablets and digitizers, including a 3D digitizer that is supported by form•Z. The system communicates with the user by displaying and continuously updating images which depict the current state of a project. It also indicates what the current states of the tools are and, when appropriate, it beeps and issues messages.

The mouse is used to select menu commands and tools, to turn switches on or off, and to select options from dialog boxes. However, the mouse is above all a drawing device. It is used to trace the lines of a shape, to generate and manipulate the form and the position of objects in 3D space, and to draft. form•Z is primarily a graphic system and one of its major strengths is its ability to interpret and map the inherently 2D input, which occurs on the screen, to 3D space.

As you work, an image of your project’s 3D world is constantly displayed and is immediately revised whenever an operation causes one or more objects to change. At the same time, the numeric values of the position of the mouse are constantly displayed, in either absolute or relative mode, in either English or metric measurements, in a variety of optional formats. At any time, you can override the position of the mouse by typing in one or more numeric values using the keyboard.

form•Z consists of a modeling and a drafting environment, each of which has its own set of tools and its own windows. The screen layout and the behavior of the operations are very similar. However, modeling works primarily in 3D, while drafting is exclusively 2D. This is reflected in the types of images that are displayed in the respective windows.
The default screen layouts for the modeling and the drafting environments of _form•Z_ are shown in Figures 2.0.0.1 and 2.0.0.2, respectively. Note that many aspects of the _form•Z_ screen layout are highly customizable and thus individual users may set it up quite differently from what we show. However, throughout this manual we refer to the default layout. For both modeling and drafting, the screen layout consists of the following elements, where the numbers refer to those marked in the figures.

1. A horizontal _menu bar_ is located at the top of the screen. The _form•Z_ menu bar contains nine menus. In addition, the Macintosh version contains the Apple (🍎) menu. Some of the menus apply to both the modeling and the drafting environments. Others apply only to the modeling or only to the drafting operations. The commands contained in the menu bar are called _menu commands_ or _items_. They are discussed in chapter 3.
2. A two column, vertical tool bar or iconic menu is positioned on the left side of the screen. The tools contained in this bar are the backbone of the operations that can be executed by form·Z. The modeling and the drafting environments have their own independent tool bars, but they both work in the same manner. The modeling and the drafting tools are discussed in detail in chapters 4 and 5, respectively.

3. A graphics project window occupies the large central area of the screen. It is where all the drawing and visualization activities occur. The modeling environment and the drafting environment have their own independent window types, but the two types behave similarly. They are distinguished by the type indicator, Model or Draft, which is contained in square brackets at the end of the window title, located in the middle of the window title bar. Directly under the title bar in the window is a gray bar with the name of the window and its respective view type. This is in red for the active and black for the inactive windows. This is most significant when multiple associated windows are open simultaneously (see subsection 3.3.1). Also available as special types of windows are the Imager Set window that contains a list of views to be batch rendered or printed, and the Script window, where scripts can be typed and edited.

Figure 2.0.0.2: The screen layout of the drafting environment of form·Z.
4. In addition to the standard Macintosh or Windows features, the graphics windows contain a small iconic menu, called *window tool bar*. It is located on the left side of the bottom margin. The tools that are contained here control the graphic environment of the window. They are referred to as *window tools* or *window icons*. They are discussed in section 2.2.

There are also a number of smaller floating windows called *palettes*. When the system is launched with the default preferences, a modeling window and seven palettes appear on the screen: the *Coordinates* (5), *Tool Options* (6), *Lights* (7), *Objects* (8), *Layers* (9), *Views* (10), *Prompts* (11), *Surface Styles* (12), and *Animation* (13). When you open a drafting window, the *Coordinates* (5), *Tool Options* (6), *Line Weights* (14), *Line Styles* (15), *Layers* (9), *Views* (10), *Prompts* (11), and *Colors* (16) palettes are available. Additional palettes are available in both environments.

All the palettes can be opened or closed by selecting their respective item from the *Palettes* menu, which also contains the *Hide Palettes* command. Each palette can be closed, collapsed, moved and minimized using the standard controls in the title bar of the palette. Additionally, the tool groups contained within the tool bar can be torn off and used as floating tool palettes.
2.1 The graphics windows and dialogs

The form•Z documents are organized by projects. Each project has its own independent set of windows. A project may have only modeling windows, only drafting windows, or some of each type.

A form•Z project window is a view into either the 3D world of modeling or the 2D space of drafting. A single large window may be covering the whole central area of the screen, or more than one smaller windows may be visible at the same time. Some of these windows may be drafting and some may be modeling.

In addition to the project windows form•Z has two special types of windows called Imager Set window and Script window. The former is used by form•Z Imager to batch render or print preset views of projects. This window is discussed in section 3.12. The Script window and Script Editor are discussed in section 3.8.1.1 of the form•Z SDK documentation. The discussion in this section applies to project windows only.
2.1.1 Associated windows

All windows of the same project and the same type are views into the same space. For example, all modeling windows of the same project are views into the 3D modeling world of that project. They are called associated windows. All the changes made to the 3D modeling objects of that project are visible in all the project's modeling windows. The same applies to all drafting windows of a project.

Each of the associated windows may be a different view into the same space. This lets you observe the same project from different points of view or at different scales and levels of detail. This is of major significance, primarily when working in 3D space, which can only be displayed on the inherently two-dimensional screen. There are two additional features in form•Z concerning associated windows: tile windows and window frames.

Tile windows are four windows opened simultaneously and displayed side by side, each occupying one quarter of the screen. Three of the windows display orthographic views and the fourth displays a 3D view. These default views may be changed. The tile windows, other than their initial arrangement, are common associated windows, only one of which can be active at a time. Being associated implies that changes made to a scene in one of the tile windows is reflected in all the others. This feature is discussed in more detail in subsection 3.3.1.

The window frames, when they are first invoked, have an appearance similar to the tile windows, but they are quite different in that they are really a single window subdivided into portions that are viewports into the same project and space. These frames are also associated, and an operation applied through one of the frames is also reflected to the others. A capability unique to the frames is continuous drawing from one frame to another, simply moving the cursor into the frame, rather than having to click on the frame. The window frames feature is discussed in more detail in subsection 3.3.2.
2.1.2 Manipulating and activating windows

The graphics windows of *form-Z* follow the standard Macintosh or Windows conventions, depending on which operating system is being used. The content of a window can be scrolled up, down, left, or right, by moving the thumb boxes at the right and the bottom scroll bars, or by clicking on the scroll arrows. The content of the window can also be scrolled with the Hand tool (see subsection 2.2.7).

A window can be resized. A window which has been resized can be toggled between the full screen size and the custom size by clicking on the zoom or maximize box, located in the upper right corner of the window. Windows can also be minimized on both operating systems.

A window can be closed by clicking on the close box at the upper left corner or by executing one of two *Close* commands. One is located in the *File* menu and closes the active project with all its windows. The other is located in the *Windows* menu and only closes the active window (see subsections 3.1.2 and 3.3.1). If the *option* key on the Macintosh or *ctrl*+*shift* on Windows are pressed when clicking in the close box of a window, all open projects are closed. This is the same as selecting the *Close All* command which is also found in the *Windows* menu. When the last window of a project is closed, you are prompted to save it if its contents have been changed since the last save.

More than one window, up to a maximum of 96, may be open at any given time. These windows may belong to a single project or a variety of projects. However, only one window at a time can be *active*. It can be a modeling or a drafting window and it may belong to any of the projects that are currently open. The active window is always the front window.

The first window of a new project is positioned with its upper left corner to the right of the tool palette and just below the menu bar. The window is sized to fill the screen. When additional windows to the same project are opened, their position is progressively adjusted. Each new window’s top left corner is positioned a little to the right and down from the top left corner of the active window. The new window has the same height and width as the active window, except when it extends beyond the right and bottom borders of the screen, in which case its size is adjusted to the border of the screen.

A window can be made the active window in two ways: either by clicking on it, provided some part of it is visible, or by selecting its name from the list of names located at the bottom of the *Windows* menu. There is complete freedom to directly switch from one type of a window to another, and from a window of one project to a window of another project.

The active window receives all the graphic input and responds to the commands which manipulate the graphic environment. Tools that cause changes to the modeling objects or the drafting elements, are independent of the active window and affect the images of all the open windows of the same type and the same project. However, these tools can only be applied through the active window.
2.1.3 Controlling the graphics of modeling and drafting windows

The parameters that control the graphic environment of a modeling window are the position of its 3D coordinate system and its reference plane, the viewing position, the scale, and the snapping switches.

When a modeling window is first opened, it is displayed with a grid and the three orthogonal axes. The directions of the axes are marked with plus and minus signs for each of the X, Y, and Z dimensions. The grid is displayed on the active reference plane, which initially defaults to the XY plane. The significance of the active reference plane is that graphic input is interpreted relative to that plane. The reference plane can be any of the three Cartesian planes (XY, YZ, ZX), or an arbitrarily positioned plane.

The grid also reflects the current scale, which is initially defaulted to 1/8" = 1' -0" (or 1: 100 for metric). It is displayed in a 3D axonometric view, which is initially defaulted to a z= 30° and x= 60° rotation.

Any of the parameters of the graphic environment of a modeling window can be changed by executing a menu command or one of the window tools found on the left side of the lower margin of the form-Z window (see section 2.2). Arbitrary reference planes can be defined and selected using the proper window tools (see subsections 2.2.1 and 2.2.3).

The parameters that control the graphic environment of a drafting window are the position of its 2D coordinate system, the scale, and the snapping switches.

When a drafting window is first opened, it is displayed with a two dimensional grid. The X direction is always horizontal and the Y direction is always vertical. The grid also reflects the current scale, which is initially defaulted to 1/8" = 1' -0" (or 1: 100 for metric).

Any of the parameters of the graphic environment of a drafting window can be changed by executing a menu command or one of the window tools. There are no tools in the drafting tool palette that affect the graphic environment of the window. The window tools are the same as those of the modeling windows, except that those performing a 3D function are unavailable. The window tools are discussed in section 2.2.
2.1.4 Communications between the modeling and drafting windows

The views of modeling scenes can be transported from a modeling window to a drafting window. Likewise, a drawing can be transported from a drafting window to a modeling window.

Entities may be transported either by first placing them in the form-Z Clipboard by executing the **Cut** or the **Copy** command, or they may be transported directly from one window to another. Both cases are executed using the **Paste From Drafting** or **Paste From Modeling** command, found under the **Edit** menu. The first command is available for the modeling and the second for the drafting windows. The plain **Paste** command is used to transport entities between windows of the same type (see subsection 3.2.2).

When an entity is moved between windows of different types, the internal structure of that entity is adjusted before it is brought into the new environment. Different options are available for transporting images of objects from a modeling window into a drafting window. They can be brought in as polylines, one polyline per outline, or they can be brought in as collections of segments. Which option is selected has implications for the type of editing that will be possible after the images have been brought into the drafting window. The desired option can be set in the **Paste From Modeling** dialog, which can be invoked from the respective item in the **Edit** menu (see section 3.2.2).
2.1.5 Dialogs

Dialogs and alerts on all platforms are movable and have a title bar. The name of the dialog appears in the title bar. Dialogs are invoked directly from the tools by double clicking on them, by menu items, or from buttons in other dialogs. They contain the options and/or parameters that affect the operation identified by their title.

Options and parameters are represented by check boxes, radio buttons, pop up menus, and alphanumeric fields, some of which may be complemented by slider bars.

A **check box** is typically represented by a little square that can be independently turned on or off by clicking on it. There is always a text string next to it that describes the activity that will take place when it is on.

**Radio buttons** are typically represented by little circles and are always in sets which are mutually exclusive. That is, only one of the buttons in a set can be on at any given time. They typically represent alternative actions that may be taken relative to an operation.

**Alphanumeric fields** are boxes where either characters or numbers can be typed using the keyboard. When they are numbers they may represent a variety of quantities. When they stand for percentages they can also be represented by slider bars with a little pointer that slides left and right or up and down, depending on the orientation of the bar. Slider bars typically accompany numeric fields and offer alternative means for setting a parameter.

**Pop up menus** are menus that are invoked by clicking on a box where the currently selected item is displayed. Their functionality is similar to the radio buttons and are typically used when the list of choices is relatively long.

A special dialog feature used by form-Z is the **animation bullets**. These are placed next to parameters that are animatable and, when they are on, the value of the respective parameter changes during an animation process. These bullets are optional and can be turned on or off from the **Project: Modeling: Animation** tab in the **Preferences** dialog, invoked from the **Edit** menu. The animation process and how to animate parameters of objects are discussed in Chapter 7 of the form-Z User's Manual.

![Figure 2.1.5.1: The Pick Options dialog.](image)
Most dialogs have a **Reset** button, in addition to the standard **OK** and **Cancel** buttons. When selected, this button resets the contents of the dialog to the state that they were in when the dialog was invoked. This is the same state that you would get if you canceled the dialog and then immediately re-opened it. The **Pick Options** dialog is shown in Figure 2.1.5.1.

The order of the standard dialog buttons matches the different Macintosh and Windows standards. That is, on Macintosh, the **OK** button is in the lower right corner, preceded by the **Cancel** and the **Reset** buttons. By default, on Windows, the **Reset** button appears in the lower right corner preceded by the **Cancel** and **OK** buttons. These defaults are shown in Figure 2.1.5.2. These settings can be changed by the **Order Of Buttons** options in the **System: Dialogs** category of options in the **Preferences** dialog (see section 3.2.8 in this manual.)

![Figure 2.1.5.2:](image)

A number of dialogs use tabs to organize their content and to keep their size more manageable. Each tab group has a number of sub groups of options identified by the tab titles at the top. A tab is selected by clicking on the tab title to make it active. When selected, the contents of the tab are shown in the area below the tab titles. Only one tab is active at a time. As an example the **Snap Options** dialog and the contents of its four tabs are shown in Figure 2.1.5.3.
Figure 2.1.5.3: The Snap Options dialog with the (a) Grid, (b) Angle, (c) Radial, and (d) Object tabs selected.

Even though the functionality is the same, the looks and style of the dialogs and their tabs vary depending on the operating system. This is illustrated in Figure 2.1.5.4 for the Vector Line Options dialog.
Figure 2.1.5.4: Dialog styles for (a) OS X, (b) Windows NT, and (c) Windows XP.
2.1.6 Preview dialogs and their viewing tools

Preview dialogs are used by a variety of form•Z features, such as the sweep operations and controlled rounding. All these preview windows include a set of tools that allow you to manipulate the viewing parameters.

Clicking on the TOP icon displays the top view of the image. Pressing the option key (Macintosh) or ctrl+shift (Windows) changes the icon to the BOT icon and clicking on it displays the bottom view of the scene.

Clicking on the RGT icon displays the right view of the scene. Pressing the option key (Macintosh) or ctrl+shift (Windows) changes the icon to the LFT icon and clicking on it displays the left view of the scene.

Clicking on the BAK icon displays the back view. Pressing the option key (Macintosh) or ctrl+shift (Windows) changes the icon to the FRT icon and clicking on it displays the front view.

Clicking on the AXO icon displays an axonometric view of the scene. When the dialog opens, this is the default view and is set to a 30°-60° axonometric.

Clicking on the FIT icon fits the image to the space available in the window.

The next three icons are for zooming, panning, and rotating the view. One of the three is always active and the respective operation is executed as soon as you click in the preview area. The active operation is also reflected by the icon of the mouse cursor which appears as soon as the cursor enters the viewing area.

When the zoom in icon is active, clicking in the preview area enlarges the view. Pressing the option key (Macintosh) or ctrl+shift (Windows) changes the cursor to the zoom out cursor. Clicking the mouse in the preview area shrinks the view.

The views are centered at the point where you click. Pressing the control key (Macintosh) or ctrl+alt (Windows) drags a rectangle from corner to corner around the area to be zoomed.

When this icon is active, clicking the mouse in the preview area and dragging it pans the view.

When the rotate view icon is active, clicking in the preview area and moving the mouse rotates the view about the center of the preview objects. Another click completes the rotation of the view.

These tools use the same shortcut keys that are assigned to the equivalent window tools.
This Arrow icon is available in many of the Preview windows, but not in all of them. It can be used to pick entities in the preview area and to act on them according to the semantics of the operation the preview window supports.

The last icon is the Render icon, which is available in many of the preview windows but not all of them. Clicking on it renders the image displayed in the preview area, using one of the rendering modes available in the form•Z version at hand. Which of the available display modes is used is determined by the selection from the pop up menu that is invoked from this icon, as shown in Figure 2.1.6.1.

Figure 2.1.6.1:
The menu that pops up from the Render icon of a preview window.
2.2 The window tools

The small icons located in the lower left margin of the form·Z window are the **window tool bar**. They are tools or operators that affect the graphic environment of that window. This tool bar is also available as a palette, which can be opened by selecting the **Window Tools** item from the **Palettes** menu (see section 3.9). All the icons in the window tool bar can be torn off to become floating tool palettes. You tear them off by dragging the mouse beyond their upper end after you pull them out. Note that when the iconic menus are popped up, they are shown with larger icons. The tool palettes can be positioned anywhere on the screen, their direction (horizontal or vertical) can be changed by clicking on the arrow box in their title bar, and they can be closed by clicking on their close box.

The window tools that are affected by dialog options are marked with a little dot at their upper right corner. These dialogs are invoked directly from the tools by double clicking on the icon or by clicking on the icon while pressing the **option** (Mac) or **ctrl-shift** (Windows) key. All the window tools can also be executed or selected from user-defined keyboard commands.

The window tools allow you to select the active reference plane, to lock graphic input to the perpendicular direction, to define and manipulate reference planes, to activate grid snapping, to select a directional or object snapping, to execute a zooming or panning operation, and to manipulate 3D views. Except for the first three and last one, the window tools apply to both the modeling and the drafting windows. Given their inherently 3D character, selecting, defining, or manipulating a reference plane, setting the perpendicular lock, and manipulating 3D views have no meaning in the 2D world of drafting. When a drafting window is active, these tools are dimmed and unavailable. Likewise, the tangent snap has no effect in modeling and is dimmed when a modeling window is active.

The window tools that require graphic input, namely Zoom ( ), Hand ( ), Zoom In/Out By Frame ( ), Set View ( ), Navigate View ( ), Walkthrough ( ), and Match View ( ) when selected, may cause the program to enter a mode that designates control to the window icons and makes the main tool palette inactive. This is indicated by graying out the tool palettes and happens when the **Continuous Window Tool Control** option is selected (default) in the **Preferences** dialog (invoked from the **Edit** menu). With this option you execute the Zoom, Pan, and View operations continuously without reselecting them. You exit this mode by clicking on another icon. If the option is off, you need to reselect these tools each time you apply them. The reference plane tools (third column) also require graphic input, but they are not affected by the continuous window tool control.
2.2.1 Switching the reference planes

In form\-Z, all graphic input is interpreted relative to a reference plane. One reference plane is always active, and only one reference plane can be active at any given time. The reference plane may be one of the Cartesian (orthogonal) planes, or it may be a plane positioned anywhere in the 3D world space.

Reference planes are only required and are only available in the modeling environment, where they can be changed by selecting one of the Set Reference Plane tools, whose torn off palette is shown in Figure 2.2.1.1. At start-up and when new projects are opened, the reference settings of the modeling windows default to the XY plane.

Associated with the reference planes are grids. Grids can be turned on or off through the Show Grid* item in the Windows menu. They can also be turned on or off from the Show Grid option in the Window Setup dialog, invoked from the Windows menu or by double clicking on one of the Set Reference Plane tools (Figure 2.2.1.2).

The grids are always positioned on their respective reference planes. They reflect the current scale and their structure is controlled by the user through parametric options that are available in the Window Setup dialog. The options in this dialog are discussed in detail in subsection 3.3.5.

The Set Reference Plane tools are complemented by the Reference Plane tools found in the third column of the window tools bar and are used for defining and manipulating reference planes.
XY Reference Plane

YZ Reference Plane

ZX Reference Plane

Selecting one of these window tools sets the active reference plane to the respective Cartesian plane. If the Show Grid option is on, the change is immediately reflected in the graphics window, as shown in Figure 2.2.1.3. Selecting a new reference plane has no effect on any previously generated modeling objects.

![Figure 2.2.1.3: (a) The XY reference plane. (b) The YZ reference plane. (c) The ZX reference plane.](image)

Arbitrary Reference Plane

When this tool is selected, the arbitrarily positioned plane is made the active reference plane (Figure 2.2.1.4). An arbitrary plane must be already defined before it can be selected as a reference plane. If no arbitrary plane definition exists, this tool has no effect. An arbitrary plane can be defined by using the Define Arbitrary Plane tool located in the third column of the Window Tools palette (see subsection 2.2.3).

![Figure 2.2.1.4: An arbitrary reference plane.](image)
2.2.2 Locking to the perpendicular direction

By default, all graphic input is interpreted on the active reference plane. When the mouse is used to draw or to represent motions, such as geometric transformations, these motions are interpreted in a direction parallel to the active reference plane. This mode of input interpretation can also be switched to a perpendicular direction.

Perpendicular Switch

This tool can be turned on or off by clicking the mouse on it. When it is turned off, the motions of the mouse have two degrees of freedom, which correspond to the axes of the active reference plane. When it is turned on, the motion of the mouse is locked only to the direction perpendicular to the reference plane.

For example, when the XY plane is the current reference plane and the Perpendicular Switch is off, only the X and Y coordinates are affected. When the Perpendicular Switch is on, only the Z coordinate is affected. When an arbitrary plane is the current reference plane, all three coordinates of the Cartesian space can be affected whether the switch is on or off. The resulting coordinate values are constrained to be parallel or perpendicular to the arbitrary plane, depending on whether the perpendicular switch is off or on.

The perpendicular behavior of the cursor takes effect as soon as the first click occurs for graphic input. Before the first click, the mouse cursor moves freely relative to the reference plane. After the first click, the mouse locks on a line that is perpendicular to the reference plane at the click point. Note, however, that none of this affects tools that must work on the reference plane, such as the drawing tools. In these cases the perpendicular switch is disabled automatically.
2.2.3 Reference planes

The tools contained in the third column of the Window Tools bar are used to define and manipulate reference planes. They complement the Set Reference Plane tools discussed in section 2.2.1. These tools are not used in the drafting environment and are dimmed when a drafting window is active. The Reference Planes palette is shown in Figure 2.2.3.1.

A plane is called arbitrary when it is not parallel to one of the Cartesian planes or it is specifically created by the user in some position in 3D space. Arbitrary planes are explicitly created through the Define Arbitrary Plane tool. They may also be automatically created by the program when operations applied to a Cartesian plane cause it to take a position that is not parallel to one of the Cartesian planes. When this happens, the Set Reference Plane tool (far left tool in the window palette) automatically switches to Arbitrary. The functions of the arbitrary reference planes are complemented by the Planes palette, where any number of named planes can be saved and recalled with a single mouse click. The Planes palette and its dialog are discussed in section 2.10.

Defining arbitrary planes

To define a new reference plane, three unique points are required. An arbitrary plane may be derived from three points of one or more objects that are currently displayed on the screen, from two segments, from an outline, or from a face. These entities may be picked before or after the Define Arbitrary Plane tool is selected. Arbitrary reference planes may also be derived as perpendicular planes to existing reference planes.

Define Arbitrary Plane

This tool derives an arbitrary plane by using the first three points of the entities that are selected. The origin of the arbitrary plane is determined by the position of the second point. The horizontal axis is determined from the first two points.

When the postpick method is used, the Define Arbitrary Plane tool is selected first. It is followed by the selection of three points, two segments, one outline, one face, or one whole object, depending on the currently active topological level. The derivation of the arbitrary plane is executed immediately and, if the Show Grid item (in the Window menu) is on, it is displayed on the screen. When the prepick method is used, three points, two segments, one outline, one face, or one object are picked first, then the Define Arbitrary Plane tool is selected, followed by a click of the mouse anywhere in the graphics window. Examples of arbitrary plane definitions are illustrated in Figure 2.2.3.2.

Whether points, segments, an outline, a face, or an object is picked is determined by the current topological level. The Define Arbitrary Plane tool is the only tool on the Reference Planes palette that is affected by a modifier.
When two segments are selected, the system uses the two points of the first segment and the first point of the second segment. The horizontal axis is placed on the first segment and the origin on the second point of that segment. When an outline or a face is used, the segment where the mouse is clicked when selecting the entity determines the position of the horizontal axis. When the two point pick option (Clicking On Edges) is active, the first click of the mouse determines the segment that will be used for the axis. When the one point pick option (Clicking Inside Boundaries) is on, the segment that is closest to the click position determines the position of the horizontal axis. When a complete object is selected, the plane is derived from its first face by placing the horizontal axis on the first segment of the face.

Figure 2.2.3.2: (a) Given a hexagonal extrusion that has been rotated, arbitrary planes are defined by picking (b) three points (b1, b2, and b3), (c) two segments (c1 and c2), (d) a face (d1, with Clicking Inside Boundaries option on), and (e) the complete object (e1).

Define Perpendicular Plane

To define a plane perpendicular to the current reference plane, you click on one of the axes that are on the plane. Clicking on the axis that is perpendicular to the plane produces no result. The new perpendicular plane appears immediately (Figure 2.2.3.3).

Figure 2.2.3.3: Selecting the Define Perpendicular Plane tool and clicking on point 1 creates a perpendicular reference plane.
Moving reference planes

Reference planes (both orthogonal and arbitrary) can be moved along their surfaces or in a perpendicular direction. Their axes can also be moved and rotated around one of their axes.

Move Plane

To move any plane, select the Move Plane tool and click on the reference plane. This causes a bounding box to be rubber banded (Figure 2.2.3.4(a)). One more click positions the plane, whose grid is regenerated. After the move, two sets of axes are shown (assuming the respective items are on), as in Figure 2.2.3.4(b). If the Perpendicular switch (window palette) is on, the reference plane moves perpendicular to its surface (Figure 2.2.3.5).

Move Plane Origin

To move the origin of a reference plane, select the Move Plane Origin tool, then click and drag the reference plane’s origin. The axes are rubber banded and follow the mouse’s motion until the second click completes the operation (Figure 2.2.3.6).

Rotate Plane

To rotate a reference plane around its perpendicular axis, click on the plane and move the mouse. The plane is rubber banded and the next click completes the rotation and refreshes the grid (Figure 2.2.3.7(a)). To rotate a plane around one of the axes that lie within the plane, click on the other axis and start rotating (Figure 2.2.3.7(b)). The next click completes the operation.

Figure 2.2.3.4:
(a) Moving a reference plane.
(b) After the move, two sets of axes are shown.

Figure 2.2.3.5:
Moving a reference plane with Perpendicular switch on.

Figure 2.2.3.6:
Moving the origin of a reference plane.

Figure 2.2.3.7:
Rotating a reference plane (a) about its perpendicular axis and (b) about an axis lying on it.
Resizing the grid

When new projects are started, the modeling reference grids have a default size of 64' x 64', or 32 feet in each direction. In metric, the default size is 20 x 20 meters, or 10 meters in each direction. Reference grids can be resized using the Extend Grid tool. When additional associative windows are opened, their grids are the same size as the size of the active window.

**Extend Plane Grid**

This tool allows you to extend the boundaries of the current reference grid. They are extended one segment at a time. This tool requires two mouse clicks.

The first click picks the boundary segment that will be moved, while the second sets the destination position of the boundary segment. The grid is updated immediately and the new version is displayed on the screen. The process is illustrated in Figure 2.2.3.8.

The reference grids of a window (XY, YZ, ZX, arbitrary) are independently sized, and are also independent of the sizes in associated windows. Note that the reference grid does not define the working area and you can draw beyond its boundaries. It is simply a reference system.

If you click with this tool anywhere in the window while pressing the *shift* key, the grid boundaries are automatically repositioned to the ends of the objects plus the size of the major grid module, as defined in the **Window Setup** dialog. An example is shown in Figure 2.2.3.9.

**Resetting the reference planes**

**Reset Plane**

This operation resets any of the Cartesian reference planes to their default positions. It has no effect on arbitrary planes. To execute it, with the Reset Plane tool active, click in the project window.
2.2.4 Snapping to grid points

Input points can be rounded to preset modules, which is commonly known as the **snapping to grid** option. When this option is used, the input point appears to jump to the points of a usually invisible rectangular grid, which is defined by the current grid module increments. In form·Z, the snapping grid is independent from the reference plane grid, but the two can also be made coincident by assigning appropriate values to their parameters.

The grid snapping parameters are set in the **Grid Snap Options** dialog (Figure 2.2.4.1), invoked directly from the Grid Snap switch.

**Grid Snap Module**: The module values in the X, Y, and Z fields can be set independently. However, the three directions can also be constrained to the same increment by selecting the **XYZ Snap Lock** (default).

**Match Grid Module**: When this option is on, the snapping modules are set automatically to increments of the current reference grid definition.

**All Windows**: When this option is selected, the grid snap parameters are simultaneously changed in all associated windows of the current project. This item is dimmed when there is only one window open.

**Grid Snap**

This switch turns the grid snap option on or off. The effects of the grid snap are illustrated in Figure 2.2.4.2. Changing the state of the Grid Snap switch has no effect on previously generated objects.

![Grid Snap Options dialog](image)

*Figure 2.2.4.1:* The **Grid Snap Options** dialog.

![Figure 2.2.4.2: The effect of Grid Snap](image)

*Figure 2.2.4.2:* The effect of Grid Snap:
(a) on and snap module at 2' for all directions;
(b) on and snap module at 4';
(c) on and snap module at 8';
(d) grid snap is off.
2.2.5 Directional snapping

Directional snapping constrains the position of each input point relative to its previous point. Since a direction is defined between two points, the directional snap does not affect the first point of a point sequence, nor does it apply when a single point is entered. All directional snapping may be turned off, which is the default. In form\textbullet\textsc{Z}, directional snapping can be applied in the horizontal, vertical, and diagonal (45°) direction, in any arbitrary slope, or at angular increments.

Directional snapping is set with one of the Directional Snap tools whose torn off palette is shown in Figure 2.2.5.1.

- **No Direction Snap**

This tool turns all directional snapping off and is the default setting for new windows.

- **Orthogonal Snap**

This tool restricts the directional snapping to the vertical and horizontal directions. The system automatically figures out which of the two applies, by comparing the horizontal distance traveled with the vertical distance. Here, the terms “horizontal” and “vertical” are relative to the active reference plane, and each can mean any of the X, Y, or Z directions. The horizontal and vertical snapping is illustrated in Figure 2.2.5.2.

- **Orthogonal/Diagonal Snap**

This icon sets the directional snapping to the vertical, horizontal and diagonal (45°) directions. This functions as a combination of the Orthogonal Snap and the Angle Snap tools, when the latter is set to a 45° slope. This snapping switch is illustrated in Figure 2.2.5.3.
Angle/Slope Snap

This tool sets the directional snapping to a user defined angle, which is set in the **Angle Snap Options** dialog (Figure 2.2.5.4) accessed directly from the Angle/Slope Snap tool. The direction can be defined either as an **Angle**, or a **Slope** that is entered as a horizontal/vertical ratio. When one of the two methods is used, the other field is automatically updated and shows the same direction. The default direction is 1/1 or 45°. The **All Windows** option is as with Grid Snap. Snapping to an angle is illustrated in Figure 2.2.5.5.

Radial Snap

This tool causes the graphic input to snap to positions defined by an angle and, optionally, by a distance. These parameters are set through the **Radial Snap Options** dialog (Figure 2.2.5.6), which is accessed directly from the tool. Radial snapping corresponds to radial coordinates. However, it applies independently of whether or not radial coordinates are selected in the Prompts or Coordinates palette.

Two alternative options are available for defining the angle in the **Radial Snap Options** dialog. You can enter a value for degrees in the **By Angle Of** field, or you can define it as a fraction of the circle by entering a number in the **By n Radial Divisions** field. To also apply distance snapping, you select the **Distance Snap** option and enter a value in its field. Again, the **All Windows** option is as with the other snaps.

Note the major difference between angle/slope and radial snapping. The former essentially restricts the motions of the mouse to two perpendicular directions. The latter restricts the motions to angular increments centered about the previous point. The number of directions allowed depends on the parameters entered in the **Radial Snap Options** dialog. Radial snapping is illustrated in Figure 2.2.5.7.

---

**Figure 2.2.5.4:** The **Angle Snap Options** dialog.

**Figure 2.2.5.5:** Angle/slope snaps.

**Figure 2.2.5.6:** The **Radial Snap Options** dialog.

**Figure 2.2.5.7:** Radial snapping.
2.2.6 Object snapping

Object snapping adjusts the points selected by the mouse cursor to coincide with a part of an object's topology. The object snaps are mutually exclusive, except when Combination Snap is on (see below). Object snapping can be used in conjunction with the directional and grid snaps. Object snap is available in both modeling and drafting. Their palettes are shown in Figure 2.2.6.1.

When one of the object snaps is selected, the graphics cursor changes to the object snap cursor ( ), indicating that an object snap is active. When the cursor is close to an entity of the type to which the snapping is set, the cursor changes to the snap lock cursor ( ) indicating that the cursor is currently snapping to an entity.

The Object Snaps Options dialog (Figure 2.2.6.2) can be invoked directly from all the object snap icons (except None) to set parameters that affect object snapping.

Show Snap Preview: This option controls whether the entities to which the mouse snaps will be highlighted. It is on by default. The snap highlight color can be set through the Project Colors dialog accessible from the Options menu. It is green by default.

Extended Intersections: This option controls whether, when Snap to Intersection is on, snapping will occur only to intersection points that are between the end points of intersecting segments, or also to intersection points that are beyond the ends of the segments.

Tolerance: This field controls the distance within which the mouse must be positioned in order to snap to an entity.

Figure 2.2.6.1: The Object Snap palette.

Figure 2.2.6.2: The Object Snaps Options dialog.
Midpoint snapping for arcs

When midpoint snapping is applied to smooth arcs, the snapping behavior is determined by the Snap To Midpoint group of options. Circles are effectively closed arcs and are also affected by these options.

**Midpoint Of Edge**: When this option is on, the Midpoint snap snaps to the midpoint of the circumference of the arc. This is the default.

**Midpoint Of Arc Chord**: When this option is on, the Midpoint snap snaps to the midpoint of the arc chord which is the line connecting the ends of the arc. When applied to a circle, the Midpoint Of Edge option is used instead.

**Center Of Arc**: When this item is active, the Midpoint snap snaps to the center of the arc or circle.

The three options are illustrated in Figure 2.2.6.3.

Setting segment parts

**Snap to Segment Part**: The options in this section of the dialog are valid when the Snap To Segment Part tool is being used. The exact positions of the snaps are determined by the two options within the section.

**Interval Snap with n Divisions**: When this option is selected, the cursor snaps to the endpoints of a segment and to points on the segment derived by dividing the segment by \( n \), the number entered in the numeric field. In drafting, when snapping to arcs (both complete circles and partial arcs), the whole arc is taken as a single segment when calculating the divisions by \( n \).

**Proportion Snap \( n\% \) From Ends**: When this option is selected, snapping occurs to the endpoints of a segment and to two points between the ends, which are located at \( n\% \) distance from each of the ends. The ratio \( n \) can take a value between zero and 50\%. In drafting, when this snap is used, arcs are considered a single segment. However, this snap does not apply to closed arcs (circles) because there are no endpoints.

Setting angular snaps

**Snap To Perpendicular/Angular**: The options in this section of the dialog are valid when the Snap To Perpendicular/Angular tool is being used. The direction of the snap may be perpendicular to or at an angle from the snap segment depending on which option is selected.

**Perpendicular Snap**: When this option is selected, the current cursor point is snapped to a line perpendicular to the segment on which the previous snap occurred. If the previous (first) input point was not snapped, the current (second) input point may be snapped. In general, both points cannot be snapped, except in special drafting cases where one point may be snapped to an arc and the other to a segment.
Angular Snap At $n$ Degrees: When this option is selected, the input point is adjusted as for the perpendicular snap, except that the snap line is at an angle of degrees $n$ from the line to which the previous or the current point is snapped to. The value $n$ is entered in the numeric field of this option.

Snapping in projections and 3D views

Projection Views: This group of options determine where shapes are drawn when working in an orthographic projection view and one of the object snaps is used. This clarification is necessary since many points often have different positions in 3D space, but project as a single point when they are on a line perpendicular to the projection plane. It is frequently desired that the shape being drawn while snapping lies on the same plane. The options in this group essentially cancel the third coordinate that corresponds to the axis perpendicular to the projection plane, when snapping to 3D entities. This allows flat shapes to be drawn.

Lock Drawing To First Point: When this option is selected, the position of the plane where a shape will be drawn is determined by the position of the first point you draw. This plane is parallel to the reference plane and passes through the first point drawn. All subsequent points we draw while we snap, take two of their coordinates from the entity to which we snap. The third coordinate is on the drawing plane that was determined by the first point. If we do not snap to a particular point, all the coordinates of the point we draw project to the plane of the first point. Selection of this option produces a flat drawing. If this option is not selected, the points of our drawing will be at the exact 3D positions of the entities to which you snap, possibly resulting in a non-planar drawing.

Project Onto Reference Plane: When this option is selected, all the points we draw are again placed on the same plane, which is the active reference plane. That is, all input points are projected and drawn on the reference plane.

Depth Snap: When working in a projection view with object snaps and there are multiple entities projecting as a single one, this pair of options determines the one to which we will snap.

Use Closest: When this option is selected, we snap to the entity closest to the viewer.

Use Farthest: When this option is selected, we snap to the entity farthest from the viewer.

3D Views: This group of options determines where shapes are drawn when working in a 3D view and one of the object snaps is used. The two options it contains are identical and work exactly as for the orthographic projection views (see above). They are repeated so that independent settings may be used for projection and 3D views.
The object snap tools

- **No Object Snap**
  When this tool is selected all object snapping is turned off. This is the default setting for new windows.

- **Snap to Point**
  When this snap is selected, the cursor snaps to the closest point of an object.

- **Snap to Endpoint**
  When this snap is selected, the cursor snaps to the nearest endpoint of the segment that is closest to the center of the cursor.

- **Snap to Midpoint**
  When this snap is selected, the cursor snaps to the midpoint of the segment that is closest to the center of the cursor. When smooth arcs are involved, the snapping behavior is determined by the **Snap To Midpoint** group of options.
**Snap to Key Point**

Key points are points of an object that are critical to the shape or the controls of the shape. Examples of key points are the centers of circles and the control points of spline or nurbz objects (Figure 2.2.6.4). The key points available per type of object are listed in Figure 2.2.6.5. Key points can be shown on the screen in interactive render mode by selecting **Show Key Points** from the **Interactive** tab in the **Display Options** dialog (see section 3.6.2 in this manual).

Selecting the Snap To Key Point icon causes the cursor to snap to the key points of an object.

**Figure 2.2.6.4:** Examples of key point on a (a) primitive and (b) nurbz object.

**Figure 2.2.6.5:** The key points of different types of objects.
Snap to Segment Part

When this snap is selected, the snapping occurs to the endpoints and to certain positions between the ends of segments. The exact positions of the snaps are determined by one of two methods, which are selected from the Snap To Segment Part group of options in the Object Snaps dialog.

Snap to Segment

When this snap is selected, the cursor snaps to the closest point along the segment that is closest to the center of the cursor.

Snap to Line

When this snap is selected, the cursor snaps to the closest point along the line that is closest to the center of the cursor. Recall that, contrary to the finite nature of a segment, a line is infinite. The line snap is essentially identical to the segment snap except that it is not limited by the endpoints of the segment.

Snap to Intersection

When this snap is selected, the cursor snaps to the closest intersection of two segments that are the closest to the center of the cursor. This snap is applied to true intersections only. In the modeling environment, when the views of two segments intersect does not necessarily imply that the segments themselves intersect. This snap is affected by the Extended Intersections option selected from the Object Snap dialog.

Snap to Perpendicular/Angular

When this snap is selected, snapping occurs in a direction relative to the segment to which we snap. This direction may be perpendicular to or at an angle from the snap segment, which is selected from the Snap To Perpendicular/Angular group of options in the Object Snaps dialog.

The perpendicular and angular snaps are directional snaps relative to snapped entities. The snapped entity may be determined by either the first input point or the second, and in a few special cases in drafting by both. The following variations are available when using perpendicular or angular snaps to draw a line.

• The first input point may or may not be snapped to a segment.
• If the first point is snapped, the second point is locked to a direction perpendicular to (or at an angle from) the segment where the first point was snapped.
• If the first point is not snapped, then the second point may or may not be snapped to a segment. That is, as soon as the cursor is close to a segment, the second input point jumps to a location on that segment to which a perpendicular line can be drawn from the first input point.
• When the first input point is snapped, the **option** key (Macintosh) or **ctrl+shift** (Windows) can be used to partially free the constrains that the snapping imposes on it. That is, the input line continues to be perpendicular to (or at an angle from) the segment to which the first point was snapped, but follows the motion of the mouse, during rubber banding. Its position is determined by the second point as soon as we click to enter it. During rubber banding, the first point slides along the snap segment as necessary, so that the perpendicular (or angular) constraint may be preserved. Whether **Extended Intersections** is on or off does make a difference relative to how the snap behaves when the drawn line is taken beyond the ends of the snap segment.

• When drawing vector lines (continuous sequences of segments), only the first segment can be snapped; for subsequent segments the perpendicular/angular snapping behaves as snap to segment does. The free (second) point attaches to the segment, but does not attempt to place itself at a certain angle, since, such a condition is not generally achievable.

• In drafting, when one endpoint of the line we draw is on an arc (or circle) and the other is on a segment, it is geometrically possible to simultaneously snap on both, and form-Z does so. After snapping the first point, as the second point is rubber banded, the position of the first is continuously adjusted until we click to enter the second point.

**Snap to Tangent**

This snap can be used to draw lines that are tangent to arcs, both partial and complete (circles). It has no effect when the snap entity is a segment. The tangent snap works exclusively with edges of smooth objects. When selected, during rubber banding, the current cursor point snaps to the tangent of the arc element on which the previous (first) point was snapped. If the previous input point was not snapped, the current (second) input point may or may not be snapped. When it is, as soon as the mouse is close enough to an arc, the input point jumps to a position on the arc that satisfies the tangency condition.

In general, tangent snapping works as the perpendicular/angular snapping does. When using any combination of perpendicular, angular, and tangent snaps in drafting with the **option** (Macintosh) key or **ctrl+shift** (Windows) pressed, it is possible to adjust both endpoints of the first segment of a vector line, and up to three input points of a circle or arc.

It is also possible to do a three-way snap, such as drawing a circle that snaps to three other circles or smooth curves, as illustrated in Figure 2.2.6.6(e). This is done by pressing the **option** (Macintosh) or the **ctrl+shift** (Windows) keys while drawing.

![Figure 2.2.6.6](image)

**Figure 2.2.6.6:** Drawing lines with Tangent Snap on: (a) Start at tangent point P1, end at P2. (b) Start at free point P1, end at tangent point P2. (c) Start at tangent point P1, end at tangent point P2. (d) Start and end at tangent points of splines P1 and P2. (e) Draw a 3-point circle by clicking on P1, P2, and P3, while pressing **option** (M) or **ctrl+shift** (W).
Snap to Center of Face
When this snap is selected the cursor snaps to the centroid of the front facing face that is the closest to the center of the cursor. This snap is available in drafting under the name Snap to Center, and behaves like its modeling counterpart.

Snap to Face
When this snap is on, graphic input snaps to the plane of the face that is closest to the cursor. If more than one face overlaps, it snaps to the front most face. This snap is not available in drafting.

Combination Snap
When this icon is on, form-Z will simultaneously snap to more than one entities. The combination is set in the Combination Snap Options dialog (Figure 2.2.6.7), which is invoked from the tool icon. It contains 12 object snap icons that can be combined. Recall, however, that the face snap is dimmed when a drafting window is active. Clicking on an icon inverts it, indicating it is selected. Clicking again deselects it.

Any number of the snap icons can be turned on at the same time. When this is done, a hierarchy is established that favors point type snaps over line type snaps and line type snaps over area snaps. For example, when midpoint snapping (point type) and segment snapping (line type) are selected and the cursor is placed over the midpoint of a segment, the midpoint is snapped to. Only when the cursor is moved out of the tolerance zone around the midpoint is the segment snapped to. Likewise, when segment snapping and face snapping (areatype) are on at the same time, the cursor is snapped to the segment first, and only when it is moved out of the tolerance zone around the segment is it snapped to the surface of the face. Point type snaps are: Point, Endpoint, Midpoint, Key Points, Segment Part, Intersection, Perpendicular/Angular, Tangent and Center of Face. Line type snaps are: Segment and Line. The only area type snap is: Face. The three icons that are on by default are shown in Figure 2.2.6.7.

The remaining options in this dialog are as the same options in the Object Snap Options dialog, shown in Figure 2.2.6.2.
Drawing in 3D space with snaps

The object snaps allow you to conveniently draw and generate objects by referring to parts of other previously created objects. This effectively allows you to draw directly in 3D space as opposed to a flat reference plane. Some examples are illustrated in Figure 2.2.6.8.

Figure 2.2.6.8: Using the object snaps in 3D: (a) The Circle by Diameter drawing tool is used in conjunction with the Snap to Midpoint to construct a cylinder directly on top of the cuboid. (b) Given two rectangular extrusions, the Rectangle drawing tool is used together with the Snap to Point to construct a slant extrusion. Note that the extrusion is perpendicular to the reference plane. (c) A rectangle is drawn as in previous examples, but the extrusion now is perpendicular to the surface of the rectangle. This is done by selecting the Perpendicular To Surface option in the Extrusion/Convergence Options dialog.
2.2.7 Zooming and panning

Zooming enlarges or reduces the area of a screen image without changing the drawing’s current scale. Enlarging an area, or **zooming in**, is useful when inspecting or working on close up details. Reducing a drawing, or **zooming out**, is useful when it is necessary to inspect the complete picture, rather than just a part of it.

**Panning** is moving an image within the screen without permanently affecting the positional parameters of the entities it represents. In general, the zooming and panning operations are tentative. You typically want to return the drawing to its initial state once the set of operations that necessitated the zooming or panning have been completed.

As implemented in **form-Z**, zooming does not change the center of interest and eyepoint. Instead, the area is enlarged to fit the screen. When this happens, the cone of vision remains unchanged. To illustrate, if we think of the cone of vision as representing a printed photograph, we can think of zooming as cutting out a rectangular portion of the photograph and enlarging it to the size of the original print. This also implies that, to be able to zoom a view out, it should first be zoomed in. This is again analogous to viewing a photograph. That is, it is not possible to see areas in a photograph which are outside of the view when the photo was taken.

The zoom factor of a view is indicated in the Zoom Percentage pop up menu at the bottom of the project window, next to the horizontal scroll bar (see subsection 2.2.9). For example, 100% zooming indicates that the view is not zoomed, while 200% indicates that the view has been enlarged by a factor of 2.

Panning using the hand tool or scroll bars in perspective behaves in a similar fashion. If the perspective view has previously been zoomed in, panning will move the zoomed rectangle inside the unzoomed cone of vision. Depending on how far the view is zoomed, the panning may move the visible area only a small distance before the edge of the cone of vision is reached and panning stops. However, unlike zooming, panning will move the view’s eye and center of interest horizontally or vertically in the picture plane if the view is not zoomed in.

As implemented in **form-Z**, up to 30 zooming/panning operations can be undone. The Go to Previous tool can be executed repeatedly to return to previous views of a drawing or the Reset tool can return the view of a drawing to its initial state.

The torn off Zoom palette of **form-Z** is shown in Figure 2.2.7.1. The zoom and pan tools apply to both drafting and modeling, where they apply to all the view types, including perspective.

![Figure 2.2.7.1: The torn off Zoom palette](image)
Four methods of applying zooming are available.

- The first zooms in or out incrementally relative to where the mouse is clicked with the Magnifying Glass tool active (Zoom In/Out about Point).
- The second allows you to precisely outline the area that will be zoomed in or out (Zoom In/Out by Frame).
- The third automatically applies a constant factor and zooms (in or out) from the center of a window (Zoom In/Out Incrementally).
- The fourth method is similar to the first, but is executed using the mouse scroll wheel, rather than a tool, whenever such a wheel is available on a multi-button mouse. This is an incremental zoom about the position of the cursor. Rolling the scroll wheel up zooms in and rolling it down zooms out.

Each of these methods is controlled by options selected from the Zoom Options dialog (Figure 2.2.7.2) that is invoked directly from the zoom and pan tools or from the Options menu (see subsection 3.7.1).

The top two options in the Zoom Options dialog affect the Zoom In/Out by Frame tools and allow you to select the manner in which the rectangular frame used for determining the zoom area is drawn and rubber banded. **Window Zoom By Center** (default) draws from center to perimeter and **Window Zoom By Corners** draws from one corner to its opposite corner.

The next three options affect the Zoom about Point and Zoom Incrementally tools.

**Zoom In By** and **Zoom Out By**: The values entered in these fields control the scaling factors used for the execution of the incremental zooms. Default factors are 133% for in and 75% for out.

**Keep In/Out Proportional**: When this option is on and one of the factors above is changed, the other is automatically adjusted to the inverse value. That is, a scaling factor such that the sequential execution of a zoom in and a zoom out preserves the original scale of the image. When the option is off the in and out factors can be set independently.
Fit All: This group of parameters affects the Fit All operation and set how much border will be introduced between a scene and the boundaries of the graphics window, when the size of the scene is adjusted to fit the size of the window.

Fit Border: The value entered in this field is a percentage used to reduce the size of the window before executing the fit. When

Include Lights: When this option is selected, the lights will also be taken into consideration when fitting the content of the project.

Include Views/Camera: When a camera is displayed and this option is on, the camera’s graphic representation will also be included when fitting.

Selected Entities Only: When this option is on, the fit operation will only take into consideration the selected entities.

If none of these options is selected, the fit will by default only be based on the visible, unghosted objects in the scene.

Snap When Using Scroll Wheel: When the scroll wheel is used to zoom in or out in the graphics window, the location of the cursor is used as the zoom center. When this option is enabled along with one of the snapping modes (grid, direction, or object), the current snap location is used as the zoom center. This makes it easy to zoom in or out on a snapped location on the screen or relative to a part of an object. This option is off by default.
**Zoom In/Out about Point (Magnifying Glass)**

Selecting this tool without pressing any key shows a magnifying glass with a plus sign. Clicking on the window enlarges the image about the click point, which is placed at the center of the screen. Pressing `control` or `ctrl+alt` allows you to drag a rectangle, from corner to corner, and delineate the area to be zoomed.

The zoom in scaling factor is set by the *Zoom In By* parameter in the *Zoom Options* dialog. Its default is 133.333%. An example is shown in Figure 2.2.7.3.

When you select this tool while pressing `option` (MacOS) or `ctrl+shift` (Windows), the sign in its icon changes to a minus. Clicking with this tool while the minus sign is displayed decreases the size of the image on the screen, using the scale factor set in the *Zoom Out By* parameter in the *Zoom Options* dialog. Its default is 75.000%. An example is illustrated in Figure 2.2.7.4.

**Hand**

Selecting this tool changes the cursor to a hand icon and two panning variations can be executed with it, which is set in the *Hand Options* dialog (Figure 2.2.7.4) invoked from the tool.

- **Dynamic Click And Drag**: With this option on, clicking and dragging on the window moves the displayed image in the direction you drag (Figure 2.2.7.5), which can be in any direction, contrary to the scroll bars, which can only pan horizontally or vertically.

- **Pan Point To Point**: With this option on, you click twice. After the first click a line is rubber banded. After the second click the pan is executed and the image move by the length of the line you drew with the pan tool. Again, the pan can be in any direction.

*Figure 2.2.7.3:* Zooming in with the Magnifying Glass tool.

*Figure 2.2.7.4:* The *Hand Options* dialog.

*Figure 2.2.7.5:* Using the Hand tool to move the image within the graphic window.

*Interface • The window tools*
**Zoom In by Frame**

With this tool, you draw a rectangular frame that bounds the area you want to enlarge. You draw the frame by clicking the mouse at the center of the area (assuming the default **Window Zoom By Center** option is selected). A rectangular shape is rubber banded and starts to grow as you drag the mouse away from the center, while holding the button. The proportions of the rectangle are locked to the proportions of the window. Releasing the mouse button completes the zoom by enlarging the area in the frame to fill the window. If the **Continuous Window Tool Control** option is selected in the Preferences dialog, any number of Zoom In by Frame operations can be executed sequentially without requiring you to reselect its icon. The Zoom In by Frame operation is illustrated in Figure 2.2.7.6.

![Figure 2.2.7.6: Zooming In by Frame twice.](Diagram)

**Zoom Out by Frame**

With this tool, you draw a rectangular frame that is used as the base of the reduction. The frame is drawn as for the Zoom In by Frame operation and reduces the area in the window to fit in the frame. The operation is the reverse of the Zoom In by Frame operation and is illustrated in Figure 2.2.7.7.

![Figure 2.2.7.7: Zooming Out by Frame twice.](Diagram)
**Zoom In Incrementally**

This tool applies a preset scaling factor to the image shown in the graphics window. The zoom in scaling factor is set in the **Zoom Options** dialog. Its default value is 133%. The zooming operation is applied relative to the center of the window. Any number of sequential incremental zooms can be applied by selecting the Zoom In Incrementally tool repeatedly, as illustrated in Figure 2.2.7.8.

![Figure 2.2.7.8: Zooming In Incrementally twice.](image)

**Zoom Out Incrementally**

The Zoom Out Incrementally tool works as the Zoom In Incrementally tool, except that it applies a different scaling factor (default 75%). The Zoom Out Incrementally operation is illustrated in Figure 2.2.7.9.

![Figure 2.2.7.9: Zooming Out Incrementally twice.](image)
Go to Previous

You use this tool to go back one step to the previous zooming/panning, if any. As zooming or panning operations are applied, form•Z saves their settings and can return to them, when the Go to Previous tool is selected. Up to 30 sequential applications of the Go to Previous operator are allowed, or until the list of zoom/pan parameters is exhausted. When that list is empty and you attempt to execute a Go to Previous operation, the system beeps, issues an error message, and cancels the operation.

Fit All

With this tool the size of the image is adjusted so that all active objects or elements fit within the graphics window. To execute the operation click on the tool. The window is completely filled except for a border space around its perimeter. The size of the border space is controlled by the Fit Border parameter in the Zoom Options dialog, discussed earlier in this section. Additional options in this dialog determine whether lights and views/cameras will also be taken into consideration when fitting a scene. The Fit All operation is illustrated in Figure 2.2.7.10. Note that it is meaningless to execute this operator more than once.

Figure 2.2.7.10: (a) Before and (b) after Fit All.

Reset

This tool returns your window to its initial state of the zooming/panning parameters and clears the zoom/pan list.
2.2.8 Viewing

The right most icon in the window palette contains four tools for manipulating 3D views. The torn off palette of the View tools is shown in Figure 2.2.8.1.

Form-Z offers a variety of methods for manipulating 3D views. Most of them are executed through items in the View menu, from which the cone of vision environment can be accessed. The four viewing tools in the window tool palette offer quick interactive methods for manipulating views.

If the Continuous Window Tool Control option is selected in the Preferences dialog (invoked form the Edit menu), the View operations can be applied any number of times continuously without requiring you to reselect their tools.
Set View

To use this tool in *modeling*, after selecting it:

- Click the mouse in the project window and move it horizontally or vertically, with or without a modifier key. As you move the mouse, the following operations are executed:
  - moving right, view is rotated counterclockwise about the vertical axis;
  - moving left, view is rotated clockwise about the vertical axis;
  - moving up, view is rotated clockwise about the horizontal axis;
  - moving down, view is rotated counterclockwise about the horizontal axis;
  - moving in other directions combines a horizontal and vertical rotation;
  - in non-perspective views: pressing `option` (Macintosh) or `ctrl+shft` (Windows) and moving up, pans the scene down in the graphics window;
  - in non-perspective views: pressing `option` (Macintosh) or `ctrl+shft` (Windows) and moving down, pans the scene up in the graphics window;
  - in perspective views: pressing `option` (Macintosh) or `ctrl+shft` (Windows) and moving up, moves closer to the scene;
  - in perspective views: pressing `option` (Macintosh) or `ctrl+shft` (Windows) and moving down, moves farther away from the scene.

In all cases of rotation, the view is rotated about the vertical axis that is at the *Center Of Interest* of the current view, *Center Of Scene* (default), or at the *Click Point On Reference Plane*. These options can be selected from the *Set View Options* dialog (Figure 2.2.8.2) invoked from the Set View tool. When editing the view in wire frame display mode, complete objects are shown, while bounding boxes are used for the other display modes.

In *drafting*, to use the Set View tool, after selecting it:

- Click the mouse in the project window. This rubber bands a line drawn from the center of the window to the position of the mouse, as the mouse moves. As in modeling, the center of the window is the center of rotation.

- Move the mouse about the center and observe the 2D draft window rotating like a piece of paper spinning on a table about a single point.

- When the view is where you want it, click again to complete the operation.
**Navigate View**

This tool, which is only available in modeling, allows you to change many of the view parameters interactively. This is in contrast to the Set View tool, which is only able to rotate the view and zoom.

To use this tool, select the Navigate View icon, which causes the navigation marquee (Figure 2.2.8.3), to appear. It consists of four circles and is used to control a variety of view parameters, depending on the area of the marquee you click on to select it. The circles are referred to as first or inner, second, third, and fourth or outer. Navigation involves the following steps:

- Click on an area of the marquee to select a view operation and to start action.
- Move the cursor. As the mouse is repositioned, the view is updated.
- Click again to complete the view change.

If the continuous mode is active, the view manipulation process is completed by double clicking on the marquee, selecting another icon, or by pressing the `esc` key. If the continuous mode is off, the operation terminates after the second click that sets the view.

The following view manipulations can be executed with the Navigate View tool.

**Pan:**

This operation moves the eyepoint and center of interest along a plane perpendicular to the sight.

To activate this operation, place the cursor between the inner and second circles of the marquee. This causes the cursor to change to;

- , if away from the horizontal and vertical axes of the marquee;
- , if positioned close to the horizontal axis; and
- , if positioned close to the vertical axis.

In the latter two cases panning is locked to the horizontal or vertical directions, respectively.
Depth pan: 

This operation moves the eyepoint and center of interest along the line of sight. This has the effect of moving into or out of the scene.

To activate this operation place the cursor in the small inner circle, at the center of the marquee. This causes the cursor to change to 
. After the initial click the view is dynamically adjusted as the cursor moves up and down. Moving up pans inwards and moving down moves outwards. Left and right motions have no effect. This operation is illustrated in Figure 2.2.8.4.

Figure 2.2.8.4: Depth Panning with the Navigate View tool. (a) Clicking on 1 and 2 (b) produces the shown result.

Rotate view: 

This operation rotates the eyepoint and/or the center of interest about the line of sight. It is similar to the Set View tool, except that the latter only rotates the eyepoint around the center of interest.

To activate the Rotate View operation, position the cursor between the second and outer circles. This changes the cursor to:

, if it is positioned away from the horizontal and vertical axes;
, if close to the horizontal axis; and
, if close to the vertical axes.

In the latter two cases the rotation locks to the respective directions.

The position of the first click relative to the outer and second circles is significant.

• Clicking close to the outer circle rotates the eyepoint, while the center of interest remains fixed.
• Clicking close to the second circle rotates the center of interest, and the eyepoint stays fixed.
• Clicking at a point between the outer and second circles causes both the eyepoint and center of interest to rotate. The rotation is about a point along the line of sight, whose position is set by where the cursor is clicked relative to the outer and inner (second) circles. The middle (third) circle represents the midpoint of the line of sight. Clicking on it causes both the eyepoint and center of interest to rotate equally. Clicking closer to the outer/inner circles produces analogous results.
Spin: 🔄

This operation changes the tilt of a view.

To execute it, position the cursor on either of the Spin icons (🔄), located at the outer ends of the horizontal axes of the marquee. This changes the cursor to the same icon to indicate that the Spin operation is active.

After the first click, move the mouse around the center of the marquee, which causes the view to spin around the line of sight. This operation is illustrated in Figure 2.2.8.5.

Figure 2.2.8.5:
Spinning with the Navigate View tool. (a) Clicking on 1 activates the operator, and (b) clicking on 2 spins the view.

Zoom: 👇

This operation changes the view angle and focal length of a view. In perspective views, it functions like a variable zoom lens. In axonometric, isometric, and oblique views, it changes the extents of the view projection similar to the zoom tools.

To activate this operation, place the cursor on one of the two Zoom icons (👇), located at the ends of the vertical axes of the marquee. The cursor changes to the same icon indicating that the Zoom operation is active.

After the initial click, the view angle or extents is changed as the cursor moves horizontally. Moving the cursor to the right increases the view angle or extents, which zooms out. Moving to the left decreases the view angle, which zooms in.
Walkthrough

This tool allows you to take an interactive “walk” through a modeling scene displayed in the window. It is available for all the 3D views except for Panoramic*. Because the axonometric, isometric, and oblique types of 3D views are unable to simulate depth, the Walkthrough tool is limited to rotating and panning for these view types. Perspectives are the only views that can properly simulate depth as seen by the human eye. The Walkthrough tool is therefore most useful with perspective views and the following discussion refers mostly to them.

To execute walkthrough viewing, select an appropriate view type, then select the Walkthrough tool from the Window tool palette. A marquee is displayed that is used for controlling the walkthrough. Click at the center of the marquee and move the mouse away from it or click directly at a position away from the center. An operation is executed as soon as you do.

Unlike the other view tools, the Walkthrough tool does not translate the movement of the mouse into a view change. Instead, it uses the position of the mouse relative to the horizontal and vertical orientations to activate a certain operation. The distance of the mouse from the center of the marquee determines the speed of the operation. The farther away it is, the faster the operation is executed. When the cursor does not move the view still changes, except when the mouse is positioned at the center of the marquee, which is a stationary point.

The Walkthrough operations, illustrated in Figure 2.2.8.6, are as follows:

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Horizontal</th>
<th>Vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macintosh/Windows</td>
<td>left/right</td>
<td>up/down</td>
</tr>
<tr>
<td>none</td>
<td>look left/right</td>
<td>move in/out</td>
</tr>
<tr>
<td>option/ctrl+shft</td>
<td>look left/right</td>
<td>look up/down</td>
</tr>
<tr>
<td>control/ctrl+alt</td>
<td>pan left/right</td>
<td>pan up/down</td>
</tr>
<tr>
<td>command/ctrl</td>
<td>rotate left/right</td>
<td>rotate up/down</td>
</tr>
</tbody>
</table>

*Panoramic* is a view type that provides a wide-angle view of the scene.

![Figure 2.2.8.6: The functions of the Walkthrough tool.](image-url)

*Interface • The window tools*
The **Walkthrough Options** dialog (Figure 2.2.8.7), can be invoked from the Walkthrough tool. It contains a single option:

![Image](image_url)

**Figure 2.2.8.7:** The **Walkthrough Options** dialog.

**Save as Animated Camera:** When this option is on, the paths that are generated from the interactive walk are stored as an animated camera. When exiting the Walkthrough tool, the **Save View** dialog is invoked for naming the animated camera. The newly saved camera can subsequently be edited and used from the Views palette.

**Match Perspective View**

When you select this tool in perspective mode, a rectangle is drawn at the edges of the reference grid. This rectangle will be referred to as the **match** rectangle. If you are in a display mode other than perspective, a warning message will be issued. This tool allows you to move the points of its rectangle to match the shape of a rectangular surface on a background image. Once this is done, the new positions of your points are used to recalculate the perspective parameters. If your points accurately match those of the background image, your new perspective parameters match those of the background image. Having a background image is not a necessary condition and this method of defining perspective parameters will also work without one. However, its intention is to facilitate the process of matching your viewing parameters to those of a background image.

Where exactly the match rectangle appears depends on the active reference plane, which can be any of the Cartesian planes or any arbitrarily positioned plane. The match rectangle will appear even when the reference plane grid is not displayed. Which of the reference planes to use depends on the details of your background. You usually look for a rectangular shape in your image that is least distorted by the perspective and use a reference plane closest to its orientation.

The points of the match rectangle are highlighted with little black boxes. To move them, you click and drag them. You can also move segments of the match rectangle by clicking on them and dragging them. Each time you move an entity of the match rectangle the perspective parameters are recalculated and the display is refreshed. However, you can suspend the recalculation of the view and redraw by pressing the **shift** key. A perspective view matching operation will typically require a number of steps to get your perspective exactly right. When you finish, save your view in the Views palette so you can select it later.

The following keys modify the behavior of this tool:

- The **shift** key disables the redrawing of the changed view.
- The **control** (Macintosh) or **ctrl+alt** (Windows) moves the complete match rectangle.
- The **option** key (Macintosh) or **ctrl+shift** (Windows) moves the view along the reference plane.
- Clicking while pressing **control+shift** (Macintosh) or **ctrl+alt+shift** (Windows) resets the match rectangle to the corners of the reference plane.

The process of perspective view matching is illustrated in Figure 2.2.8.8.
(a) A previously rendered perspective image is placed in the window as an underlay. The window's view is initially at the default 30/60 perspective, which is the view at which the reference grid is displayed. Selecting the Match Perspective View tool displays the match rectangle, whose points are initially at the corners of the reference grid.

(b) The points of the match rectangle are moved to correspond to four corners of a rectangle in the image. If the `shift` key is pressed while the points are moved, nothing will happen until the `shift` key is released. When it is released a new view is calculated and the reference grid is now displayed at that view.

(c) Drawing two 3D objects, a cuboid and a cylinder, their display matches the view of the background image.

(d) The objects are displayed in a shaded image using `RenderZone*`. Note that while in the previous illustrations the background image was placed as an underlay, here the same image has been rendered as a background.

For clarity in this example, we matched the view to a rendered image. The backgrounds of real projects are typically photographs of environments to which new objects need to be blended. An example of blending a dining table into a real room is shown in Figure 2.2.8.9.

**Figure 2.2.8.8:** Matching the perspective view of a background image.

**Figure 2.2.8.9:** Blending a dining table to a room.
2.2.9 The memory box and zoom percentage menu

Next to the window tool palette, on the lower margin of the project window, there are two optional information items: the memory box and the zoom percentage pop up menu. They are both turned on/off from options in the Preferences dialog, as discussed in section 3.2.7. When on, they appear as shown in Figure 2.2.9.1(a).

The memory box keeps track of the memory consumption and status on your machine. There are two formats for displaying the information: one numeric and one graphic. The desired format can be set in the Preferences dialog or locally through a pop up menu invoked by pressing the mouse on the memory box, as shown in Figure 2.2.9.1(b).

**Available Memory:** When this item is selected, the memory box displays a number that represents how much memory is still available on your machine. When running on MacOS, this is the amount of memory allocated to the program (as set in the file information box in the Finder), less the amount that has been consumed. On Windows, this is the total memory available for the system. In both cases this number includes both physical and virtual memory. This display format is the default for Macintosh machines.

**Memory Meter:** When this item is selected, a meter bar that graphically represents the ratio between used and unused memory is displayed, as shown in Figure 2.2.9.1(c). The more memory is used, the more area of the bar will appear solid from left to right. The bar is drawn in green to represent physical memory and red to represent virtual memory (if available). As the memory consumption increases into virtual memory, the red area of the bar increases also, warning of a loss of performance. This format is the default for Windows machines.

The zoom percentage is essentially a scale factor, which is 100% when no zoom has been applied. It changes when zooming is applied through one of the Zoom tools, or by selecting a zooming factor from the zoom percentage pop up menu, which is as shown in Figure 2.2.9.1(d).
2.3 Palettes

In form-Z, palettes are small floating windows that support a variety of features of the program. There are a total of 24 palettes, which includes the Window Tools palette. Some of them are only available in modeling, some only in drafting, and some in both. The table in Figure 2.3.0.1 lists all of the palettes in alphabetical order and shows where they are available.

Note that the term palette is also used for the tool palettes that can be torn off the modeling and drafting tool bars. While these share some characteristics with the floating palettes discussed here, their functions are different. The complete modeling and drafting tool bars themselves are actually floating palettes that can be closed.

<table>
<thead>
<tr>
<th>palette</th>
<th>modeling</th>
<th>drafting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animation Editor</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Animation Score</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Animation Time Line</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Colors</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Coordinates</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Display Options</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Drafting Tools</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Hatch Patterns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layers</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Lights</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Line Styles</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Line Weights</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Objects</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Modeling Tools</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Pick Options</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Planes</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Profiles</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Prompts</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Scenes</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Snap Options</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Status Of Objects</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Surface Styles</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Symbols</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Tool Options</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Views</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Window Options</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Window Tools</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Figure 2.3.0.1: List of form-Z palettes.

On the other hand, the Window Tool bar is attached to the lower margin of the window, but may also become a floating palette through an item in the Palettes menu, the same way all the floating palettes can be opened, when closed. The Window Tool palette is also included when the Hide Palettes command is executed. This palette was discussed in the previous section. The general features of the floating palettes are discussed in the remainder of this section. The palettes and their palette specific features are discussed in subsequent sections. Note that the palettes are presented in a logical, rather than alphabetical, order.
2.3.1 Palette features

While all floating palettes are not identical they share certain characteristics. All palettes have a *title bar* at the top that contains the name of the palette. Clicking in the title bar brings the palette to the front so that it is not obscured by other palettes. Clicking and dragging in the title bar brings the palette to the front and repositions the palette as the mouse is moved. Pressing the *command* key (Macintosh) or *ctrl* (Windows) while clicking and dragging, prevents the palette from being brought to the front. Pressing the *option* key (Macintosh) or *ctrl+shift* (Windows) while clicking and dragging, simultaneously repositions all palettes which overlap or are adjacent to the selected palette. When palettes are repositioned, the edges of the palettes will snap to each other when they are within 5 pixels of each other.

The title bar also contains a *close box*. The appearance and location of the close box varies on the different operating systems. Clicking on the close box closes the palette. Clicking on the close box with the *option* key (Macintosh) or *ctrl+shift* (Windows), causes all of the palettes to be closed. Once a palette is closed, it can be opened again by selecting its name from the *Palettes* menu. Individual palettes can also be closed by selecting their name in the *Palettes* menu.

On OS X and Windows there is a *title bar menu* that is accessed by clicking in the title bar with the *control* key for OS X and by clicking in the title bar with the right mouse button (Windows). The appearance and content of this *title bar menu* may vary depending on the version of operating system that you are using. This menu provides standard system functionality at the top with *form*-*Z* specific functionality at the bottom. These are described below where appropriate and shown in Figure 2.3.1.1.

![Figure 2.3.1.1:](image)

The palette title bar menu on (a) OS X and (b) Windows XP.
The remainder of the standard palette features appear in some but not all of the palettes. See the descriptions and images of each palette in the following sections for details on which features they use. These standard palette features are described and shown in Figure 2.3.1.2.

- The **collapse window box** and **Collapse Window** title bar menu item collapses the palette and only its title bar remains visible. Collapsed palettes are expanded by clicking on their **collapse window box** or **Collapse Window** menu item again.

- The **orientation box** and **Change Orientation** title bar menu item changes the palette’s orientation from vertical to horizontal, and vice versa.

- The **scroll bar** is located on the right edge and is used to move the palettes content up or down.

- The **resize box** is located at the lower right corner and used to change the shape and size of the palette. When palettes are resized, the edges of the palettes will snap to each other when they are within 5 pixels of each other. Note that the individual palettes may have certain requirements for their dimensions. When resizing these palettes, the palette size will snap to the modular requirements of the palette.

*Figure 2.3.1.2:* The standard palette features of (a) OS X and (b) Windows.
There are a number of similar features in the content of the palette that are shared between some of the palettes. The following is a list of the commonly used features and which palettes use them:

- The Colors, Surface Styles, Symbols, Line Weights and Line Styles palettes include a horizontal bar at their lower margin end where there are tools for switching the format in which the information in the palette is presented.

- The Layers, Planes, Line Weights, and Line Styles palettes include a column on their left where a check mark is displayed in front of the active entity. A similar column is used in the Objects, Views, and Lights palettes to indicate which items are currently selected.

- The Objects, Layers, and Lights palettes include three or four columns on their right side where their visibility, selectability, and snapability, or shining and shadows status are displayed and may be changed. The Views palette includes a column in the right for controlling visibility.

  - The following selection/status methods work for both the pick and attributes columns:
    - To change the status of a single item, you **click** on the respective row and column.
    - To change a sequence of items, you **click and drag** in the respective column. The attributes will change to the status of the first item. The palette’s list of items will automatically scroll when the cursor reaches the top or bottom of the palette.

    - Clicking in the header of a column toggles the selection of all the items in that column. Clicking while pressing **option** (Macintosh) or **ctrl+shift** (Windows) selects all or none. If there is at least one item that is off, all are turned on. If all items are on, they are turned off.

    - The Colors, Layers, Line Weights, and Line Styles palettes contain items that can be used by symbols. When one of these items is referenced by a symbol, its parameters cannot be changed, because they are stored in the symbol library. These items are shown in the respective palette with a read only icon (X) next to the name or on top of the icon.

    - Double-clicking on an item in the palette invokes the dialog for editing it. This is equivalent to the **Edit** button in the palette’s dialog (see subsection 2.3.3).
2.3.2 Grouping and placing entities into groups

Groups can be defined for three of the form•Z palettes, Layers, Objects, and Lights, as follows:

- **Creating groups:** To create a group of objects, click in the Objects palette under the last name listed. To create a group of layers or lights, click in the respective palette while pressing `control` (Macintosh) or `ctrl+alt` (Windows).

After groups are created entities need to be placed in them, which is done as follows:

- **Placing a single entity into a group:** Click on the name of the entity (as listed in the respective palette) and drag it onto the name of the group.

- **Placing sets of entities and/or groups into a group:** First select the set of consecutive names by clicking to the right of a name and dragging the mouse up or down to select all the entities or groups that fall in the path of the mouse. Then click on one of the selected names and drag it onto the name of the destination group. This will drag and place into the group the whole set of preselected entities. Non-consecutive names can be selected by pressing the shift key and clicking on the desired names.

- **Collapse/uncollapse groups:** In the Layers and Objects palette names of groups are displayed with a little triangle that initially points down, indicating that the group is open. When open, all the entities it contains are displayed. Clicking on the triangle, turns it to the right and collapses the group. When collapsed the content of a group is not shown. Clicking on the triangle again opens the group.

- **Turning on/off visibility, selectability, and snapability of groups:** Click in the respective column next to the group name. This assigns the status of the group to all its members. If subsequently the status of some of the entities in the group is changed, the symbol at the group level is grayed. For layers, if one of the entities in the group is the active layer, the attributes can not be turned off for the complete group.
2.3.3 Palette dialogs and their commands

Some of the floating palettes are complemented by dialogs that can be invoked by *option* (Macintosh) or *ctrl+shift* (Windows) clicking in the palettes. These dialogs can also be invoked from items in the *Options* menu. The dialogs that are invoked from the Colors, Layers, Lights, Line Styles, Line Weights, Objects, Planes, Surface Styles, and Views share some common button commands, which are discussed in this subsection. These are illustrated in Figure 2.3.3.1, where the *Views* dialog is shown. Some of the dialogs also include additional type specific commands and these are discussed in the respective sections.

All dialogs have a button through which a new entity is created and saved. This button is either labeled *New...* or *Save...*, depending on the nature of the entity to be created. For example, the *Surface Styles* dialog contains *New*; clicking on it creates a new surface style. The *Views* dialog contains *Save...*; clicking on it saves the current viewing parameters into a new view. The *Layers* dialog is the only one that has two buttons for creating new entities: *Layer...* and *Group...*. Clicking on them creates a new layer or group, respectively. In all cases a dialog is invoked for entering the name of the new entity.

**Delete**: Clicking on this button deletes the entity selected in the dialog.

**Edit...**: Clicking on this button invokes a dialog through which the selected entity can be edited and changed. This is equivalent to double-clicking on the item in the palette.

**Copy**: Clicking on this button creates a copy of the active entity. The name of the new entity is created by appending the word “copy” to the original entity’s name.

**Top, Bottom**: Clicking on these buttons moves the active entity to the top or bottom of the list.

**Sort**: Clicking on this button sorts the list of entities, alphabetically.

**Clear or Purge**: Clicking on the former button, which is available in the *Lights, Planes*, and *Views* dialogs, clears all the entities from the list. Clicking on the latter button also deletes from the list all the empty entities, provided they are not used as follows:

- for the *Objects* dialog, it deletes all the empty groups;
- for the *Layers* dialog, it deletes all the empty layers and empty layer groups;
- for the *Colors, Surface Styles, Line Style*, and *Line Weight* dialogs, it deletes all the entities that are not assigned to an object or element.

*Interface* • *Palettes*
2.3.4 The contextual palette menu

Each palette has a contextual menu that is a shortcut for accessing frequently used items. It is designed to improve workflow by minimizing access to features that are available in dialogs or reducing the number of clicks to perform a task. This menu is derived from the standard method for contextual menus established by the operating system.

Different methods for accessing the contextual menu are available for single-button and multi-button mice, as follows:

- When using a multi-button mouse, click on the right (secondary) mouse button while the cursor is over a palette. The menu can also be accessed by using the control + left (primary) mouse button (Macintosh) or ctrl + left (primary) mouse button (Windows).

- When using a single-button mouse, click while pressing control (Macintosh) or ctrl (Windows).

The contextual menu key combination can be changed to any desired combination with a mouse button, using the key shortcuts manager.

The contextual menu appears at the click point and is different for each palette. Each item represents functionality that is available in a related dialog. The contents of some palettes also vary depending on the click location in the palette. For example, in the Objects palette, the content depends on the column in which the click occurs. Two of the Object palette menus are shown in Figure 2.3.4.1.

At the bottom of all of the contextual menus is an item for accessing the options or manager dialog that corresponds to the palette. The last item in the menu is Manual, which opens the form-Z help viewer to the section describing the palette.

Figure 2.3.4.1: Two of the contextual menus invoked from the Objects palette.
2.3.5 Transferring palette items

Items from one palette are often transferred to the same palette of another project. For example, when you copy/paste an object between projects its layer and color will be transferred to the new project. Most of these transfers follow similar rules, which are discussed in this subsection.

There are three ways by which items from one palette can be transferred to the same palette of another project. Two of these ways are explicit and one is implicit.

- The **Load From Project** button found in the dialogs that support the palettes, invoked by clicking in the palette while pressing `option` (Macintosh) or `ctrl+shift` (Windows), can be used to transfer the content of a palette in a project opened through the Open File dialog, into the same palette of the active project.

- The **Copy/Paste** items in the **Edit** menu can be used to copy items from a palette in one project, and paste them into the same palette of another project.

- Attributes listed in palettes, such as color and layer, are transferred from the palette of one project into the palette of another project when an object or an element to which they have been assigned is copied from one project into another. This is an implicit way to transfer palette items.

When palette items are transferred from one project to another, one of the following cases occurs:

1. The palette item transferred into a palette of another project is unique relative to the items already contained in the palette. That is, the transferred item has a unique name and set of parameters. This being the most common case, the transferred item is simply added to the palette.

2. The destination palette already contains an item whose name and parameters are identical to those of the transferred item. In such a case, the program does not add the transferred item to the destination palette.

3. The destination palette contains an item whose parameters are the same as those of the transferred item, but the two items have different names. In such a case, the **Same Parameters** dialog is invoked. Exceptions to this are items transferred when symbols are placed, in which case, no dialog is invoked and the item is added to the palette. The dialog is shown in Figure 2.3.5.1 as it appears when transferring lights with the same parameters. It contains one check box and three buttons. Clicking on a button executes the respective action, and exits the dialog.

![Figure 2.3.5.1: The Same Parameters dialog.](image)
**Apply To All**: If this check box is selected, the option you choose from the dialog will always be applied in similar cases involving the transfer of items to the palette.

**Add New**: Clicking on this button adds a new item to the palette. If the transferred item is an attribute assigned to an entity also being copied into the project, the entity will reference the transferred palette item.

**Use Existing**: Clicking on this button does not add the transferred item. If the transferred item is an attribute assigned to an entity also being copied into the project, the entity will reference the existing item with the same parameters as the transferred item.

**Use New**: Clicking on this button (default) replaces the existing item with the new transferred item. If the item is an attribute, all the entities to which either the existing or the transferred item was assigned will reference the newly transferred item.

4. The destination palette contains an item whose name is the same as the name of the transferred item, but the parameters of the two items are different. In such a case, the **Same Name** dialog is invoked, as shown in Figure 2.3.5.2 for the lights. An exception to this are items transferred when placing symbols, in which case no prompt is posted and the item is added to the palette.

![Figure 2.3.5.2: The Same Name dialog.](image1)

**Apply To All**: This check box is as for the **Same Parameters** dialog.

**Rename New**: Clicking on this button invokes the **Rename** dialog (Figure 2.3.5.3), in which a new name may be entered for the item in conflict. If you enter a name that is already in use, an error will be issued and the **Rename** dialog will be invoked for each conflicting name.

**Use Existing, Use New**: As for the **Same Parameters** dialog.

![Figure 2.3.5.3: The Rename dialog.](image2)
2.4 Coordinates and Prompts palettes

Prompts are the instructions that the program displays for guiding you through the execution of an operation. Each tool has its own set of prompts, but similar operations issue similar prompts. The number of prompts issued while executing an operation varies between tools and depends on the steps required to complete the operation.

The modeling objects and the drafting elements of form•Z are created within a world space in which positions are identified using an orthogonal coordinate system. Coordinates can be measured using the English or the metric system, and can be expressed in a variety of formats. The type of measurement, the type of coordinates, and the formats in which they are displayed can be selected from the Project Working Units dialog, shown in Figure 2.4.1.3. As the mouse is moved about the screen, its coordinates are displayed in the Coordinates palette.

At start up, the Coordinates palette (Figure 2.4.1.1) appears at the upper right corner of the screen. It consists of two or three fields (for X and Y, or for X, Y, and Z). These fields are arranged vertically by default, but can also be displayed horizontally, as discussed later in this section.

The Prompts palette appears in the lower left corner of the graphics window when form•Z is launched using default preferences. The Prompts palette is offered for convenience, and it is not necessary to have it open when running the program. It can be closed by clicking on the close box and can be reopened from the Palettes menu.

The Prompts palette (Figure 2.4.2.1) offers three main functions:

(a) It guides the user through the execution of an operation by issuing prompts and indicating the steps of an operation.
(b) It displays (echoes) the numeric values of the parameters used for the execution of an operation.
(c) It offers the option to enter parameters numerically.

The Coordinates and Prompts palettes complement each other but also offer distinct functionalities. The Coordinates palette is discussed in the next subsection (2.4.1) and the Prompts palette follows in subsection 2.4.2.
2.4.1 The Coordinates palette and types of coordinates

The Coordinates palette, which by default appears at the upper right corner of the screen, consists of two or three fields, depending on whether it is displayed in drafting or modeling. These are for the X, Y, and may be Z directions. By default, they are arranged vertically, but can be displayed horizontally by clicking on the arrow in the upper right corner of the palette, as shown for modeling in Figure 2.4.1.1.

![Coordinates palette](image)

*Figure 2.4.1.1: The modeling Coordinates palette in (a) a vertical and (b) a horizontal arrangement.*

The Coordinates palette can be positioned anywhere on the screen by clicking and dragging its title bar. It can be closed by clicking on the close box in its top margin. Once closed it can be invoked again through the Palettes menu. The Coordinates palette can also be collapsed by clicking on the hide box located next to the orientation arrow.

As the mouse moves, the coordinates of its position are constantly displayed in the Coordinates palette. The position of the mouse may be echoed:

- in **absolute** or **relative** values;
- relative to the coordinate system of the **world space** or **reference plane**;
- in **Cartesian** or **polar** coordinates (distance and angle).

These options can be selected from the Project Working Units dialog, or directly from the Coordinates palette by clicking on the boxes labeled **A**, **W**, and **C**.

- **A**: When this box is on, the unit mode is set to absolute coordinates. The absolute method displays the coordinate values relative to the origin of the coordinate system. When deselected, the unit mode is set to relative. With the relative method, the displayed coordinate values represent the distance traveled from the previously entered point.

- **W**: This box switches between the world and reference plane coordinate systems. This distinction is significant only when a Cartesian plane has been moved from its original position or an arbitrary (custom defined) plane is the active reference plane. Selecting **W** displays the coordinate values of the world space. Deselecting **W** displays the coordinates of the reference plane.
• **C**: This box switches between Cartesian and polar coordinates. A **polar** coordinate consists of two values: the **distance** of the current point relative to another point and the **angle** of the line defined by these two points relative to the horizontal direction. The other point is the Cartesian origin of the world space if the **A** option is selected, or the previously entered point if the relative option is selected (**A** is off). Depending on whether the **C** box is selected or not, the numbers echoed in the Coordinates palette represent the distance/angle polar coordinates (labeled **d** and **a**) or the **x**, **y**, and **z** Cartesian coordinates. Displays of the two types of coordinates are shown in Figure 2.4.1.2.

![Figure 2.4.1.2: Displaying (a) Cartesian and (b) polar coordinates.](image)

Both English and metric units are available for displaying coordinates. The English units may be displayed in **feet** or in **inches**. The metric may be displayed in **meters**, **decimeters**, **centimeters** or **millimeters**. All these options can be selected from the **Project Working Units** dialog (Figure 2.4.1.3), which is invoked from the **Options** menu or by **option** (Macintosh) or **ctrl+shift** (Windows) clicking in the Coordinates or the Prompts palette.

![Figure 2.4.1.3: The **Project Working Units** dialog.](image)

It contains two fields for determining the desired accuracy of numeric and angular values and two tabs (**Numeric Options** and **Angle Options**) that contain additional dialogs for setting the formats in which numbers and angles are displayed. These formats are discussed in more detail in subsection 3.3.6. Examples of displays of the four numeric formats available are shown in Figure 2.4.1.4.

![Figure 2.4.1.4: Formats available for coordinates: (a) **Architectural**, (b) **Engineering**, (c) **Fractional**, and (d) **Decimal**.](image)
2.4.2 The Prompts palette

The Prompts palette is a floating window that has many of the standard window features as well as four check boxes. Although it can be moved, closed, resized, and scrolled vertically, its orientation cannot be changed. Most of it is occupied by a display area where prompts are issued and the parameters of an operation are echoed as it is executed.

As soon as a tool is selected, the name of the operation is displayed in bold characters and the first prompt is issued. For example, as shown in Figure 2.4.2.1(a), if the Extrude modifier is activated and then the Rectangle tool is selected, the Prompts palette will display the words “Rectangle” on its top row, and will issue the prompt “First Corner Point (x,y,z):” on its next row. As soon as the mouse is clicked in the graphics window, a coordinate value is displayed next to the first prompt and the prompt “Second Corner Point (x,y,z):” is displayed on the next row. After the next click of the mouse, its coordinate value is displayed. If Graphic/Keyed is selected in the Heights menu, the final prompt “Height:” is issued, as shown in Figure 2.4.2.1(b).

As soon as an operation is completed, the previous prompt displays scroll up and the name of the operation reappears on the top row of the palette. The name changes as soon as another tool is selected. Whenever an operation that takes many entries is executed (such as the drawing of a polyline), the input lines scroll up as soon as the display area of the Prompts palette is filled. The display area can be enlarged using the resize box (lower right corner) to allow for more input lines. Once enlarged, the scroll bar of the palette window can also be used to scroll down so that prompts of previously executed operations are also displayed. The system retains up to twenty lines of previously issued prompt lines together with the numeric values entered during the execution of the operations. Whenever the Prompts palette is resized, closed, and then opened again, it appears with the size and shape it had when it was closed.

Figure 2.4.2.1: The Prompts palette (a) before and (b) after the generation of a cube.
Next to its scroll bar on the right margin, the Prompts palette contains four check boxes. Three of them are labeled $A$ (for absolute), $W$ (for world), and $C$ (for Cartesian). These boxes are the same switches found in the Coordinates palette and work the same way (see previous subsection).

The fourth box is labeled with a $T$, which stands for tracking. This check box controls whether or not the motion of the mouse or digitizer is continuously tracked and echoed in the palette. When deselected, the system simply displays the final position of the mouse (the position where the mouse was clicked). Turning the tracking on or off affects the way in which numbers are entered through the keyboard. For example, when tracking is off, numbers are typed into a blank field. When on, numbers are typed on top of numbers echoed by the system. The latter may be confusing in some cases, while in others it makes it easy to enter one or more numbers by simply changing those already there. The mouse is only tracked when it moves in the graphics window. The tracking freezes when the mouse cursor is anywhere on the Prompts palette, which is appropriate when numeric input is entered through the keyboard (see next section).

The Prompts palette is to some extent redundant with the Coordinates palette, but the two can complement each other. You can choose to have one, both, or neither open. One way in which the Prompts palette may be complemented by the Coordinates palette is when each is set to its own coordinate format (such as relative or absolute). In certain instances, having access to both values facilitates an operation. However, the two palettes complement each other primarily due to the nature of the numeric values they echo and/or accept from the keyboard. The Coordinates palette echoes the coordinates of the mouse or digitizer. The Prompts palette echoes the numeric value of the parameter that will be used in the execution of an operation. When that parameter is a coordinate value, then the information in the palettes is redundant. When the parameter is of a different character, such as an angle of rotation or a scaling factor, then the two palettes display different information.
2.4.3 Indirect and direct numeric input

form•Z has been designed to maximize interactive graphic communications. Complex models can be derived without ever touching the keyboard. However, precise numeric input is sometimes necessary and is provided as a global alternative to the graphic input. Whenever the system requires some input in order to execute an operation, that input may be entered graphically or numerically. The two methods may actually be freely mixed during the execution of a single operation, by entering some of the required input graphically and some through the keyboard. Entering values through the keyboard is considered direct numeric input. Numeric input can also be entered graphically, which is referred to as indirect numeric input.

Indirect numeric input

Indirect numeric input occurs when numeric values are entered through the mouse or digitizer. Graphic input is accomplished by moving the mouse or digitizer and clicking its button. As the motion of the mouse is constantly echoed in the Coordinates palette and the parametric values used to execute an operation are constantly displayed in the Prompts palette. The values shown in these palettes at the time the mouse is clicked are those entered into the system. You use the indirect numeric input method when you continuously read the values displayed in the palettes as the mouse is moved and you click as soon as the desired value appears. The method of indirect numeric input works best when some snapping is on. Due to the integer nature of the screen’s bit map, it is not always easy to move the mouse to fractional positions. In these cases, direct numeric input should be used (see below). When the desired positions have integer values, then turning some snapping on allows the motions of the mouse to generate them rather easily.

Direct numeric input through the Prompts palette

Direct numeric input involves using the keyboard to type numbers in the numeric fields of the Prompts or the Coordinates palette, whenever the system expects an input. Note that numbers being typed into the field must follow the unit format (English or metric) set in the Project Working Units dialog.

To enter numeric input through the Prompts palette, first select the desired tool. As soon as a prompt appears, type the numeric value expected by the system. When a coordinate value is expected, enter three numbers separated by commas and then press the return or enter key. If a single numeric value is expected (such as degrees, height, or a scaling factor) type it in and press return. The numbers typed should be at least as many as required by the prompt, otherwise zero values will be entered. As soon as you press return/enter, another prompt is issued until all the required parameters have been entered and the operation is executed. The result of the operation appears on the screen as when you use graphic input.
When you enter numeric values through the keyboard, the mouse should remain idle. If it is moved on the graphics window before the return key is pressed, the position registered by its movement takes precedence over what was typed. To avoid this when you use numeric input, either turn off mouse tracking (deselect the little box labeled “T”) or place the mouse cursor in the Prompts palette, where its movement is not registered. Numeric input can be freely mixed with graphic input. That is, if the execution of an operation requires two values, you can enter one by clicking the mouse on the appropriate position, and you can type the other on the keyboard.

The vast majority of the operations require a fixed number of parameters and are completed as soon as the required input is entered. When you enter vector lines by typing in the numeric values of their points, you may use the mouse or you may type “e” or “c” to complete the input sequence. A double click (and “e”) generates an open line. A triple click (and “c”) generates a closed shape. Retyping the first entry at the end of the input list and double clicking (or enter “e”) also produces a closed shape. Note that the key commands “e” and “c” are the defaults and other assignments can also be made in the Key Shortcuts Manager dialog (see subsection 3.2.5).

The use of mouse clicks with the numeric input of vector lines is illustrated in Figure 2.4.3.1. The input of the line to the left (a) was completed after double clicking at the end of the first sequence of numbers (shown in the Prompts palette). For the shape to the right (b), the mouse was triple clicked.

The best way to execute certain operations is to mix graphic input with numeric. This allows actions that are of a graphic nature to be executed graphically, and actions that may benefit from the inherent accuracy of numeric input to be executed numerically.

For example, assume that a previously generated cube needs to be moved by exactly 20 feet in the X direction. Objects are moved by first picking them, and then indicating the position to which they should be moved. Graphically, an object is picked by clicking the mouse on it. Numerically, an object is picked by typing in the coordinates of a point that is close enough to one of its segments. This can be done by entering values in the Coordinates palette (see subsection 2.6.4) and pressing return/enter with the Pick tool active. However, the pick operation is much easier when executed graphically. Consequently, when a pick is required, the user is prompted to do so, rather than entering a numeric value. After the object has been picked, it can be moved by graphically entering its destination point. But, given that the object needs to be moved by exactly 20 feet in the X direction, it is easier to type the number. In this particular case, the move of the cube by 20 feet is relative to its previous position, and the best way to complete the operation would be to switch to relative coordinates by deselecting the A box in the Prompts palette, then type 20’ and press the return key. If you want to move the cube by 20’ in the Y direction, type 0, 20’ and then press the return key. If you want to move the cube by 20 feet in the Z direction, type 0, 0, 20’ and then press the return key.
Numeric input through the Coordinates palette

Coordinate values can be typed directly into the Coordinate palette fields at any time. To do so, select a field in the Coordinates palette in the usual manner (by dragging the mouse over it or by placing the insertion bar in the field) and type the desired number. Use the tab key to move between the fields, and press return to enter the value into the program’s memory. The coordinate value will also be displayed in the Prompts palette, next to the current prompt, as soon as the return key is pressed.

The tab key moves the insertion bar from one field to another, in a circular manner. As numeric values are typed in the X, Y, and Z fields and the tab key is pressed, the position of the graphic cursor is adjusted. That adjustment is visible in the graphics window. When the return key is pressed, the values currently shown in the X, Y, and Z fields are entered into the system, and are assigned to the active parameter.

When entering numeric values into the Coordinates palette, the mouse should be kept stationary until the input is completed by pressing return. Moving the mouse overrides what was typed, in the same way that typing a value overrides the previous motion of the mouse. The most recent input, whether from the mouse or the keyboard, overrides the previous one.
2.5 The Tool Options palette

The Tool Options palette by default appears at the upper right portion of the screen. It shows the options of the currently active tool and of the modifiers that affect it. When an icon is selected, the Tool Options palette is refreshed to show the options in effect. These options are the same as the options found in the tool dialogs that are invoked by double clicking on the tools. The Tool Options palette offers a more direct way of accessing them.

The options in the palette are organized in logical sections, which are the same as those in the corresponding dialogs. In general, the dialogs and Options palette share the same internal settings for the different parameters. When one is changed through a dialog, it shows up with the same value in the palette, and vice versa. Options that appear in different dialogs, are grouped in separate sections in the Options palette. For example, the Status Of Objects options are displayed in their own section, whenever a tool that uses them is active.

The Tool Options palette can be resized, closed, and collapsed as any window. However, it has no scroll bars and its content can only be scrolled by dragging the mouse in it (see below). Its sections and subsections can also be collapsed to hide the options and preserve space in the palette. Frequently, all the options that are relevant to an operation and need to be displayed can not fit in the Tool Options palette. While its size may be increased, there are also operations that allow you to move through the palette and/or collapse sections of options that you do not need:

- To **scroll the content** of the palette click in the palette on blank space (i.e. away from any options) and drag the mouse up or down. Alternatively, `control` (Macintosh) or `ctrl+alt` (Windows) clicking anywhere in the palette and dragging up or down achieves the same result. Note that the Tool Options palette has no scroll bars, consequently this is the only way to pan its contents up and down.

- To **collapse a section** of the palette, click on the little down pointing arrow (Macintosh) and plus sign (Windows), at the upper right portion of the section. This hides the content of the section and all of the options below the section move up in the palette. On the Macintosh, the arrow turns to the right and on Windows, the plus sign changes to minus to indicate the collapsed section. To **open the section** click on the arrow/minus sign again. Note that these are analogous to folder operations in the operating system. These operations are illustrated in Figure 2.5.0.1.
When selecting the Polygon tool with the 3D Enclosure modifier active, the options of both tools need to be displayed. If you are working on a relatively small monitor, all the options do not fit, as shown in Figure 2.5.0.1(a). Note that the size of the Tool Options palette is constrained by the default position of the Lights palette at its bottom. You can close or collapse some of the other palettes and resize the Tool Options palette to display all its content. You can also scroll its content or collapse some of its sections, as illustrated in Figure 2.5.0.1.

To scroll the content of the palette, click the mouse in a “white” area of the palette and drag it up, as shown in Figure 2.5.0.1(b). Note that the upper portion that was visible before contained the Polygon options. The lower part contains the options for the 3D Enclosure. There are no Status Of Objects options listed since they are not used by this operation.

The alternative is to close some sections of the palette. Scroll back to the top of the palette and then click on the collapse triangles of the Edges and Pattern sections. This closes them and hides their content, which now allows all the other options to be visible, as shown in Figure 2.5.0.1(c).
2.6 Color, surface styles, and their palettes

In form•Z color is used in the interface for enhancing the readability of a variety of features and for painting the objects that the program creates. The colors used for drafting elements are selected from the Colors palette. The colors used for painting and rendering modeling objects are contained and selected from the Surface Styles palette. In other words, the Colors palette is available in the drafting environment, while the Surface Styles palette is used in the modeling environment. Surface styles include color, but may incorporate additional rendering effects such as transparencies, reflections, and bumps. The surface styles available in the plain version of form•Z consist of color only, which may be made transparent. The surface styles in form•Z RenderZone Plus incorporate the full range of rendering effects, as required for supporting photorealistic rendering.

Color display is sensitive to the graphics card(s) in your computer. If you have multiple monitors connected to your machine, form•Z is sensitive to the main monitor as set up in the monitors (Macintosh) or display (Windows) control panel. form•Z works best with graphics cards which support 16, 24 or 32 bit color display (also known as direct, true, real, thousand or millions of colors.) form•Z will work with older 8 bit (or 256 color) graphics cards; however, all of the resulting colors will be dithered. Dithering is a technique by which a color shade that is beyond the value of available colors is approximated by mixing two or more of the available colors through rasterized patterns.
2.6.1 The Colors palette

In the drafting environment of form·Z, the Colors palette is displayed in the upper left corner of the screen. By default, it consists of 16 squares, arranged in two horizontal rows, as shown in Figure 2.6.1.1(a). Each square displays a different color. The third row has extra space for defining additional colors, if desirable. The Colors palette can be freely moved and positioned anywhere on the screen, can be closed, collapsed, and resized. In the lower margin it has three icons that allow you to select one of the three display methods, namely large image, small image, or name list. When closed, it can be opened again through the Palettes menu.

More colors can be created and added to the Colors palette, or colors can be edited or deleted, using the following operations:

- To **add a color** to the Colors palette, click anywhere on its blank space. This invokes the Color Picker dialog and a color is defined in the usual manner.

- To **delete a color** from the palette, click on the box of the color you wish to delete while pressing control (Macintosh) or ctrl+alt (Windows).

One of the squares of the Colors palette is always highlighted with a black box, indicating that this is the **active color**. Its name appears in the lower margin, together with three icons. By default, the first square at the upper left corner is the active color and is highlighted. A new active color can be selected by clicking the mouse in the desired color's square. The name of the active color is shown in the lower margin, next to the three display icons.

The active color is assigned to elements at generation time. The colors of elements can also be changed after their initial generation with the Color ( ) tool. When clicking on an element with this tool active, the active color is assigned to it (see subsection 4.25.1).
Any of the colors shown in the palette can be freely changed. To redefine a color, double click the mouse on the respective square. The system responds by presenting the Color Picker dialog. Note that four variations of the dialog are available on the Macintosh (two shown here) and a different one on Windows (Figure 2.6.1.2).

The Colors palette is supplemented by the Colors dialog, shown in Figure 2.6.1.3. This is invoked by option (Macintosh) or ctrl+shift (Windows) clicking in the Colors palette or from the Options menu. Its buttons work as is discussed in subsection 2.3.3.

Figure 2.6.1.3: The Colors dialog.

Figure 2.6.1.2: The Color Picker dialog on (a) and (b) OS X and (c) Windows.
2.6.2 The Surface Styles palette

In modeling, the Surface Styles palette appears in the upper left corner of the screen. In its simplest form, a surface style is a plain color similar to the colors in the Colors palette. Such are actually the surface styles of the plain version of form•Z. Surface styles can also be much more than simple colors and can include textures, reflections and refractions, transparencies, and bumps, as they do in form•Z RenderZone Plus. The surface styles are defined through four layers of procedures called shaders, set through the Surface Style Parameters dialog, invoked by double clicking on a surface style in the Surface Styles palette. These operations are discussed in detail in the form•Z RenderZone Plus User’s Manual. The Surface Styles palette is similar to the Colors palette and also has a title bar, a close box, a hide box, a scroll bar and a resize box. In addition, in the lower margin, it has three icons that allow you to select one of three available display methods, namely large image, small image, or name list. Two of these formats are illustrated in Figure 2.6.2.1.

Contrary to the Colors palettes, where the colors are displayed on flat boxes, the surface styles are shown on 3D objects. These are cubes by default, but in RenderZone Plus they can be changed to cylinders, spheres, or 3D surfaces, which is selected when a surface style is defined. The types of display objects can actually be mixed in the same palette. Displaying surface styles on 3D objects makes it possible to show the rendering effects contained in a surface style.

New surface styles can be added and existing can be deleted as with the Colors palette. Also, one of the surface styles is the active surface style that is highlighted with a black box and its name appears in the lower margin, next to the three icons. This is assigned to all new objects at the time they are generated. You can select another active surface style by clicking on it. You change the color and the other rendering effects that a surface style may incorporate by double clicking on it, to invoke the Surface Style Parameters dialog.

Option (Macintosh) or ctrl+shift (Windows) clicking in the Surface Styles palette invokes the Surface Styles dialog (Figure 2.6.2.2), which can also be invoked from the Options menu. Its buttons are as discussed in subsection 2.3.3.
Double clicking on a surface style in the palette or selecting the **Edit...** button in the **Surface Styles** dialog invokes the **Surface Style Parameters** dialog, shown in Figure 2.6.2.3. If you are running **form•Z RenderZone Plus**, this dialog contains all of the parameters for realistic surfaces (see section 6.2). In the regular version of **form•Z**, the surface style contains basic color and transparency information:

**Name**: This is the name of the surface style. It can be any unique name up to 31 characters.

**Color**: This is the color of the surface style. Clicking on the color box to the right invokes the Color Picker dialog shown in Figure 2.6.1.2.

**Transparency**: This slider and numeric field determine how transparent the style is. 0% (slider all the way to the left) indicates no transparency and objects that use this style will render opaque. 100% (slider to the right) indicates full transparency, which makes an object completely invisible. Settings between the two end values specify analogous levels of transparencies

**Preview**: This box shows a cube rendered with the surface style defined in the dialog, which allows you to inspect the effects of your settings.

**Copy From**: Clicking on this button invokes a dialog that displays the palette with the list of surface styles (Figure 2.6.2.4). Any of the surface styles displayed can be selected by clicking on it. When **OK** is selected to close the dialog, the parameters of the highlighted surface style are copied into the surface style, which is active in the Surface Styles palette.

Note that this operation is different from the **Copy...** operation available in the **Surface Styles** dialog. While the latter duplicates the active surface style and creates a new item in the Surface Styles palette, this **Copy From...** operation copies the parameters of a selected surface style into the surface style active in the Surface Styles palette, which is an existing surface style.

*Figure 2.6.2.3*: The **Surface Style Parameter** dialog as it appears in the regular version of **form•Z**.

*Figure 2.6.2.4*: The dialog invoked from the **Copy From...** button.
2.6.3 Project colors and transferring colors or surface styles

Certain colors have been assigned by default to a number of features, as shown in the **Project Colors** dialog (Figure 2.6.3.1), which invoked from the **Options** menu. Clicking on a color box in the dialog invokes the Color Picker dialog, where the color can be changed the usual way.

Also included in the **Project Colors** dialog is a pop up menu that controls how the colors of drafting elements and the surface styles of modeling objects are treated when the elements or objects are pasted from one project to another.

**When Pasting Between Projects:** This pop up menu contains three items (Figure 2.6.3.2).

- **Use Active Color:** When this item is selected, the pasted entity takes the color or surface style that is active in the receiving project.

- **Use Best Matching Color:** When this item is on, the pasted entity takes the color or surface style in the receiving project that is the closest to its original color or surface style.

- **Transfer Original Color:** This option is the default. When selected, if a color or surface style in the project being pasted into matches the color or surface style in the project from which the entities were cut or copied, this color or surface style is assigned. If a match is not found, the color or surface style is added to the project being pasted into, and this color or surface style is assigned.
Layers are available in both the modeling and the drafting environments. Layers are essentially a method by which entities in a project can be organized in distinct groups. The entities in one layer may be treated independently and may even be completely isolated from entities in another layer. form-Z offers an unlimited number of layers, layer groups, and three tools for manipulating them: the Layers palette, the Layers dialog, and Set Layer tools, one in modeling and one in drafting. The first two are discussed here. The Set Layer operators are discussed in subsections 4.25.8 and 5.15.4.

The layers have names, which may be defaulted by the system or assigned by the user. A layer can be active or inactive. Only one layer can be active at any given time and one layer is always active. The active layer is where new objects/elements are placed when you generate them. Layers can be independent or part of a group. If they are part of a group, layers can share or have independent attributes as discussed below.

Layers have three attributes that can be accessed from the palette. They are similar to those found in the Lights and Objects palettes. The attributes of the layers apply equally to all the layers independently of whether they are active or not. The rest of the layer attributes are available in the Layer Attributes dialog discussed in subsection 2.7.2.

The visibility attribute has three states: visible, ghosted, and invisible.

The selectability attribute determines whether or not an entity in a layer can be selected (picked) or not, and has two states: locked and unlocked.

The snapability attribute determines whether the system will snap to the entities of a layer, when one of the object snaps is active. It has two states: snapable and not snapable.

New projects start with a layer called “Layer 1,” which is automatically the active layer. By default, it is visible, unlocked, and snapable. Its name and its attributes can be changed by the user, and new layers can be added. The layers are listed in the Layers palette, which can be used to switch layers and to change their attributes. These and additional operations can also be executed through the Layers dialog. The Set Layer tools allow you to place modeling objects or drafting elements onto the active layer (see subsections 4.25.8 and 5.15.4).
2.7.1 The Layers palette

When starting form•Z using the system defaults, the Layers palette (Figure 2.7.1.1) appears on the right of the project window. For a new project it contains a single layer carrying the default name Layer 1. This name can be changed either through its palette or through its dialog.

Under the title bar there is a horizontal headers bar, which contains the labels of its five columns. These are (from left) the active status column, indicated by a check mark (√), the name column, the visibility column, the selectability column, and the snapability column. The main operations that can be executed through the palette are as follows:

- **Creating a layer**: Clicking in the white space under the last layer name listed in the palette generates a new layer whose name appears in the palette in text edit mode. While in this mode a new name can be typed. Clicking again under the name exits the edit mode and enters the name. The new layer defaults to visible, unlocked, and snapable.

- **Creating a layer group**: Clicking as above while pressing the `ctrl` (Macintosh) or `ctrl + alt` (Windows) key generates a group of layers. A new name is typed as above and the name is entered after clicking under it in the palette.

- **Placing layers into groups**: To place a single layer, click on the name of the layer (as listed in the palette) and drag it onto the name of the group. To place sets of layers into a group, first select the set of consecutive names by clicking to the right of a name and dragging the mouse up or down to select all the layers that fall in the path of the mouse. Click on one of the selected layer names and drag it onto the name of the group. This will place all the layers into the group.

- **Opening and closing groups**: Groups are displayed with a little triangle that initially points down, indicating that the group is open. When open, all the layers it contains are listed (Figure 2.7.1.1(a)). Clicking on the triangle turns it to the right and closes the group. To open a closed group, click on the triangle again.

- **Selecting an active layer**: Clicking in front of a layer name in the left column of the palette displays a check mark and makes it the active layer. There is only one active layer and all new objects are placed in the active layer. The active layer can not be a layer group. To pick a set of non-consecutive names, click on each while pressing the `shift` key.

![Layers palette](image)

*Figure 2.7.1.1: Layers palette with (a) all groups unfolded and (b) two groups folded.*
• Changing a layer's attribute: Clicking in an attribute’s column, next to the layer’s name, switches the status of that attribute. The selectability and snapability attributes will toggle on/off each time the mouse is clicked. Locked (non-selectable) layers are indicated with the lock symbol (🔒), and snapable layers are indicated with a bullet (●). The visibility attribute cycles through three states: visible (🔹), ghosted (◇), and invisible (blank), in that order.

• Changing sets of attributes: Clicking on the header of an attribute changes the state of that attribute for all the layers in one step. However, the active layer always stays visible, and is not affected when you click on the header of the visibility column to turn all the others off. Pressing the shift key while you click on the header of the visibility, selectability, or snapability column will toggle (reverse) all the entries in that column. Pressing option (Macintosh) or ctrl+shift (Windows) when you click on one of these headers will turn them all on if some are on and some are off. It will turn them all off if they are all on.

• Changing the attributes of groups: To turn on/off the visibility, selectability, and snapability of layer groups, click in the respective column next to the group name. This assigns the status to all group members. If subsequently the status of some of the entities in the group is changed, the symbol at the group level is grayed. However, if one of the layers in the group is the active layer, the visibility attribute cannot be turned off for the entire group.

• Suspending redraws: When the visibility attribute of a layer is changed, the screen is immediately refreshed to show the objects/elements in their new state. When the visibility state of more than one layer needs to be changed, and you want to avoid the screen redrawing process as the visibility attribute of each layer is changed, hold the shift key down as you change the attributes. The screen will be refreshed as soon as you release the shift key.

• Changing the sequence of a name in the list: Press the mouse on the name and drag it to the desired position, up or down the list. Drag the mouse over consecutive names to select them and move them, or shift-click on individual names to select non-consecutive names.

• Selecting, changing, and deleting names: To select a layer name, click on it. Selected names are highlighted. The name of a layer listed in the Layers palette can be changed after it has been highlighted using standard text editing methods. Selecting the name of a layer and pressing the delete key erases the name, but not the layer itself. This is normally followed by the typing of a new name in the name field.

• Deleting a layer: Select the layer name in the Layers palette and press command + delete (Macintosh) or ctrl+backspace (Windows).

The layers and their groups can also be manipulated through the Layers dialog, invoked from the Layers palette or from the Options menu. It is discussed in the next subsection.
2.7.2 The Layers dialog

The Layers dialog (Figure 2.7.2.1) can be invoked by selecting the Layers... item in the Options menu, or directly from the Layers palette. Some of the operations that are executed from the Layers dialog are the same as those executed through the palette. Others are only available in the dialog. The Layers dialog consists of a window similar to the Layers palette and behaves similarly when the previously discussed operations are executed. It also contains a number of button commands and check boxes. Many of these are as in other dialogs and as discussed in subsection 2.3.3.

Layer...: This button invokes the Name New Layer dialog (Figure 2.7.2.2) where a new name is typed. The name is entered at the end of the list as a new layer.

Group...: Clicking on this button creates a new layer group and works as for new layers.

Delete: This button deletes the selected layer and all the entities assigned to it, after an alert.

Edit...: Clicking on this button with a layer name selected invokes the Layer Attributes dialog (Figure 2.7.2.3), which allows you to assign a layer the same attributes available to individual objects or elements.
When an attribute is set for a layer, it overrides object level attributes. The attribute is set by selecting the check box to the left of the attribute name. For example, if the Surface Style attribute is on, then all the objects on the layer will render with that surface style regardless of their object surface style. Note that, since the modeling and drafting environments involve different attributes that are assigned to objects and elements respectively, different versions of the Layer Attributes dialog are used, as shown in Figure 2.7.2.3. For more details on assigning attributes see sections 4.25 (modeling) and 5.15 (drafting).

**Copy...**: This option copies the selected layers or groups. It does not copy the objects on the layers, just the layers in the palette.

**Top** and **Bottom**: They move the selected layer to the top and bottom of the list, respectively.

**Sort**: This button sorts the names alphabetically. Upper case characters precede the lower case.

**Purge**: This button deletes all the empty layers and all empty later groups, after issuing a warning.

**Active**: This button makes the selected layer active, and indicates this by placing a check mark in the active column, in front of the layer’s name.

**Load Project Layers...**: This button invokes the Open File dialog in which a project is selected whose layers will be transferred to the active project.

**Draw In Order Shown**: When this option is selected, the contents of the layers are plotted in the order the layers appear in the Layers dialog, each time the screen is refreshed.

**Draw Active Last**: When this option is selected the active layer is plotted last.

**Paste On Active Layer**: This check box determines the layer on which an entity will be placed when pasted into a project. If not selected, the system will place the pasted entity on the layer which has the same name, provided such a name exists. If it does not exist, it will create a new layer with that name. If this option is selected, the pasted entities is placed onto the active layer.

**Show Color**: When this option is on, the names of layers are displayed in the palette with the color that has been assigned to them by default or through Edit.... This option is off by default.

**Highlight Picked**: When this option is on, picking an object in the graphic window (or through the Objects palette) will highlight the name of the layer it is in. The name is highlighted by displaying it in the default red highlight color.

**Default Name, Default Group Name**: When new layers and groups of layers are created, default names are generated for them. These names are derived from the strings entered in these fields and a numeric index appended to them.
2.8 Objects and their palette

Objects are geometric entities that are generated by drawing them or through a variety of other operations available in the modeling environment of form•Z. When objects or groups are generated, they are automatically assigned a name, and they are listed in the Objects palette. The initial names are generated by form•Z based on a keyword, which is by default “Object” for objects, and “Group” for groups. Names are constructed by appending numeric indices to these key words. The default keywords can be changed in the Objects dialog (see section 2.8.2). You can change the initial name generated by the program either directly in the Objects palette, or through the Query dialog, which can be invoked directly from the Objects palette as well as through the Query tool.
2.8.1 The Objects palette

When you first launch form-Z, or open a new project, the Objects palette is empty, as shown in Figure 2.8.1.1(a). As you create objects and possibly group them, their names and the names of their groups are listed in the Objects palette as shown in Figure 2.8.1.1(b). Groups can also be created directly from within the palette and objects can be placed into them as follows:

- **Creating a group**: Click in the name column in the blank space under the last item in the list. This creates a new, initially empty group. The group’s name appears highlighted and can be changed. A second click under the new name completes the creation of the group.

- **Placing objects into a group**: Drag the name of an object or of another group onto the name of a group. This makes it a member of that group.

- **Removing members from a group**: Drag the name of an item in a group outside that group.

![Object palettes](image)

*Figure 2.8.1.1:* Object palettes: (a) Empty, no objects in project. (b) Listing objects and groups. (c) Listing with the groups compressed.
The Objects palette consists of five columns, similar to those in the Layers palettes:

- The first column (left to right) is the **selection** column, and displays a check mark in front of the object or group that is selected.

You can select an object by clicking in the selection column, which also highlights the object in the graphics window. When the Pick tool is used to select an object in the graphics window, a checkmark is placed in front of the object’s name in the Objects palette. Clicking on a checkmark causes it to disappear, and the object is deselected. If an object is deselected in the graphics window, the checkmark is also deleted from the Objects palette. The same operations can be applied to groups, and work in an analogous manner. Contrary to the Layers palette where only one checkmark appears in the first column, any number of checkmarks may appear in the Objects palette, since any number of entities can be selected.

- The second column is the **name** column, and displays the names of the objects or groups. The names of groups are denoted with a small triangle, which points down when the entities it contains are also listed, or to the right when they are not. You can switch from one type of listing to the other by clicking on the triangular marker (see Figure 2.8.1.1(c)).

- The third column is the **visibility** column, which shows three states: visible (♦), ghosted (◇), and invisible (blank). Clicking on the diamond marker switches the visibility state. The Ghost/Unghost tools and the **Query Object Attributes** dialog can also be used to change the ghosting of an object.

- The fourth column is the **selectability** column. All objects and groups are created selectable. Clicking next to their name in the selectability column displays a lock symbol (●), and makes them non-selectable. Clicking on the lock symbol makes them selectable again. Locked objects cannot be picked and no operation can be applied to them while they are locked.

- The fifth column is the **snapability** column. All objects are created snapable, denoted with a bullet next to their name. Clicking on a bullet makes it disappear, and the object non snapable.

The visibility, selectability, and snapability settings of objects can be toggled globally by clicking on the title of the respective column, and the redrawing action can be delayed by pressing the shift key when a setting is clicked, as for the other form.Z palettes. All the attributes can be turned on and off at the group level by clicking on the attribute in the group name row. All objects in the group will reflect the change. The following operations can also be executed through the palette:

- Clicking in the title bar of the same palette invokes the **Objects** dialog (see next section).

- Clicking on an object/group in the Objects palette selects the name, which can be changed.

- Double clicking on the name of an object invokes the **Query Object Attributes** dialog, in which the name of the object, or any other attributes, can be changed. Double clicking on the name of a group invokes the **Query Group** dialog. This is equivalent to clicking on the display of an object with the Query tool active (see subsection 4.22.1).
2.8.2 The Objects dialog

Clicking on the title of the name column of the Objects palette invokes the **Objects** dialog, shown in Figure 2.8.2.1, which can also be invoked from the **Objects...** item in the **Options** menu. This dialog is similar to the other palette supporting dialogs, and allows you to execute many of the same operations as can be executed directly through the palette, plus additional functionality.

![Objects dialog](image)

*Figure 2.8.2.1:* The **Objects** dialog.

Most of the operations available in the dialog are applied to the **active** listing of an object or group. You make a listing active by clicking on it. Depending on whether an object or a group listing is active, as well as the grouping level of the active listing, some of the operations available in the **Objects** dialog may not apply, in which case they are dimmed and inactive.

**New**: Clicking on this button creates a new group. Note that this button cannot be used to create a new object. New objects can be created only by using one of the modeling tools.

**Delete**: Clicking on this button deletes the selected object or group. If a group is selected, all of its children groups and objects are also deleted.

**Edit...**: Clicking on this button invokes the **Query Object** dialog for an active object or the **Query Group** dialog for an active group. In these dialogs, the attributes of the object or group can be changed.

**Copy**: Clicking on this button creates a copy of the active object or group. The name of the new entity is created by appending the word “copy” to the original object’s name. When a copy of a group is made, copies of all the objects it contains are also made. Note that these copies are made in exactly the same positions as the original objects, and are not visible until they are moved or transformed in some other way.

**Top, Bottom**: Clicking on this button moves the active listing to the top or bottom of the list for the same grouping level. To move a listing that is contained in a group to the top or bottom of the complete list, it must first be extracted from its group.
**Sort:** Clicking on this button sorts the list of objects and groups at all grouping levels.

**Purge:** Clicking on this button deletes from the list all the empty groups, that is, groups which contain no objects.

**Show Color:** When this option is on, the names of layers are displayed in the palette with the color that has been assigned to them by default or through **Edit...**. This option is off by default.

**Highlight Picked:** When this option is on, picking an object in the graphic window (or through the Objects palette) will highlight the name of the object in the Object palette. The name is highlighted by displaying it in the highlight color.

**Default Object Name, Default Group Name:** These editable text fields contain the strings that are used to create default names for objects and groups, respectively. By default, these strings are “Object” and “Group,” but can be changed to any other string.

**Default Object Axes:** This group of options sets the initial placement and orientation of an object’s axes when the object is created.

- **Per Object Type:** When this option is selected, which is the default, the object origin and axes are positioned according to the type of the object. Some objects have what may be thought of as a “natural” origin. For example, the center of the sphere or the center of the base of a cylinder. This is also true for the symbols and all the analytic primitives. For objects not in these categories a default location is set using other object characteristics or its generation method.

- **Center Of Gravity:** With this option selected, the origin of the axes is placed at the exact center of the object (mass) and the axes directions are aligned to the active reference plane.

- **Average Of Points:** With this option selected, the origin of the axes is placed at the average of all the coordinate points of the object. The axes are aligned to the reference plane.

- **Center Of Bounding Volume:** With this option selected, the centroid is placed at the midpoint between the minimum and maximum of the bounding volume in each of the three directions (x, y, and z). The axes are aligned with the active reference plane.

**Object Centroid:** This group of items controls the positions of the centroid of the objects.

- **Center Of Gravity, Average Of Points, Center Of Bounding Volume:** These options are as for their counterparts in the **Default Object Axes** group. Default is **Center Of Gravity**, which is the most accurate but also the most computationally intensive.
2.9 Views and their palette

A **view** of a 3D modeling scene is determined by the position of the viewer and a number of other viewing parameters. **form-Z** offers a variety of methods for producing and manipulating different types of views. Some of them are simple and direct, while others require a more precise definition of the viewing parameters, either through numeric input or through graphic interactions.

The **View** menu contains a number of items that produce quick or precise views (see section 3.5). The Window Tools palette contains four tools that produce views through graphic interactions (see subsection 2.2.8).

These methods produce new views that replace the previous one, which is lost unless it has been saved. Views can be saved through the Views palette and its companion dialog. Once saved, you can retrieve a view by simply clicking in the Views palette.

While views are typically thought of as referring to 3D space, in a limited way, they are also meaningful in drafting. Different views can be derived in drafting by panning and/or rotating the draft space around the center of the screen. In drafting, views can also be changed through items in the **View** menu, and through the Set View tool in the Window Tool palette. They can be saved to and invoked from the Views palette, as for modeling.
2.9.1 The Views palette

The modeling Views palette, shown in Figure 2.9.1.1, is very similar to the Objects palette. In addition to the standard window features, it consists of a horizontal **headers** bar and six columns for **selection, active** status, **name, type, visibility, and selectability**. The drafting Views palette is simpler than this and only has two columns, one for the **active** status and one for **name**.

In the modeling Views palette, orthographic projections are designated as Top, Back, Left, etc. and 3D views are designated as Axon (axonometric), Isom (isometric), Oblq (oblique), Pers (perspective), and Panm (panoramic).

Initially the Views palette is empty. To create a view, click the mouse in the name column. The system defaults a name using the word “view” and a numeric index in the name column. The name is highlighted and can be changed by typing another. To complete the creation of a named view, click the mouse again anywhere in the name column. The system stores the current view under the name entered and marks it as the active view. Any number of views can thus be named and saved through the Views palette.

Once saved, you can switch from one view to another by clicking in the active view column, next to the view you wish. This places an eye icon next to the view name and the screen is refreshed to show the new view. View names and their order can be changed and views can be deleted.

If the visibility attribute of a view is turned on, the graphic representation of the view will be visible in the graphics window. When views are visible, they are displayed in the modeling window with wireframes and they can be picked and transformed. Figure 2.9.1.2 shows the graphic representation of axonometric and perspective views. When visibility of a view is on, you can select it in the selection column. When a view is selected, its graphic representation is highlighted, and a checkmark is present in the selection column. If the representation of a view is selected with the Pick tool, the checkmark will appear automatically in that view’s selection column.

![Figure 2.9.1.1: The Views palette (a) before and (b) after saving named views.](image)

![Figure 2.9.1.2: Representations: axonometric and perspective views.](image)
2.9.2 The Views dialog

The Views palette is complemented by the **Views** dialog (Figure 2.9.2.1) which can be invoked from the **Views** menu (see subsection 3.5.4) or directly from the Views palette by clicking the mouse on the name header or by clicking anywhere on the palette while pressing the **option** (Macintosh) or **ctrl+shift** (Windows).

The **Views** dialog contains all the common button commands that work as discussed in subsection 2.3.3, with the following clarifications:

- You **Save...** new views through the **Name View** dialog (Figure 2.9.2.2).

- You **Edit...** views through the **View Parameters** dialogs (Figure 2.9.2.3). Note that there are different dialogs in modeling and drafting.

- When you want to load views from another project, you click on the **Load Project Views...** button to invoke the normal Open File dialog.

![Image of Views dialog](image1)

**Figure 2.9.2.1:** The **Views** dialog.

![Image of Name View dialog](image2)

**Figure 2.9.2.2:** The **Name View** dialog.

![Image of View Parameters dialog](image3)

**Figure 2.9.2.3:** The **View Parameters** dialog in (a) modeling and (b) drafting.
2.10  Lights and their palette

In 3D programs, lights are a necessary compliment to the colors that are assigned to objects. Their type and their position affect how colors look and how objects are shaded.

Only one type of light, **distant** light, is available in the regular version of **form•Z**. More types are available in **form•Z RenderZone Plus**. Distant (or direct) light emits parallel rays from an infinitely distant light source, such as the sun. Its intensity remains constant throughout a scene. In **form•Z**, one of the lights has a special designation, and will be referred to as the **sun**. Lights affect the **Quick Paint***, **Surface Render***, **Shaded Render***, and **Interactive Shaded** modes of rendering. The first two only use the sun light, which is always the light designated as sun. If more lights are defined in the Lights palette, they are ignored. **Shaded Render** works with any number of lights, but frequently a few or just one light produces the best results. The sun designation has no effect on **Shaded Render** and all the lights are treated the same way. Examples of the effects of single and multiple lights and their shadows are shown in Figure 2.10.0.1.

At start up, **form•Z** automatically creates one distant light, which is designated as the sun. In addition, a scene always contains **ambient** light. The lights and their names are displayed in the Lights palette. New lights can be defined and existing lights can be deleted or edited through the **Lights** dialog. The parameters and the positions of lights can be set in the **Light Parameters** dialog. The positions of lights can also be manipulated using the geometric transformation tools of **form•Z**. Lights may or may not cast shadows at different levels of quality. The specific functionalities of lights and how they are activated are discussed in the remainder of this section.

![Figure 2.10.0.1](image)

*Figure 2.10.0.1: Renderings with (1) one light and (2) two lights. (a) Lights as shown in wire frame. Renderings (b) without and (c) with shadows.*
2.10.1 The Lights palette

All currently defined light sources are displayed in the Lights palette, as shown in Figure 2.10.1.1. If not open, this palette can be invoked from the Palettes menu. The Lights palette is similar to the other form•Z palettes and responds to the same operations. It consists of five columns.

- The first column is the selection. Clicking in it picks the corresponding light, highlights it in the project window, and displays a check mark in front of it. Clicking on the check mark deselects the light. The same occurs when lights are selected in the window with the Pick tool. Only visible and selectable lights can be selected.

- The second column is the type. It displays a light type icon: = distant light and ☀ = sun, and the name of the light. To change a distant light to a sun (and vice versa), click on its icon. There can be only one sun at a time, so the program makes the necessary adjustments when the sun designation is changed. Lights can also be rearranged by clicking and dragging the light name as in other palettes. Additional types of lights are available in form•Z RenderZone Plus (see section 6.6).

- The third column is the visibility column. When a black diamond (♦) is on, the light is visible in interactive rendering modes and a graphic representation is drawn. When a white diamond (◇) is shown, the light is drawn in the ghosted color. If no icon is shown, the light is not drawn.

- The fourth column is the selectability column. When the lock (❖) icon is displayed, the respective light is locked and cannot be selected. If there is no icon, the light can be selected and operated on.

- The fifth column is the shining column. When a bullet (●) icon is displayed, the respective light shines. If no bullet is present, the light does not shine.

- The sixth column is the shadows column. When the shadow icon (◉) is on, the light casts shadows. It does not, otherwise. The shadow parameters are set in the Light Parameters dialog.

Lights can be grouped in a fashion similar to those available for objects and layers.

- Creating a lights group: Click in the Lights palette while pressing control (Macintosh) or Ctrl + Alt (Windows).

- Placing lights into a group: Drag the name of a light or another group onto the name of a group.

- Removing lights from a group: Drag the name of a light from within a group outside that group.

- Picking groups of lights: Set the topological level to Group and with the Pick tool click on a light displayed in the graphic window. Or click on the name of a light group in the Lights palette. When a group of lights is picked, all the lights in the group are picked and highlighted. Clicking in the top bar where “Light Name” is displayed invokes the Lights dialog. Double clicking on the name of a light in the second column invokes the Light Parameters dialog.
2.10.2 The Lights dialog

New lights can be created and existing lights can be deleted, copied, and edited through the **Lights** dialog (Figure 2.10.2.1), which is invoked by clicking in the name bar of the Lights palette, or by selecting **Lights...** from the **Options** menu. The left side of the dialog contains a window similar to the palette, while the right side has a number of button commands which execute the following operations:

**New...**: Clicking on this button creates a new light. The **Light Parameters** dialog is invoked, which displays the attributes of the new light and allows these attributes to be edited.

**Group...**: Clicking on this button creates a new group of lights. This is equivalent to clicking in the Lights palette while pressing **control** (Macintosh) or **Ctrl +Alt** (Windows).

**Delete**: Clicking on this button deletes the currently highlighted light.

**Edit...**: Clicking on this button invokes the **Light Parameters** dialog, which displays the attributes of the currently highlighted light, and allows these attributes to be edited.

**Copy**: Clicking on this button creates a copy of the active light. To name the new light, the name of the active light is used with the word “copy” appended to it. A different name can be assigned by selecting the **Edit** command.

The lights are initially entered into the Lights palette in the order in which they are created. This order can be changed using one of the following button commands:

**Top, Bottom**: Clicking on one of these buttons moves the highlighted light to the top or bottom position of the list, respectively.

**Sort**: Clicking on this button sorts the list of lights alphabetically, according to their names.
**Clear**: When this button is selected, all lights except for the one used as the sun are deleted. If you try to delete this light, a warning will be issued and the operation cancelled.

**Sun**: With a distant light selected in the list of lights, clicking on this button designates the distant light as a sun light. This is equivalent to clicking on the distant light icon in the Lights palette. If the selected light is of a type other than distant or no light is selected, this button is dimmed.

**Load Project Lights....**: Clicking on this button invokes the standard Open dialog, which allows you to select a form-Z project file whose lights will be appended to the end of the current list of lights, and will be listed in the Lights palette.

**Show Color**: With this option on, the names of the lights are sown in the palette with the color that has been assigned to them. This option is off by default.

**Highlight Picked**: With this option on, picking a light or a group of lights in the graphic window (or through the Lights palette) will highlight the name of the light or group of lights. This option is off by default.

**Ambient Light**: The two options in this group control the parameters of the ambient light.

  **Color**: This box displays the color of the ambient light, which can be changed by double clicking on it to invoke the Color Picker.

  **Intensity**: This slider parameter controls the brightness of the ambient light.

**Default Light Name, Default Group Name**: When new lights or groups of lights are created, default names are generated for them. These names are derived from the strings entered in these fields with a numeric index appended to them.
2.10.3 Editing the parameters of a light

The Light Parameters dialog (Figure 2.10.3.1) allows you to change the attributes of a light. This dialog is invoked each time you execute the New... or Edit... commands in the Lights dialog. It can also be invoked by double clicking on the name of a light in the Lights palette.

The Light Parameters dialog contains the Name and Color fields at its top, a preview window in its upper right, light attribute parameters at its lower right, and three tabs: Intensity, Location, and Shadows. The RenderZone Plus version of the dialog contains one more tab: Parameters.

Name: A new name for the light can be entered in this field. It can be up to 31 characters long.

Color: The current color of a light, which is white by default, is displayed in this box. It can be changed in the usual manner through the Color Picker dialog from which another color can be selected.

Intensity tab

Brightness: The value entered in this field controls the intensity of the light. Note that the normal range for this parameter would be from 0% to 100%. However, you can also enter a value greater than 100%. If you do, be aware that you will be defining an excessively bright light which may overexpose your image. On the other hand, you may also enter values less than 0%, which result in removing brightness from a scene. In reality, lights with less than 0% intensity do not exist, but they may be used to correct overexposed areas of an image.

Figure 2.10.3.1: The Light Parameters dialog with the (a) Intensity, (b) Location, and (c) Shadows tab open.
**Location tab**

**Origin:** The values entered in these $X$, $Y$, and $Z$ fields determine the position of the light source in 3D space.

**Center Of Interest:** The values entered in these $X$, $Y$, and $Z$ fields determine the position of the center of interest of the light source. The location and center of interest of a light taken together determine the *direction* of a distant light.

When a light is generated, its position is defined by the **Origin** and **Center of Interest** parameters. After it has been generated, its position can be adjusted either by changing the values of the parameters, or through graphic manipulations. Note that, if more than one light is created with the same values entered in the **Origin** and **Center Of Interest** fields, all the lights will be on top of each other and will not be individually visible. To be able to see them, some of the lights must be moved.

Any of the lights can be repositioned using one of the geometric transformation tools, namely Move, Rotate, Independent and Uniform Scale, and Mirror, available in the modeling tool palette. You can move the origin, the center of interest or the entire light. When applying a geometric transformation to a light, the setting of the topological level is ignored. Which part of the light is transformed depends on where the mouse is clicked when selecting the entity to be moved:

- Clicking on the light origin transforms the point of origin of the light source.
- Clicking on the center of interest transforms the point of the center of interest.
- Clicking on the light line moves the complete light.

Other than these distinctions, all the geometric transformations (move, rotate, resize, and mirror) work as for the common objects, and can be executed in both prepick and postpick mode. Transformations on lights can also be combined with the Self/Copy modifiers to make transformed copies.

**Light Direction From Sun...:** Clicking on this button invokes the **Sun** dialog (see section 3.5.5 and Figure 3.5.5.1 in *form*-$Z$ User’s Manual), which allows you to use its options to define the position of a distant light. Note that this applies to all the distant lights and not only to that designated as sun. Setting sun parameters to more than one distant light is useful when you wish to depict the shadows of a model over a period of time in the same image. Specifying the direction of distant light through the **Sun** dialog simply copies new values into the location fields of the light.

**Shadows tab**

**Shadows:** When this option is on, the light will also cast shadows. However, to do so the **Shadows** option must also be selected in the dialog of the type of rendering being executed. The parameters and the process of shadow casting is discussed in detail in section 3.6.7.
**Preview**: This preview window allows you to verify the effects of the current light. The displayed image is generated by applying the current light settings to a simple white object sitting on a white surface. By default this object is a cube. In **form-Z RenderZone Plus** the object can be a plane, cube, cylinder, or sphere, which is selected from the pop up menu under the preview window. Note that the **Origin** and **Center Of Interest** parameters are not used when generating the preview image, which is always generated from a fixed position and direction for the light source.

Under the preview window there is a pop up menu and four checkboxes. They control the attributes displayed in the columns in the Lights palette, which are initially set according to the selections made from this dialog.

The pop up menu has three items, as follows:

**Invisible**: When this item is selected, nothing is displayed in the visibility (3rd) column of the Lights palette and the light representation is not displayed in the project window.

**Visible**: When this item is selected, a black diamond (◆) will be entered in the visibility column, resulting in a wireframe image of the light being drawn in the project window when the wireframe display mode is used.

**Ghosted**: When this item is selected, a white diamond (◇) will be entered in the visibility (3rd) column of the Lights palette, resulting in a ghosted image of the light being drawn in the project window when the interactive display modes are used.

**Locked**: When this option is selected, a lock (■) icon will be entered in the selectability column (4th) of the Lights palette, which prevents the light shown in wireframe mode from being picked.

**Shining**: When this option is selected, a bullet (●) will be entered in the shining column (5th) of the Lights palette, which makes the respective light an active or illuminating light. If this option is off, then a light still exists, but does not contribute to the illumination of a scene until it is turned on.
2.10.4 Editing the parameters of groups of lights

Similar to invoking the Light Parameters dialog, double clicking on the name of a group in the Lights palette invokes the Light Group Attributes dialog, shown in Figure 2.10.4.1. Light groups contain attributes that can override the parameters of individual lights in the group.

The Light Group Attributes dialog contains a Name field, four pop up menus, one for each of the light attributes, and a group of options labeled Override Attributes.

Visible, Locked, Shining, and Shadows: These pop up menus display the status of the respective attribute, if all the lights in the group are assigned the same state of the attribute. If different members of the group are at a different state, three asterisks (*** are displayed. Selecting a new item from the pop up menu changes the respective attribute for all the lights in the group to the selected state.

Override Attributes: This group contains options that, when checked, override the options that may be assigned to individual lights. If an option is not checked, then the option assigned to an individual light is applied when the group of lights is used in a rendering. When overriding individual assignments, the latter are not lost but are preserved and are used again when an option is unchecked or a light is removed from a group.

Override options are the Color, Intensity, Shadow Type, Glow, and Radius/Angle. These are the same options available for individual lights. However, the Intensity option has some additional functionality, as follows:

Replace: When this option is on, the intensity parameters of the lights in the group are replaced by the intensity parameters of the group.

Scale By: When this option is selected, the intensity output of the light is scaled by the factor in the text field next to the option. This option is useful when the overall exposure of a scene needs to be adjusted without changing the intensity of the lights relative to each other. To do this, a new light group should be created and all lights in a scene placed in it. Then the intensity override can be turned on with the Scale By option and an appropriate scale factor selected.
2.11 Scenes and their palette

When working with 3D models, it is often desirable to work with a sub set of the entire model or be able to render in specific views with specific rendering parameters. A scene is a collection of display characteristics including: the display of objects, layers, and lights; view and rendering parameters; grid, reference plane, and underlay settings. The visibility state (visible, ghosted, or invisible) is retained in a scene for objects and layers. The visibility, shining, and shadow parameters are retained in the scene for lights.

Scenes are created and restored in the Scenes palette and are managed in the Scenes dialog. The basic workflow with scenes is to create a scene and then select the desired characteristics of the scene (which objects are visible for example). This process is repeated for each desired scene. Making the scene the active scene restores a scene. If new objects, lights, and layers are created after the scene is created, they are added to the scene in the state they are in when the scene is made active (unless the scene is locked).

The state of the active scene is automatically retained (unless the scene is locked). For example, if a scene is made active and then an object is made visible, it will be retained as visible in that scene. Only one scene can be active at once. It is also possible to have no scene active, which is the default state.

Scenes contain information that is both project (display of objects, layers and lights) and window level (view and rendering parameters). When a scene is made active, the object, layer, and light changes are visible in all windows of the project. The scene view and rendering parameters only affect the active window. As it is desirable to have scenes that only include certain information, scenes can be set to only retain certain types of information (see Scene Options dialog in Figure 2.11.3.1).

The Imager is capable of rendering scenes as described in section 3.12.
2.11.1 The Scenes palette

The Scenes palette, shown in Figure 2.11.1.1, is enabled by selecting its name from the Palettes menu. The palette contains a list of the scenes in the project. Clicking in the name header at the top of the palette or clicking anywhere in the palette while pressing the option (Macintosh) or ctrl+shift (Windows) keys invokes the Scenes dialog, which is used to manage the scenes (see Figure 2.11.2.1). The palette is empty by default. The palette has three columns.

The first column is the active column. When a scene is active, a check mark appears in this column on the row of the currently active scene. To make a scene active, click in this column on the row of the desired scene. To make no scenes active, click on the check mark of the active scene.

The second column is the name column. Clicking in this column at the bottom of the list creates a new scene. Clicking on the name of an existing scene enables the editing of the scene name.

The third column is the lock column. When a scene is locked, changes made to the project while a scene is active are not retained. Objects, lights, and layers created after a scene is locked are not visible when the scene is made active.
2.1.1.2 The Scenes dialog

The Scenes dialog, shown in Figure 2.1.1.1, is invoked from the Scenes palette and is used to manage the scenes. It contains a scene list on the left and a number of buttons on the right side. The scene list looks similar to the Scenes palette. A scene from the list can be selected by clicking on it. Many of the buttons on the right side affect the selected scene.

- **New...**: Clicking on this button creates a new scene at the bottom of the scene list. The Scene Options dialog (see Figure 2.1.3.1) is invoked to enter the name and other parameters of the scene.

- **Edit...**: Clicking on this button invokes the Scene Options dialog that is used to edit the name and other parameters of the currently selected scene.

- **Delete**: Clicking on this button deletes the currently selected scene from the scene list.

- **Copy**: Clicking on this button makes a copy of the currently selected scene from the scene list.

- **Top, Bottom**: Clicking on one of these buttons moves the highlighted scene to the top or bottom position of the list, respectively.

- **Sort**: Clicking on this button sorts the list of scenes alphabetically, according to their names.

- **Clear**: When this button is selected, all scenes are deleted.
2.11.3 The Scene Options dialog

The **Scene Options** dialog, shown in Figure 2.11.3.1, is invoked by pressing **Edit...** in the **Scenes** dialog. Its options are as follows:

**Name**: This is the name of the scene.

**Scene Includes**: This group of items determines what information is retained in the scene. These options are all on by default.

- **Objects**: When this option is selected, the visibility state (visible, ghosted, or invisible) of objects is retained in the scene.
- **Layers**: When this option is selected, the visibility state (visible, ghosted, or invisible) of layers is retained in the scene.
- **Lights**: When this option is selected, the visibility, shining, and shadow parameters of lights are retained in the scene.
- **View Parameters**: When this option is selected, the view parameters are retained in the scene. If the active window uses a view from the view palette, the active view is retained in the scene.
- **Rendering Parameters**: When this option is selected, the rendering parameters for all rendering modes are retained in the scene.
- **Grid**: When this option is selected, the grid display characteristics from the **Window Set-up** dialog (see section 3.3.5 in the **form•Z** User's Manual) are retained in the scene.
- **Reference Plane**: When this option is selected, the current reference plane and arbitrary plane definition (if one exists) are retained in the scene.
- **Underlay**: When this option is selected, the underlay parameters from the **Underlay** dialog (see section 3.3.8 in the **form•Z** User's Manual) are retained in the scene.
2.12 Symbols, their palette and dialog

Symbols are a method by which groups of modeling objects or drafting elements that need to be repeated in a project are stored in libraries, from which they can be placed as many times as desired. In both modeling and drafting, form·Z offers tools for constructing symbol definitions, for placing and editing symbol instances, and for decomposing symbol definitions to the parts from which they were constructed.
2.12.1 The Symbols palette

Once defined and stored in symbol libraries, symbol definitions are displayed in Symbols palettes, from which they can be selected and placed into a modeling or drafting project. In the symbol palettes, the symbols are displayed in one of three available formats, as shown in Figure 2.12.1.1. You can switch from one format to another by selecting one of three icons ( ), found at the left end of the lower margin of the symbol palettes.

The lower margin of the Symbols palette also contains three icons, labeled with the numbers 1 through 3 ( ). They represent the three levels at which a symbol definition may exist. You switch from one level to another by selecting the respective icon. The name of the symbol library is displayed under the title bar of the Symbols palette. The name of the active symbol is displayed on the right end of the lower margin of the palette.

Figure 2.12.1.1: The Symbols palette: definitions displayed (a) by name, (b) by small icon, and (c) by large icon.
2.12.2 The Symbol Libraries dialog

Clicking on the top row of the symbols library (where the word “Library” is) invokes the Symbol Libraries dialog (Figure 2.12.2.1) The dialog displays the list of currently loaded symbol libraries, and the symbols contained in the active library. Both the libraries and their symbols can be manipulated through the use of button commands available in the dialog.

![Symbol Libraries dialog](image)

Figure 2.12.2.1: The Symbol Libraries dialog.

The dialog also contains two Display options labeled Modeling and Drafting. Selection of one of these options displays the symbols of the respective environment, if such symbols exist (Figure 2.12.2.2). A Symbols library may consist of a modeling and a drafting part. The functionality of the symbols and their palette are discussed in more detail in section 4.19 (for modeling), and 5.8 (for drafting).

![Symbols palette](image)

Figure 2.12.2.2: The “Living room” library’s Symbols palette for (a) the modeling, and (b) the drafting environments.
2.13 More modeling palettes

The three palettes discussed in this section can be thought of as special palettes, unique to the modeling environment.

- The Planes palette lists reference planes that have been saved so that one may switch to them at a later time. Reference planes are unique to the 3D world of modeling, where they are used to draw and generate objects.

- The Profiles palette is a list of 2D shapes known as profiles. These are used for applying smooth movements to regions of meshed objects.

- The Status of Objects palette is redundant with a dialog with the same name. It contains the options that determine the status of both the operands and an operator (after the operation is executed) and of the resulting one or more objects.

There are three more palettes that are grouped with the modeling palettes and support animation: the Animation Editor, Animation Score, and the Animation Time Line palette. These are shown in Figures 7.1.6.1, 7.1.3.1, and 7.1.2.1 and are discussed in detail in sections 7.1.6, 7.1.3, and 7.1.2 respectively.
2.13.1 The Planes palette and dialog

Reference planes are used by the modeling operations to generate and manipulate objects and their parts. Any of the three Cartesian planes can be used as a reference plane or a user can define any arbitrarily positioned plane as a reference plane, using the Define Reference Plane tool in the window tool palette. Other tools in that same palette allow you to freely move and rotate planes and their origins. They are discussed in subsection 2.2.3. You can switch from one reference plane to another using one of the Set Reference tools in the window tool palette, as discussed in subsection 2.2.1. You can also switch reference planes by simply clicking in the Planes palette after reference planes have been named and saved in it.

The Planes palette, shown in Figure 2.13.1.1 can be opened from the Palettes menu and contains the standard features. It consists of three columns. The left column is the active status column and is labeled with a check mark (√). In the middle is the name column, where the names of the reference planes are listed. The column to the right is the type column, where one of four type indicators (XY, YZ, ZX, or ARB) appears.

Initially, the Planes palette contains no plane names (Figure 2.13.1.1(a)). To store a plane, click the mouse in the name column. The system defaults a name which consists of the word “plane” followed by a numeric index. The name appears highlighted and you can change it by typing another name. To complete the creation of a named reference plane, click the mouse again anywhere in the name column, away from the selected name field. The system stores the current reference plane under the name entered, and marks it as the active reference plane.

Figure 2.13.1.1: The Planes palette (a) before and (b) after saving reference planes.

Any number of planes can be generated, named, and saved through the Planes palette. You can switch from one to another by clicking in the active status column. When this is done, the check-mark is moved to that plane, and the screen is refreshed to show the new reference plane.

In the Planes palette, the names of the reference planes are initially listed in the order they are saved. You can change this order by clicking and dragging a name to another position. A reference plane can be deleted by selecting its name and then pressing command+delete (Macintosh) or ctrl+backspace (Windows). Pressing delete or backspace alone only deletes the name.
Reference planes can also be made active, saved, and deleted through the Reference Planes dialog (Figure 2.13.1.2) which is invoked from the Options menu. It can also be invoked directly from the Planes palette by clicking on the name header, or by clicking anywhere in the palette while pressing option (Macintosh) or ctrl+shift (Windows). The Reference Planes dialog works as all the other dialogs and so do its button commands.

Save: This button invokes the Name Reference Plane dialog (Figure 2.13.1.3) where a name can be typed. After clicking OK, the active reference plane is saved under the name entered.

Delete: This button causes the selected plane to be deleted. If that plane is the currently active plane, the plane listed under it will become the active plane.

Rename: This button invokes the Rename Plane dialog (Figure 2.13.1.3) in which a new name can be typed. It will replace the name of the currently selected plane.

Active: This button causes the selected plane to become the active reference plane. The change is visible in the active window after the Reference Plane dialog is closed.

Top and Bottom: These buttons move the selected plane to the respective positions in the list.

Sort: This button sorts the names of the planes alphabetically, where the upper case characters precede the lower case.

Clear: This button deletes all the arbitrary reference planes.

Load Project Planes...: This button invokes the standard Open file dialog in which a project can be selected for the purpose of transferring its arbitrary reference planes to the active project. The transfer is completed as soon as you click on the Open button in the Open file dialog.
2.13.2 Profiles and their palette

A profile is a 2D shape that is used to apply mesh movements to surfaces of meshed objects (see subsection 4.11.1). They are stored in the Profiles palette (see Figure 2.13.2.1) where the active profile is highlighted with a black box and is applied when the move mesh operation is executed.

When launching form•Z the Profiles palette is closed by default. It can be opened from the Palettes menu. It contains six predefined profiles and more can be added by the user. How this is done is discussed in details in section 4.11.1.

The Profiles palette has all the standard features, namely a title bar, a close box, a hide box, a scroll bar, and a resize box. The Profiles palette is only available in modeling.
2.13.3 Status of Objects palette

This is a special type of a palette that resembles the Tool Options palette in that it also makes more directly accessible options that can also be set in a dialog. These options are also available in the Tool Options palette for tools that are affected by them. The redundancy is due to the fact that, in the Status tab of the Tool Options palette, the status of objects options appear in a tab, which may require an extra click to access from. Being accessible in their own palette makes them easier to change, which is appropriate due to the frequency with which they are used. The status of objects options apply when new objects are generated from other objects and control what happens to the objects used in the operation. They also control the structure of volumes, when the result of an operation consists of more than one volume.

**Operand Status**: Three options are available: *Keep*, *Ghost*, and *Delete*. They determine what happens to an object used in an operation after the operation is completed.

**New Object Status**: When the result of an operation consists of more than one volume, two options are available: Selection of *Object Per Volume* results in a separate object being made from each volume; Selection of *Single Object* results in all the volumes being returned as a single object.

The Status Of Objects palette is not opened by default when launching form•Z. It can be opened by selecting the Status Of Objects item in the Palettes menu. It can be closed by deselecting this item or by clicking on its close box.

The Status of Objects options can be set to apply globally or per individual tool, through a box at the top of the palette, as follows:

- If the box is checked, the word “Global” is displayed next to it indicating that the settings in the palette apply globally (Figure 2.13.3.1(a)).
- If the box is not checked, the name of the currently active tool is displayed, indicating that the settings in the palette only apply to that tool (Figure 2.13.3.1(b)).
- Selecting a new tool displays its settings or the global settings, depending on whether the box is checked or not for that tool. If the new tool is not affected by the status of objects options, the Status Of Objects palette is dimmed.

*Figure 2.13.3.1:* Status Of Objects palette applying (a) globally and (b) to Union only.
2.14 More drafting palettes

There are three palettes that support features unique to drafting and are only available in that environment. They are discussed in detail in chapter 5 and are also summarized in the remainder of this section. A fourth palette, the Colors palette, is also unique to drafting but it functionally resembles the Surface Styles palette and has been discussed in subsection 2.6.1.

- The Line Styles and Line Weights palettes list a number of line types and weights, respectively, either graphically or by their names. These line styles and weights can be selected from the respective palette and applied to form•Z elements.

- The Hatch Patterns palette displays the patterns available in the drafting environment of form•Z. These patterns can be selected from the palette and applied to drafting elements.
2.14.1 The Line Styles and Weights palettes

Drafting elements can be drawn using different *line styles* and *line weights*, which are stored in the Line Styles and Line Weights palettes respectively, as shown in Figures 2.14.1.1 and 2.14.1.2. These palettes contain lists of styles and weights which can be shown either graphically or by their names. You can switch from one display format to the other by selecting one of the tool icons located at the lower margin of these palettes.

Both the Styles and Weights palettes contain a number of predefined styles and weights. Up to 256 can be added. One line style and one line weight are always *active*, which is indicated with a check mark in a column in front of the line style or weight. The active line style and weight are used when drawing elements. After they have been drawn, the line styles and weights of elements can be changed using the Set Line Style and Set Line Weight tools in the drafting tool palette. The Line Styles and Weights palettes have the standard feature and they appear by default when a new drafting window is opened. These palettes are discussed in subsection 5.15.1.

The **Line Styles** and **Line Weights** dialogs can be invoked from these palettes or from the respective items in the **Options** menu. They are shown in Figures 2.14.1.3 and 2.14.1.4. These are also discussed in subsection 5.15.1.
2.14.2 The Hatch Patterns palette

The Hatch Patterns palette, which is closed by default when a new drafting window is opened, is composed of boxes where a variety of patterns are displayed. It contains both vectorized and bit-mapped patterns, one of which can be selected as the active hatch pattern and is displayed in a large box contained in the palette (Figure 2.14.2.1). The active pattern is applied to hatch a drafting element when using the Hatch tool.

The Hatch Patterns palette includes a close box, title bar, hide box, and an orientation arrow. While it can be displayed in either a horizontal or a vertical orientation, it cannot be resized. Hatch patterns and their palette are discussed in more detail in section 5.16.

Figure 2.14.2.1:
The Hatch Patterns palette.
3.0 Introduction

At the top of the form•Z screen is a menu bar. It consists of ten menus. The Macintosh version has two more menus, the Apple (🍎) menu and the form•Z menu. Each menu contains a number of commands or items that can be selected in the standard manner. Many of the menu items have keyboard equivalents, which are indicated to the right of the item. Note that the key equivalents on the Macintosh and Windows are different, because of differences in the keyboards that these operating systems use. In this chapter the key equivalents for both systems are shown.

Types of menu items

There are four variations of menu items. The plain items execute an operation as soon as they are selected. Menu items with three dots at their end invoke dialog boxes. Menu items with a small arrow invoke submenus. Items with an asterisk at their end execute operations, but can also be used to invoke dialogs that affect the operations they execute. To do so, select the item while pressing the option key (Macintosh) or ctrl+shift (Windows).

Overview of menu items

On MacOS X, the Apple menu contains a set of system defined utility commands. Next to the Apple menu is the form•Z Application menu. This contains the items About form•Z, Preferences and the Quit item. The remainder of the items in this menu are standard items provided by OS X. On Windows, the About form•Z item is at the end of the Help menu. Selecting it displays the standard form•Z start-up image with information about the version of the program you are running. This will remain on the screen until the mouse is clicked or return on a Macintosh or enter on Windows is pressed.

The File menu contains commands for creating, opening, saving, and exporting files. It also contains items that support the printing functions and the Exit command on Windows. The Edit menu includes commands for undoing and redoing, cutting and pasting, global selection and clearing, and preference selection.

The Windows menu is used to open new windows, to close existing windows, and to control the grid, rulers, and underlay images. The bottom of the menu contains a list with the names of all the open windows. By selecting the name of an open window you make it the active window.
The Heights menu is only available in modeling and is used to set the height mode and to select the active height, used by the modeling environment. Fourteen preset heights are provided along with commands that can be used to change the preset heights and add new height values.

The View menu contains items for viewing models and drafting drawings. In modeling, it offers preset 3D views, all six Cartesian projections, arbitrary plane projections, and allows you to select among four types of 3D views. In drafting, it provides preset view rotations.

The Display menu controls the current viewing scale and screen display type. The Options menu is used to invoke dialogs that control a variety of system and window level options. The Palettes menu allows you to open and close palettes or to hide them. The Extensions menu (which is introduced by v.5.0) offers commands for managing, running, and debugging extensions. The Help menu allows you to access program and project information.

In addition to the menus that are entirely dimmed when they are not applicable, some individual items in other menus may also be dimmed when they do not apply. For example, in the Options menu items that affect modeling tools are dimmed when a drafting window is active, and vice versa.

**Contextual menus**

Frequently used items can be accessed through the contextual menu, which can be invoked by a proper click on the screen. This menu is designed to improve workflow by minimizing access to features that are available in dialogs or regular menus and is derived from the standard method for contextual menus established by the operating system.

Different methods for accessing the contextual menu are available for single-button and multi-button mice, as follows:

- When using a multi-button mouse, click on the right (secondary) mouse button while the cursor is over a modeling or drafting window. The menu can also be accessed by using the control + left (primary) mouse button (Macintosh) or ctrl + left (primary) mouse button (Windows).

- When using a single-button mouse, click while pressing control (Macintosh) or ctrl (Windows).

The contextual menu key combination can be changed to any desired combination with a mouse button, using the key shortcuts manager.
Dialog box conventions

The dialog boxes, through which the parametric options are set, use the standard features. The **alphanumeric** or **text fields** are text boxes in which numeric values and names can be typed and edited. Note that when numeric values are typed in, they are generally interpreted according to the currently selected unit of measurement, set through the Project Working Units dialog (see subsection 3.7.1). When English units are used, the inch (") and feet (') markings are significant. If not specified, the values are interpreted in inches.

The **check boxes** are small squares to the left of a descriptive name. They can be selected or deselected independently. Selection is indicated with an “X” mark in their box.

The **radio buttons** are groups of small circles with descriptive labels to their right or left. A radio button is selected by clicking on it. They are mutually exclusive within their group. Only one button can be selected at a time, and selecting one deselects the previous selection in the group.

The **buttons** are rounded rectangles that represent commands which relate to the individual dialogs. Most dialogs have a **Reset**, **Cancel** and an **OK** button. The latter two close the dialog. The **Cancel** button causes any changes made in the dialog to be ignored and, depending on the function of the dialog, it may terminate an operation. Pressing `command+period` (⌘ .) on the Macintosh or `ctrl+period` on Windows, or `esc` on either platform, while a dialog is open is equivalent to selecting its **Cancel** button. The **OK** button accepts the changes made in the dialog and applies the new parameters. The **Reset** button restores all of the fields in the dialog to the state they were in when the dialog was opened. This is equivalent to closing the dialog with cancel and then immediately re-invoking it. One of the buttons in every dialog is shown with a heavy border. This is the default button that is executed if the `return/enter` key is pressed.
3.1 The File menu

On Windows, this menu contains six groups of commands. It has one less on OS X, where the Quit command is found under the form·Z menu. The File menus for OS X and Windows are shown in Figure 3.1.0.1.

The first four groups execute project/file related operations.

• The first group contains commands for opening new and existing files and the second for saving them. These first two groups only apply to native form·Z files, which are listed in the next section.

• The items in the third group are for importing and exporting non-native files. Non-native are all the formats that allow files to be transported between form·Z and other applications or plug-ins. For a list of these formats see the next section.

• The fourth group contains commands for viewing files.

• The fifth group contains commands that support printing.

• The sixth group, on OS X, contains the single command Quit, and on Windows, the command Exit. They both terminate the current session of form·Z.
### 3.1.1 Files and naming conventions

The files or documents of form•Z are referred to as **projects**. A project consists of a modeling and a drafting environment. Every project has both environments, but it is not required that both be used. It is possible to have a project that has objects in the modeling environment but none in the drafting environment, and vice versa.

A graphics window is a view into a project’s modeling or drafting environment. Since these environments are independent of each other, form•Z provides two types of graphics windows.

A **modeling window** is a view into the project’s 3D modeling space, and a **drafting window** is a view into the 2D drafting space. A project may have more than one window of each type open simultaneously. Multiple windows of the same type and project are referred to as **associated windows**. Associated windows are views into the same world and are updated simultaneously as changes are made in the active window (see subsection 2.1.1). More than one project may be open simultaneously, but only one window of one project can be active.

form•Z includes a utility, called **form•Z Imager**, which allows you to render or print images in batch mode. This is done through an additional type of window that is available in the application, called **Imager Set** window. This is not a graphics window, but contains a list of the views to be rendered or printed. There can be one or more **Imager Set** windows open at one time, and they can be open at the same time other modeling and drafting windows are also open.

Each project window is titled with the name of the project, the index, and the type of window. The format of the title is `name-I [type]`, where `name` is the project name, `I` is the numeric index of the window for the project, and `type` is the word **Model** or the word **Draft**, which identifies the type of the window. The numeric index indicates the sequence in which the window was opened. When a new project is opened, the system labels it with `Untitled-I [type]`, where `I` is the count of new projects opened in the current work session and `type` is as above. A project is given a custom name the first time it is saved. This changes the title in all the project windows to the new name. For Imager set windows the same default names are used and are designated **Untitled Imager Set 1**.

form•Z is started or launched by either selecting the form•Z application (executable) that you wish to use, or files that you wish to open, or both. This is different for Macintosh and Windows and is discussed in your operating system manual. If form•Z is launched by selecting the application only, by default it creates a new form•Z project named **Untitled** and opens a modeling window with the default view into the world. Once a new project is saved for the first time, it is recognized by the system as a form•Z project file. If form•Z is launched by selecting a file or a group of files, the files are all opened as if the **Open...** item had been selected from the **File** menu and each file had been individually selected. The **Open...** command is discussed in more detail below.

When working with form•Z and disk files, there are two factors that are important: the name of the file and its type. The name identifies the file on the disk and its format or type indicates the kind of information that is contained in the file. Naming conventions are relative to the operating system in use. On OS X, there is no restriction on the length of file names, other than that they can not include the characters “:” or “/”. Likewise, under Windows there is no restriction on the length of the file name, but the names cannot include the characters “\", "\", or ":".

---

*Menus • The File menu*
The format or type of a file is used to determine how the data in the file is read. On the Macintosh, files have an internal file signature used to identify the file’s type. However, the file signature may not be unique enough to identify the file’s format. For example, all ASCII Text files on the Macintosh carry the “Text” signature. However, the contents of the file could be in a variety of file formats (DXF, IGES, 3DMF, etc.). In Text format, when the signature is not enough to recognize a format, then the file extension is used to determine a file’s type. The file extension is usually three characters following the last period in the file name. On Windows there is no internal file signature, therefore all files are identified by their unique extension. The list of file formats available in form•Z and their extensions are as follows:

**form•Z files**
- form•Z Project .fmz
- form•Z Symbol Library .zlb
- form•Z Imager Set .fis
- form•Z Project Backup .fzb
- form•Z Script .fsl
- form•Z Script Binary .fsb
- form•Z Suspended Rendering .sup

**form•Z support files**
- form•Z Animation .fan
- form•Z Extensions .zxt
- form•Z Geography .zge
- form•Z Help .zhl
- form•Z Heights .zht
- form•Z Icons .zic
- form•Z Materials .zmt
- form•Z Menus .zmn
- form•Z Preferences .zpf
- form•Z Defaults .zdf
- form•Z Shortcuts .sct

**Other format files**
- 3DGF (MacroMind) .3dg
- 3DMF (Apple’s QD3D Metafile) .3df
- 3DS (3D Studio) .3ds
- Art•lantis .opt
- BMP (Windows Bitmaps) .bmp
- Digital Elevation Model .cdo
- DWG (AutoCAD) .dwg
- DXF (AutoCAD) .dxf
- EPS (Encapsulated Postscript) .eps
- FACT (ElectricImage) .fac
- HDR (High dynamic range image) .hdri
- HPGL .plt
- IGES .igs
- Illustrator .ai
- JPEG .jpg
- Lightscape .lp
- LightWave .lwo
- Windows Metafile .wmf
- OBJ (WaveFront) .obj
- Open EXR .exr
- PICT (QuickDraw Picture) .pict
- Piranesi .epx
- PNG (Portable Network Graphics) .png
- QuickTime Image .qif
- QuickTime Movie .mov
- RIB (RenderMan) .rib
- SAT/SAB (ACIS entities) .sat
- Shockwave 3D .w3d
- STL (Rapid Prototyping) .stl
- Targa .tga
- TIFF .tif
- Viewpoint VET .mts
- VRML .wrl

form•Z automatically adds the file extensions for all formats on all platforms, when saving files. form•Z files are binary compatible between the Macintosh and Windows. Files saved on Macintosh can be directly opened on Windows and vice versa. Note, however, that the file extension is critical for recognizing file types on Windows. If you transfer a file that does not have an extension from a Macintosh to a Windows machine, you must add the file extension manually.
3.1.2 Creating and opening native files

**New [Model]**  
**New [Draft]**

These commands create and open a new, untitled project and a window into the project. The type of the first window opened is determined by the label [Model] or [Draft]. The new window becomes the active window and appears on the top of all other graphics windows.

**New Imager Set**

Selecting this command opens the **Imager Set** window, shown in Figure 3.1.2.1. In this window form•Z projects with saved views can be set up for batch rendering or printing by **form•Z Imager**. This is a rendering/plotting utility, which is incorporated in form•Z and is also available as an independent application. When independent there are two versions, one for the regular version of form•Z, and one for **form•Z RenderZone Plus**.

The **Imager** is discussed in more detail in section 3.12.

![Figure 3.1.2.1: The Imager Set window on the Macintosh.](image)

**New Script**

Selecting this item opens a new form•Z Script window and switches form•Z into the Script Editor environment. The Script Editor is an environment that allows you to edit and compile form•Z scripts without the need to leave the form•Z application. The Script Editor environment is discussed in section 3.8.1 of the form•Z SDK documentation. This documentation can be found in the PDF file "formZ SDK.pdf" which is installed in the "Doc" directory of the "form•Z SDK" directory by the form•Z SDK installer.
Open...

This command opens an existing form•Z native file. The document to be opened is selected through the Open: form•Z dialog, shown in Figure 3.1.2.2. How the available files are listed in the dialog depends on the following options:

**File Format**: This pop up menu contains one item labeled <Automatic> and an item for each of the form•Z formats (see previous section). Selecting <Automatic> lists all the form•Z native files. Selecting one of the other items causes only the files of that format to be listed and selectable, while all the other files are listed dimmed.

**Show All Files**: When this Macintosh option is on, all files, including non-native form•Z files, are listed as selectable.

**List Files Of Type**: This Windows menu has the same effect with above. It can be used to view all files (*.*) or just one of the supported formats (*.fmz for example). This option is useful when a file may have been imported from a different platform and may not be tagged properly.

![Figure 3.1.2.2: The Open dialog on (a) OS X and (b) Windows.](image)
The format of a file is recognized by the tag assigned to it by the operating system. Text files such as DXF, IGES, 3DMF and 3DGF are distinguished by their extension. On the Macintosh, when an extension is not found, files can be identified by information contained in their headers. It is possible that text files saved in another platform may not be tagged as text files when imported into the Macintosh environment. In such cases the system is not able to recognize their format and to list them together with the other files of the same format. The Show All Files option allows you to bypass this limitation. For example, if you know that a certain file is of format A and it is not listed when format A is selected, then selection of the Show All Files option will force the file to be listed and subsequently to be opened. Note, however, that if you are mistaken and the file you attempt to open under format A is not of format A, then the attempt to open it will fail.

When a form·Z project file is opened, it becomes the active project, and all of its windows are opened as they were when the file was saved. The active window is the window that was active when the project was last saved. However, this can be changed by the following option:

Add To Project: This option is enabled when selecting form·Z project files. When selected, the content of the file is added to the current project rather than opening the project on its own. This item is dimmed if there is no open project or the current window is not a project window.

Open Recent

This item is a pop out menu that contains a list of recently referenced files that were opened with the Open command, or saved with the Save, Save As..., or Save A Copy As... commands. This menu provides a shortcut to opening recently used files instead of using the Open command and the standard Open dialog to locate the file. To open any recent file, simply select it from the menu to automatically locate and open it. The options that affect this pop out menu are set in the Recent Files Preferences dialog accessed through the Preferences dialog (see section 3.2.7).
3.1.3 Closing and saving native files

Close

This command closes the currently active project or Imager set and, hence, all its windows. If changes were made to the project or Imager set since last saved, an alert box is presented and gives you the opportunity to save the changes before closing. If it was not saved before, the form·Z Save Project As... (Figure 3.1.3.1) or Save Imager Set As... dialog is presented before closing. If the project or Imager set being closed is the only open project or Imager set, there will be no windows left on the screen after the project is closed. The system hides all the open palettes and deactivates most of the menu items. A new or existing project must be opened before any work can continue. If there are other open projects, then the front-most window of the remaining projects becomes the active window.

Save

This item saves the active project into a disk file, including both its modeling and drafting environments, or an Imager set. The project remains open, and its current window remains active. If you save the project or Imager set for the first time, the Save As... dialog (Figure 3.1.3.1) is presented to allow you to name the file before it is saved. If the file was saved once, it is saved under the same name, replacing the previously saved file on the disk.

If the Keep Backup option is selected in the Project: General section of the Preferences dialog (see subsection 3.2.7), then the existing file is renamed with a .fzb extension and the previous .fzb file is removed (if it exists).
Save As...

This is normally the command to be executed when a form•Z project or Imager set is saved for the first time, or saved with a different name. This command invokes the form•Z Save As... dialog (Figure 3.1.3.1) and a file is saved under the name you type in the name field. After typing a name, clicking on the Save button produces a file immediately. Using this command with a project that has already been saved and named effectively makes a copy of it with the new name. The previously saved file remains intact with the original name.

When saving form•Z project files, by default the Undo and Redo records are cleared as soon as you save the file. However, you can tell form•Z to save these records with the project through the Undo Options dialog which is accessed from the Edit menu (see section 3.2). Other options that affect the saving process can be found in the Save As dialog, as follows:

Keep Backup: When this option is selected (default), the system makes a backup copy of the previously saved project file before saving it again. Thus the previous file is transformed into a backup file and the project is saved into a new file. This feature constitutes a safety valve against instances when the system is unable to find enough disk space to save a project or when some other irregularity is encountered. The names of the backup files are constructed by attaching the suffix .fzb to the end of the name of the file.

Compress: Selecting this option compresses the saved project file. The default is off. Compressed files take less disk space but require more time each time you save. The Keep Backup and the Compress options can also be selected from the Preferences dialog (see subsection 3.2.7).

Figure 3.1.3.1: The Save Project As... dialog (a) for OS X and (b) for Windows.
Save A Copy As...
This command is used to save a copy of the active project, regardless of whether the active project has already been saved or not. It resembles the Save As... item in that it invokes a dialog and requires you to enter a name and location under which the project will be saved. It is also different from the Save As... item in a few major ways.

- It does not affect the currently active project as it appears on the screen. That is, while Save As... changes the name in the title bar of the window, Save A Copy As... does not. If the displayed title is Untitled it remains the same after the execution of the Save A Copy As... command. Likewise, if the active project has already been saved, it retains the name under which it has been saved.

- Save A Copy As... only applies to form•Z file.

- When the Save Project As... dialog is invoked from the Save A Copy As... command, clicking on the Options... button invokes the form•Z Project Options dialog (Figure 3.1.3.3). The options in this dialog offer the ability to save parts of a project rather than the complete project as with the Save As... command. These options are discussed below.

Save As Version: This pop up menu allows you to select which version of form•Z you want to save to. The default is the current version, but it is possible to also backsave to earlier versions. If there are incompatibilities of data between versions, alerts are posted to inform you that you may lose some information. The pop up menu lists only versions of form•Z in which the file format actually changed. form•Z file formats between the listed versions are always compatible with the immediately preceding file version.

Save: Modeling/Drafting: This group of options allows you to select only parts of the project to be saved. The selections can be made for the modeling and drafting environments independently.

All Entities: When this option is selected (default), all entities are saved to the file. This option is always available.

Visible Entities Only: This option is available when at least one entity in the modeling or the drafting environment is a visible entity, otherwise the option is dimmed. A visible entity is an object or element that is neither ghosted nor on an invisible or ghosted layer. When this option is selected, only the entities that are visible are saved.
Picked Entities Only:  This option is available when at least one object or group in modeling or one element in drafting is selected.  When this option is on, only the selected entities will be saved.  Only complete objects and elements can be saved.  Entities selected on the other topological levels (point, segment, outline, face, hole/volume) do not qualify for this option.

None:  When this option is on, no entity of the respective environment is saved.  All other data is saved.  This option allows you to save only the modeling or drafting part of a project.

When backsaving to form•Z file version 2.8.0, only the All Entities and the None options are available.  It is not possible to save only visible or picked entities to form•Z version 2.8.0 files.

Save Symbol Libraries:  This option is available when there are symbol libraries attached to the project.  When it is selected, copies of the attached symbol libraries are saved to the same folder that is selected in the Save Project As... dialog.  The copies are saved in the same form•Z version that is selected from the Save As Version pop up menu.

All Attached Libraries:  When this option is selected, all attached libraries are copied.

Libraries With Referenced Definitions Only:  This option is available when the project contains symbol instances.  When it is selected, only symbol libraries containing symbol definitions which are referenced by symbol instances are saved.

All Symbol Definitions:  When this option is selected, each symbol library containing referenced definitions is copied with all its definitions.

Referenced Symbol Definitions Only:  With this option, a symbol definition is saved to a symbol library only if it is referenced by at least one symbol instance saved to the file.

Save Image Files:  When a copy of a project is saved with this option on, all the image files that are used and linked in the project are also copied in the same folder with the project.  These are images used as textures in a surface style (with color, transparency, or bump shaders), as backgrounds, environments, displacement maps, underlays, or image elements in drafting.

Referenced Files Only:  When this option is on (default), only images that are actually referenced are copied.  For example, if an image is used in a surface style, but the surface style is not used in the project, that image is not saved when this option is on.

Save Font Files:  When saving a copy with this option on, the font files also will be copied.

Referenced Files Only:  When this option is on (default), only the fonts that are actually used will be saved.  If off, all the fonts that appear in the Font menu of the Text Editor dialog will be saved.

Revert To Saved

This item returns the current project to the state it was in at the time of the last save.  This effectively cancels all the changes made to the project since the last save.  The system presents an alert box and requests confirmation before the command is completed.  This operation also resets the Undo and Redo records.  Revert To Saved can not be used when an Imager Set is the active window and hence it appears dimmed.
3.1.4 Importing and exporting non-native files

The commands in the third group of the File menu apply to non-native formats only.

**Import...**

This item is used for opening files in non-native form•Z formats. When selected, it invokes the **Import** dialog, shown in Figure 3.1.4.1.

**File Format**: This pop up menu contains an item labeled **<Automatic>** and an item for each of the non-native formats enabled by form•Z. They work as in the **Open** dialog. In form•Z, the file formats are handled by translators, which are plugins and are controlled by the **Plugins And Scripts** dialog accessible from the **Edit** menu.

**Options...**: Clicking on this button invokes the **Options** dialog for the file format selected from the **File Format** pop up menu. These dialogs are discussed in sections 3.13 and 3.14. The button is dimmed if **<Automatic>** or a file format that has no export options is selected.

**Show All Files, Add To Project**: These options are as in the **Open** dialog.

**Import Model**: When this option is selected, the imported file is treated as 3D modeling data and imported into the modeling portion of a project. This item is dimmed if the format of the selected file does not support 3D modeling data or the form•Z modeling module is not installed.

**Import Draft**: When this option is selected, the imported file is treated as 2D drafting data and imported into the drafting portion of a project. This item is dimmed if the format of the selected file does not support 2D data or the form•Z drafting module is not installed.

**Import Recent**

This menu item works as the **Open Recent** item, except that it displays files of non-native formats that were recently imported. It invokes the **Import Recent Options** dialog shown in Figure 3.1.4.2. The options in this dialog are the same as those in the **Import** dialog, and are as discussed above.

*Figure 3.1.4.1: The Import dialog.*

*Figure 3.1.4.2: The Import Recent Options dialog.*
Export

Placing the mouse cursor on this item invokes the **Export** menu that displays all the available formats for exporting the current window. This menu is context sensitive and displays the available formats for the active window. If the active window is a modeling window, the menu displays all the formats that support 3D modeling data, as shown in Figure 3.1.4.3. If the active window is a drafting window, the menu lists all the formats that support drafting data. If the active window is not a form•Z project window, this item is disabled.

Export Image

Placing the mouse cursor on this item invokes a menu, as shown in Figure 3.1.4.4. It displays all of the available image formats that can be exported from the current window. Image formats are those that result in a file which represents the image on the screen rather than the real object data. If the active window is not a form•Z project window, this item is disabled.

Export Animation...

Selecting this command converts an animation from the form•Z animation format (.fan) into either a QuickTime Movie (MacOS and Windows) or an AVI file (Windows only). For best results use the latest version of Quicktime available.

When this item is selected, a standard **Open** dialog is presented for selecting the .fan file to be exported. After identifying the file, click on **Open**. The standard **Save As** dialog is invoked where you can type in the name and select either **QuickTime** or **AVI**. After selecting the desired location, press **Save** to export the movie. A progress bar shows the progress as each frame of the animation is exported.

When exporting to QuickTime, the **Options...** button accesses the standard QuickTime compression dialog. Since QuickTime supports a variety of compression types and can be expanded with third party plug-ins, this dialog's appearance may vary. For more details on Quicktime, visit www.quicktime.com.
3.1.5 Viewing image files

**View File...**

This command allows you to open and view images saved in any of the supported image formats (except Illustrator), QuickTime and QuickTime VR movies, AVI movies and form•Z animation files (.fan). Selecting this item invokes the Open File dialog, in which the file to be displayed is selected. Upon closing the Open File dialog the View File window shown in Figure 3.1.5.1 is invoked, and displays the image, movie or model contained in the file.

This window has all the standard features of a window and can be resized. In addition, under its title bar, it contains two image manipulation tools and displays the size of the image in pixels.

You select an image manipulation tool by clicking on it. When you place the mouse inside the View File window, its cursor changes to an icon that corresponds to the image manipulation tool currently selected.

- **Zoom In:** With this tool selected, clicking inside the image enlarges the image. If the size of the enlarged image exceeds the size of the window, the window is also enlarged (up to the limits of the screen) to fit the whole image.

- **Zoom Out:** The Zoom In tool changes to this tool when pressing the *option* (MacOS) or *ctrl+shift* (Windows). When selected and you click on the image, the size of the image is reduced.

- **Pan:** With this tool selected, when you click and drag the mouse, you move the image within the window. The effect of this tool is identical to the effect of the scrolling bars. You can pan or scroll an image only when it extends beyond the limits of the window. When the complete image is displayed in the window, this tool is dimmed and inactive.

**View Recent**

Placing the mouse cursor on this item invokes a menu that shows all recent viewable files that can be opened with the **View File...** command. This includes all exported image files, QuickTime files and files previously selected with the **View File...** item.
### 3.1.6 Printing

Prints can be made from both modeling and drafting. **form-Z** offers direct support for printing documents on a variety of postscript or non-postscript printers using the standard drivers provided with your operating system or by the printer manufacturer. Plotter support is provided through third party drivers.

The print outs produced by **form-Z** are based on what is displayed in the active window or on the screen. If the active window is a modeling window, then the modeling objects are printed in their current view and according to their current type of display, which may be a line drawing or a shaded rendering. If the active window is a drafting window, then only line drawings are printed. If the active window is an Imager set, the printing menu items are dimmed and inactive. Printing controls for the Imager set are handled by items in the **Imager Set** window (see section 3.12).

The user controls the size and the direction of the printer paper as well as a number of options that refer to the scale, extent, and justification of the drawing relative to the printer paper. If the drawing is sufficiently small, a single sheet of paper will be used. If it is larger than what a single sheet can hold, the system will tile as many sheets as necessary to fit the complete drawing. The direction of the tiles depends on the paper direction selected in the **Setup** dialog (Figure 3.1.6.1). The **Page Preview** command is very handy for reviewing the layout produced by the system before sending it to the printer.

**Page Setup...**

This item displays the **Setup** dialog (Figure 3.1.6.1) for the current printer. This dialog allows you to specify parameters such as the paper size and printing orientation. The **Setup** parameters work in conjunction with the parameters set in the **Plot/Print Setup** dialog (Figure 3.1.6.2).

![Page Setup dialog](a)

![Page Setup dialog](b)

**Figure 3.1.6.1:** The **Page Setup** dialog (a) for a printer on OS X and (b) a printer on Windows XP.
Plot/Print Setup...

This item displays the form•Z Plot/Print Setup dialog, shown in Figure 3.1.6.2, that allows you to specify the parameters by which the drawing will be scaled and positioned on one or more sheets of paper. These parameters work in conjunction with the Paper size and Orientation parameters in the Page Setup dialog.

Plot Scale: The scale of the drawing is entered in this field. The initial default value is 1/16" = 1'-0" (or 1:200 in metric).

Scale To Fit Media: When this option is selected, the drawing is automatically scaled to fit on a single sheet of paper (and the plot scale is ignored).

Plot/Print Type: The four options in this group control what portion of the image shown in the graphics window will be printed or plotted.

Extents: When this option is selected, all the objects or elements in the project are printed regardless of what is visible in the window.

Window Contents: When selected, only the objects or elements that are visible in the window are printed. This option prints the window in a “what you see is what you get” fashion.

When the Window Contents button is selected and a hidden line plot or a scan line rendering is displayed, the system prints the screen image from the display buffer. The display buffer is maintained by the system in order to minimize the need to recalculate the image, when it needs to be refreshed. The display buffer speeds up the printing process significantly. The use of the display buffer is also noticeable when a Page Preview is executed. The previewed image is shown as it is displayed on the screen, rather than as a wire frame plot.

Dump Window, Dump Screen: These options provide the ability to print a bit-mapped image of the active window or the entire screen, respectively. This style of printing gives a much lower resolution print but is often faster than the other printing options. The bit-mapped printing ignores the current Plot Scale and the Scale To Fit Media options. Bit-mapped images can be scaled using the Reduce or Enlarge percentage in the Setup dialog (Figure 3.1.6.1), provided the currently connected printer supports such operations.
X, Y Justification: These two groups of options control where the drawing will be positioned relative to the edges and center of the paper. The default is justification at Center, for both directions.

Image Options: This group of options controls how the image appears on the page. These options are disabled when Dump Window or Dump Screen are selected.

Plot Grid, Plot Axis: These options control whether the reference grid and/or the Cartesian axes will be included in the print. Note that whether or not these are printed is independent from whether they are currently displayed in the window.

Include Background: When this option is selected, the background of the plotted/printed image is filled with the background color. This option is most useful when using color printers. It is off by default.

Print Text As Paths: This option affects how TrueType and PostScript text in the drafting environment is plotted/printed. When this option is selected, form\textsc{Z} sends the outline of each character in a text element to the printer/plotter. When this option is off, which is the default, form\textsc{Z} requests the printer to draw the characters using its own native graphics, which could be PostScript, TrueType, Quickdraw, etc., depending on the operating system and the plotter or printer. In general, this method will give the best looking text results, although there may be slight differences between the printed output and the image on the screen. When using a plotter, or when it is important that the printed image match the screen image exactly, the Print Text As Paths option should be selected. Since plotter/printer resolution is usually higher than the screen resolution, for smoother text, also select the Override Text Smoothness option (Drafting Display Options dialog), and increase the outline smoothness using the slider control.

Solid Color Printing: When this option is off, which is the default, color polygons are printed filled with a pixel pattern based on the color of the polygon. When this option is on, color polygons are printed filled with a solid color instead of a pixel pattern.

Keeping the Solid Color Printing off will give the best results on most printers since their drivers can take the pixel pattern and convert it into the proper image information. However, some printers either do not support the printing of pixel patterns or they may not print them in the proper color. To determine which method works best for your particular printer you will need to run test prints with this option on and off and then choose the setting based on which printed output you prefer. In general, printers that have software built into their driver designed to optimize color and dithering will work better with Solid Color Printing on. Note that this option may affect some black and white printers as well since the drivers may convert pixel patterns and solid colors to gray scales differently.

Image Detail: This pull down menu affects prints made from the drafting environment only and determines how images that have been imported or pasted as image elements into a drafting project are printed. Three options are available (Figure 3.1.6.3):
**Bounding Box:** When this item is selected image elements are printed as a rectangle that bounds the image with two diagonal lines crossing the rectangle.

**Preview:** When this item is selected, the low resolution image stored in the image element is used for printing.

**High Resolution:** This item, which is on by default, produces the most accurate print by using the full resolution of an image element. An image element may either be stored in a project file or may be stored in a separate file which is linked to the project file. These options are available when an image element is pasted or imported. When an image is stored with the file, the print is produced faster. If the image is linked but the system fails to find the file of the linked image, a preview level image is printed.

**Page Options:** This group of options offers additional printing features that appear outside of the image on the page.

**Crop Marks:** When this option is selected, crop marks are also printed on the four corners of each page, to facilitate collating the pages together. This is particularly useful when more than one page is used to print a drawing (tiled pages). When crop marks are printed, the usable area of the paper is reduced by 1/4" on each of its four sides.

**File Name & Date:** With this option selected, the system will also print the following information, with the drawing:

- The *name* of the file, *time* and *date* are printed on the upper left edge of the paper.
- The *page number* is printed on the upper right edge of the page.
  For example: Page 1.
- The *display type*, *view type*, *print type*, and *scale* are printed on the lower left edge.
  For example: Wire frame Perspective of Window Contents  Scale: 1/8" = 1'-0"
  or Hidden line Axonometric of Extents Scaled To Fit Page.

This text is just outside the crop marks. When crop marks are not printed, then this text is just inside the edges of the page and will overlap with portions of a drawing which may be close to the edge of the paper.

**Overlap Pages:** When selected and a drawing is printed on more than one page (tiled pages), the neighboring ends of the pages overlap by 1/4", to facilitate assembling them. When the *Crop Marks* option is also selected, then two sets of marks are printed on the corners of the pages. The second set of marks is distinguished from the first by an “X” printed on the crop mark.

**Frame:** When this option is selected, a rectangular frame is printed around the image. When printing with the *Extents* option selected, the frame is drawn at the limits of the page or pages being printed. When printing with the *Window Contents* option on, the frame is drawn at the limits of the window.
The line weight (width) of the frame may be selected through a pop up lines menu located next to the Frame option, as shown in Figure 3.1.6.4. Any one of 11 line weights may be selected. The first is a “hairline” that is printed at whatever the thinnest possible line on a particular printer is, and may vary from printer to printer. The second line weight is 1/2 point (where 1 point = 1/72”). The remaining weights range from 1 to 9 points. The line menu is available when Frame is selected. It is dimmed and inactive otherwise.

**Page Preview...**

This item shows a miniature view of what the document will look like, when printed. This view is scaled to fit within the size of the screen. The total printed area is represented by the rectangle colored with the background color of the current window. If the drawing is tiled, the page preview shows the page breaks as dashed lines. Examples are shown in Figure 3.1.6.5.

**Figure 3.1.6.5:** Page preview for (a) a single page and (b) a tiled image.

The Page Setup... and Print... buttons perform the same operations as the menu items with the same names in the File menu. After the setup or printing is complete, it returns to the Plot/Print Setup dialog. These buttons are useful to inspect the effect of the printing options without exiting the dialog.


**Print...**

This item displays the Print dialog box (Figure 3.1.6.6) for the current printer. This dialog provides the options for controlling the number of pages and copies printed. Clicking the **OK** button accepts the print parameters and sends the document to the printer. While the document is printing, the page preview display is shown on the screen.

![Image of the Print dialog box](image)

**Figure 3.1.6.6:** The **Print** dialog for (a) a printer on OS X and (b) a printer on Windows XP.

### 3.1.7 Terminating the session

A session with **form-Z** running under Windows is concluded by selecting, **Exit**, the last command in the **File** menu.

The **Quit** item terminates the current session of **form-Z** on a Macintosh. On OS X the **Quit** command is found in the **form-Z** Application menu.

Before termination, the system sequentially closes all open projects and the Imager set, if open. If changes were made since the last time you saved, the system asks you if you wish to save before quitting.
3.2 The Edit menu

The **Edit** menu, shown in Figure 3.2.0.1, consists of four groups of commands.

- The first group controls the undo, redo, and replay operations.
- The second contains the commands that provide the ability to cut, copy, and paste entities from one window or project to another, to duplicate objects, and to inspect the **form•Z** Clipboard. This functions like the standard system clipboard and is used as a temporary storage space for moving objects and/or elements from one project window to another.
- The third group offers commands for picking all the ghosted or unghosted entities and for deleting all the objects or elements of a project. It also contains commands for deselecting all and for reselecting previously selected entities.
- The fourth group contains the ability to manage **form•Z**’s key shortcuts and to define and save preferences.

![Edit menu on (a) Macintosh and (b) Windows.](image)

**Figure 3.2.0.1:** The **Edit** menu on (a) Macintosh and (b) Windows.
3.2.1 Undoing and redoing

The undo and redo operations in form•Z are designed to encourage experimentation by eliminating the fear of making mistakes. This is accomplished by allowing an unlimited number of these operations.

Only operations that change modeling objects, groups, lights, surface styles, modeling views or drafting elements can be undone and redone. The Undo and Redo commands do not affect operations that change the graphic environment, select options, or set parametric values. The undo/redo operations rely on the undo records which they generate and store on the scratch disk, which are controlled by the Scratch Disk section of the Preferences dialog (see subsection 3.2.7).

**Undo**

Selecting this command cancels the effect of the previous operation and returns the objects or elements to their previous state. A sequence of operations can be undone, and are undone in reverse order from which they were executed. In the menu, this item appears with the name of the operation that will be undone next, (i.e. Undo Extruded Object).

By default, previously executed operations can be undone up to the point where the system last reset the project’s undo records. The system starts with a fresh undo record when a new or an existing project is opened, and resets it each time the project is saved. However, you can change the form•Z preferences to save the undo records with the project. Each project has its own undo record, hence, the undo record is not reset when the project is deactivated or activated. If the undo record is reset or empty, the Undo command appears dimmed and cannot be selected.

When you select Undo while pressing the option key (Macintosh) or ctrl + shift (Windows), the Undo List dialog (Figure 3.2.1.1(a)) is invoked. The operations are listed with the latest on top and first on the bottom. This allows you to select a group of operations to be undone in one step. You make the selection by clicking on the operation up to which you wish to undo. Note that the operations can only be undone in sequential order, starting at the top of the list. The list is displayed in a scrollable mini-window, which allows you to scroll up and down in the usual manner.
Redo*

This command reverses the effect of the most recent Undo* command and returns the objects or elements to their previous state. It can be executed repeatedly to redo a sequence of undone operations. The Redo* item can only be executed immediately after an undo or sequence of undo operations have been executed. In the menu, Redo* appears with the name of the operation to be undone. The redo record is reset when any tool is used or when a window is deactivated. When the Redo* command is not available, it is dimmed and inactive.

When you select Redo* while pressing the option key (Macintosh) or ctrl + shift (Windows), the Redo List dialog is invoked. This works as the Undo List.

Redo All

This command reverses a continuous sequence of undos and returns the objects or the elements to the state they were in before the first undo operation. A sequence of undos is continuous when it has not been interrupted by any other operation. Only a single Redo All per sequence of undos is possible. This command can only be selected following the execution of an Undo command or a sequence of such commands. When the Redo All command is not available, it is dimmed and inactive.

Replay

This command clears the screen and plays back all the operations executed since the undo record was last reset. The operations are shown in order of execution, from the first to the latest operation. Holding the mouse button down pauses the playback for visual inspection; releasing it resumes the playback. The Replay command can be executed whenever the Undo command is active. When the Replay command is not available, it is dimmed and inactive.

Reset Undo/Redo

The Reset Undo/Redo command allows you to manage your undo/redo records. When selected it will clear these records, and is recommended when there will be no need to return to early operations. It can be used to free disk space.

The Undo and Redo commands rely on records that they automatically generate and store onto disk space. These records are reset (erased) each time a project is saved. The system will also reset the undo/redo records whenever it runs out of disk space. It always issues a warning message before erasing the records. The system tends to run out of disk space when a user is working on a large project and only limited space is available on the disk. A typical recommendation is that at least 10 MB of disk space be available when you start a form-Z session.
Undo Options...

Selecting this item invokes the Undo Options dialog shown in Figure 3.2.1.2. This dialog allows you to specify the depth of the undo records, which applies to all projects, not just the active project.

Use Undos: This checkbox is on by default and offers the ability to undo at a depth specified by the following options. When it is off, undos will not be available and no records will be maintained.

Unlimited Undos: When this option is selected, form-Z does not limit the number of undos that are available since the project was started. This option is on by default.

Last n Operations: When this option is selected, only the specified number of operations are available to be undone. Operations that were executed before that number of operations cannot be undone anymore.

At Most n KB Disk Space: This option allows you to limit the undo records to the specified amount of disk space. Once the specified space limit is reached, earlier operations are overwritten and cannot be undone.

Last n Minutes: When this option is selected form-Z limits the undos to operations that are not older than the specified number of minutes. Operations executed after more than n minutes have passed cannot be undone.

Undo List Shows n Operations: This parameter allows you to instruct the program how many items it should list in the Undo List dialog. The desired number is entered in the numeric field.

Save Undos in Project: With this option on, the undo records are saved with the project file. When opening the project file again, the undos are available, and operations executed in the previous session may be undone. Note, however, that undo records are not cross-platform and they are only available when the project is opened in the same platform (Macintosh or Windows in which the undos were saved. The time limit for the undos applies to the accumulated time that the project was open. Note that, when undos are saved to the project file, form-Z cannot clean up obsolete data that may have accumulated during the session. Therefore, when the Save Undos In Project option is selected, project files may grow quite a bit. Exactly how much larger the project file will be when the undo records are also saved with it, also depends on the undo options that affect the depth of the records.

Reset After Saving Project: When this option is selected, form-Z resets the Undo every time a project is saved.

Figure 3.2.1.2: The Undo Options dialog.
3.2.2 Cutting, pasting, and duplicating

This group of commands provides the standard cut, copy, and paste features. The effect of these commands depends on the type of the environment involved. Objects can be transferred between modeling windows, and drafting elements can be transferred between drafting windows of different projects. It is also possible to transport images of modeling objects to the drafting environment and drafting elements to the modeling environment. The system automatically adjusts the internal representations when entities are transported between different environments.

The cut, copy, and paste operations, whether they transport entities between the same or different types of windows, use the form•Z Clipboard, which has a modeling and a drafting part. Objects cut or copied from a modeling window are stored in the Clipboard's modeling part and vice versa. The contents of the form•Z Clipboard may be inspected through the form•Z Clipboard... command, discussed toward the end of this section.

Cut

This command deletes all the currently picked objects or elements from the active window and places them in the form•Z Clipboard. The Cut item requires that objects and elements be pre-selected before it is executed. Only entities picked at the object or element level are affected by Cut. Entities picked at other levels are ignored.

The Cut command replaces the previous contents of the form•Z Clipboard with the objects or elements that are cut. This replacement discriminates between environment types. Modeling objects replace all the objects that may be in the Clipboard’s modeling part and drafting elements replace all the elements in the Clipboard’s drafting part. As a delete operation, the Cut item is complemented by the Clear items discussed in subsection 3.2.5 and the Delete tools in the tool palettes, discussed in section 4.25 for modeling, and in section 5.17 for drafting.

Copy

The Copy command is identical to the Cut command, except that it does not delete the objects, elements, or text when they are placed into the Clipboard. The Copy command also requires objects and elements to be prepicked. The elements that were picked for the operation remain picked after its execution.
Paste

This item pastes the contents of the form•Z Clipboard into the active window. Modeling objects are pasted from the Clipboard’s modeling part when a modeling window is active, and drafting elements are pasted from the Clipboard’s drafting part.

The Paste command can be used as a pure paste or as a replacement operation. If no objects or elements are picked in the active window, the contents of the Clipboard are simply added to the project, without affecting any of the other entities in the project. If the active window contains picked objects or elements, these entities are replaced by the entities brought in. The elements brought in preserve the position they had at the time they were placed in the Clipboard. Pasted objects and elements appear selected and highlighted to facilitate immediate operations on them. The Paste command has no effect on the contents of the Clipboard, which can be pasted any number of times into the same or a different project.

Duplicate*

This item allows you to make copies of modeling objects and groups of objects, or drafting elements within the same project in one step and directly, without first placing an entity in the Clipboard. In form•Z modeling objects and drafting elements can also be copied by using the Move tool with the Self/Copy modifier set to Copy (see section 4.23.1).

To make copies of objects using the Duplicate* item, preselect the objects/elements you wish to copy (using the Pick operator with the topological level set to Object or Element) and then select the Duplicate* item. The operation is executed immediately and the new object(s)/elements(s) appear on the screen. If the default offset values are used, the new objects/elements will be moved by 2' - 0” in the X and Y directions.
Duplication Offset...
The offset value can be set in the Object Duplication Offset dialog for modeling (Figure 3.2.2.1) and in the Element Duplication Offset dialog for drafting (Figure 3.2.2.2). These dialogs can be invoked directly from the Duplicate* item, by selecting it while pressing the option key (Macintosh) or ctrl+shift (Windows) or by selecting the Duplication Offset... line directly below the Duplicate* item. Which of the two dialogs will appear depends on the type of the active window.

X, Y, Z (modeling) and X, Y (drafting): The desired offset value is entered in these fields, independently for each direction; assuming the following option is not selected.

XYZ / XY Offset Lock: Selection of this option constrains the offset fields to the same value. When one is entered in one field, the content of the other fields is automatically adjusted. When deselected, independent offset values can be entered for each direction. This option defaults to on for drafting and to off for modeling.

The new duplicated entities appear selected and highlighted, while the original entities are deselected. The Duplicate* operation can be executed sequentially any number of times. Only entities selected as objects or groups (or elements in the drafting) will be duplicated. Entities selected at topological levels lower than Object (modeling) or Element (drafting) will be skipped by the operation and will be deselected.

Figure 3.2.2.1: The Object Duplication Offset dialog (modeling).

Figure 3.2.2.2: The Element Duplication Offset dialog (drafting).
Paste From Modeling...
Paste From Drafting...

This is an environment sensitive menu item. When the active window is a drafting window, it reads as Paste From Modeling.... It appears as Paste From Drafting... when a modeling window is active. These commands allow entities to be transported from modeling to drafting, and vice versa. They can be transported either by placing them in the Clipboard or directly from one type of window to the other. When this is done, their internal structures are translated to the environment of the destination window. When an object is transported from a modeling to a drafting window, a view of that object is actually converted into a two-dimensional drawing. When a drafting element is transported into the modeling environment, it becomes a surface object and is placed on the current reference plane.

Any modeling object can be transported into the drafting environment. The view and display type at the time objects are transported determine how they will be transferred to the drafting environment.

The Paste From Modeling... and Paste From Drafting... items cannot be used for replacement operations. That is, they have no effect on any entity that may be highlighted at the time the pasting from the other environment occurs.

When Paste From Modeling... is selected the Paste From Modeling dialog, shown in Figure 3.2.2.3, is invoked. This dialog displays a list of windows that contain objects rendered with a vector or polygon renderer (Wire Frame, Hidden Line, Doodle, Quick Paint, or Surface Render). These are the objects that can be pasted into the drafting window. The list consists of three columns. The first column is the name of the window as it appears in the window’s title bar. The second column is the window’s render type and the third column is the window’s view type.

![Paste From Modeling Dialog](image)

**Figure 3.2.2.3:** The Paste From Modeling dialog.

If objects have been copied into the form·Z Clipboard, using the Cut or Copy command, the form·Z Clipboard will appear at the top of the list. Selecting an item in this list will specify the objects, the rendering, and view used to create the draft elements. Only objects within the window’s view (or contained in the form·Z Clipboard, if selected) will be pasted into the drafting window. Objects pasted from the clipboard will always be pasted using the Wire Frame render type.

The following options can be set in the Paste From Modeling dialog and they determine how a view from modeling will be converted to a drafting drawing.
Paste 3D Faces As: This group of options controls how the modeling entities will be treated.

Each Face/Outline As A Polyline: When this option is selected (default) the closed outlines of the modeling faces/holes are transported into drafting as single, closed polylines.

Generate Hatches: This option fills polylines with a solid hatch element. This option is only available for polygon renderings (Quick Paint, or Surface Render).

Make Compounds: When this option is selected, pasted elements are grouped as compounds, as follows:

From Objects: When this option is on, pasted elements from the same object are grouped as a compound.

From Faces: When this option is on, pasted elements from the same face are grouped as a compound.

Each 3D Segment As A Single Line: This option results in each segment of the modeling view becoming an independent drafting element.

Remove Duplicate And Overlapping Lines: When this option is selected, the system filters all duplicate and overlapping lines. Duplicate lines are line segments with identical end points. Overlapping lines are coincident segments that lie partially on each other, but their endpoints are not necessarily coincident. Such overlapping and/or duplicate lines result from the fact that each edge of a solid object belongs to two distinct faces and, in the general case, it is drawn twice.

Join All Segments: When this option is selected, all pasted line segments with coincident end points are joined.

Tolerance: This specifies how close segment end points must be for them to be joined.

Make Short Segments Point Objects: When this option is on, line segments that are shorter than the length specified in Minimum Segment Length are converted to point elements.

Minimum Segment Length: This specifies a length up to which line segments are converted to point elements.

When Paste From Drafting... is selected the Paste From Drafting dialog, shown in Figure 3.2.2.4, is invoked. This dialog displays a list of projects that contain drafting elements. These are the elements that can be pasted into the modeling window. If objects have been copied into the form•Z Clipboard, using the Cut or Copy command, the form•Z Clipboard will appear at the top of the list. Selecting an item in this list will specify the draft elements used to create the modeling objects.

Figure 3.2.2.4: The Paste From Drafting dialog.
3.2.3 The form•Z Clipboard

A menu item is available that allows you to inspect the contents of the form•Z Clipboard. Both the modeling and the drafting portions of the Clipboard can be visited either by switching windows before selecting the item, or from within the dialog invoked by the item.

Show form•Z Clipboard...

Upon selection of this item, a dialog is opened which shows the objects or elements in the form•Z Clipboard (Figure 3.2.3.1). Whether the Clipboard's modeling or drafting part is initially shown depends on the type of window active. The Modeling and Drafting radio buttons can also be used to switch between the two parts of the Clipboard. The modeling environment is by default initially viewed from the \( z = 30^\circ \times x = 60^\circ \) view and in wire frame mode. For both environments, the objects are scaled to fit within the graphics area of the dialog. However, tools are also available which allow you to change the viewing parameters (see next section). The Clear button can be used to clear the contents of the current environment of the Clipboard. This is useful when the Clipboard is using a large amount of memory. The dialog remains visible on the screen until the OK button is clicked or the return/enter key is pressed.

The form•Z Clipboard is a preview window. Like all the preview windows used elsewhere in form•Z it includes a set of tools that allow you to manipulate the preview image. These tools are discussed in section 2.1.6 of this Manual.
3.2.4 Selecting, deselecting, and clearing

This group of menu items complements the Pick and Delete tools (see sections 4.3 and 4.26 for modeling, 5.3 and 5.17 for drafting), found in both the modeling and drafting tool palettes. They affect both ghosted and unghosted modeling objects or drafting elements of the active window. Recall that ghosted is a state of an object/element where it is inactive and cannot be selected for an operation. Ghosted entities may be produced by the system or by the user through tools available in the tool palettes (see subsections 4.25.7 and 5.15.4).

**Select Previous**

This item unpicks any currently selected entities and repicks the previously selected entities. This command is useful when a (possibly large) number of selections are accidentally deselected.

**Select All UnGhosted**

This item picks and highlights all the modeling objects or drafting elements (depending on the active window) that are currently unghosted (active).

**Select All Ghosted**

This command selects and highlights all the ghosted (inactive) modeling objects or drafting elements, depending on the type of window that is currently active.

**Select By...**

Similar to the Select All Ghosted and Select All UnGhosted commands, this item offers methods for selecting groups of modeling objects or drafting elements that share common characteristics. The criteria by which entities are picked are set in the Selection Criteria: Modeling (Figure 3.2.4.1) or the Selection Criteria: Drafting (Figure 3.2.4.4) dialogs, which are invoked from the Select By... command. The dialog invoked depends on the active window (modeling or drafting).

The Selection Criteria dialog, in both modeling and drafting, offers a list of selection sets, on the left side of the dialog, as shown in Figure 3.2.4.1. In this list, one or more selection sets are displayed, which means that multiple selection sets can be defined. These sets are saved with the project. A selection set is a group of selection criteria. One of the selection sets is highlighted and is called the current set. The criteria for the current set are displayed in the right portion of the dialog. Clicking on a different item in the selection set list, makes the picked item the new current set and the criteria on the right side of the dialog are updated. Changes made to the criteria are applied to the current set. Under the selection set list are three buttons: New, Copy, and Delete.

**New**: Clicking on this button creates a new selection set, which is initially empty and becomes the current set. Selecting criteria while the new set is the current set, applies them to the new set.

**Copy**: When this button is pressed, a new selection set is created and the criteria of the current set are copied to the new set. The new set becomes the current set.

**Delete**: When this button is pressed, the current set is deleted and the next set in the list becomes the current set. When only one selection set remains in the list, the Delete button is disabled.

When pressing OK to exit the Selection Criteria dialog, the criteria defined in the current selection set are used to perform the object/element selection.
Once the selection criteria are set in the appropriate dialog, all of the objects/elements in the active project which match the selection criteria will be picked. There is a set of selection criteria for each open form-Z project, therefore changes made to the criteria dialogs for one project will not affect the selection criteria in other open projects. The selection criteria are organized into categories. Each category is in a boxed area of the dialog, with a check box at the top. A category is used only when its check box is selected.

Within each category, any number of selection attributes can be checked to determine which attributes of an object or element must match so that the object or element will be picked. If multiple attributes are selected in a category, the attributes of an object or element can match any one of the selected attributes. If multiple categories are selected, an object or element must match at least one attribute in each of the selected categories. For example, if the category Topology Types is selected with the attributes Solids and Surfaces, and the category Layer is selected with the attributes “layer 1” and “layer 3,” all objects which are solids or surfaces, and which are located on layer 1 or layer 3 are picked. However, the objects can have any color or smoothness, since those categories are not checked. By default, all selection criteria categories are deselected, which selects all objects.

Select By... in modeling

When clicking on the Select By... item while a modeling window is active, the Selection Criteria: Modeling dialog, shown in Figure 3.2.4.1 is invoked. As shown, it contains four tabs, each containing distinct categories of selection criteria.

Geometry tab

Topology Types: When this category is selected, an object must match at least one of the selected topology types (see subsection 4.0.1 in the form-Z User’s Manual) in this category in order to be picked.

Object Types: When this category is selected, an object must match at least one of the selected object types in this category in order to be picked. When the Symbols... item (last item in the list) is checked while the option key (Macintosh) or ctrl+shift (Windows) is pressed, the Symbol Instance Selection Criteria dialog is invoked. When the Text... item is checked while the option key (Macintosh) or ctrl+shift (Windows) is pressed, the Text Selection Criteria dialog is invoked. These dialogs are discussed at the end of this section.

Model Types: When this category is selected, an object must match the selected type in order to be picked. There are two types: Facetted and Smooth.

# Of Faces: When this category is selected, the number of faces of an object must be greater than or equal to the value shown in the Min field, and less than or equal to the value shown in the Max field, in order to be picked.

Attributes tab

Surface Style: When this category is selected, the color attribute of an object must match at least one of the selected colors in order to be picked.

Menus • The Edit menu
Figure 3.2.4.1: The Selection Criteria: Modeling dialog with the (a) Geometry, (b) Attributes, (c) Rendering, and (d) Script tab selected.
Layer: When this category is selected, the layer attribute of an object must match at least one of the selected layers in order to be picked. Layers which are not visible, or are ghosted or locked are not selectable, and therefore are not shown in this category.

Name: When this category is selected, the names of objects and/or groups are used to select them. The topological level (object or group) to be used is selected from the pop up menu next to the option’s check box, as shown in Figure 3.2.4.2. The type of string matching to be applied is selected from the Which pop up menu.

Which: One of the following six methods of string matching can be selected from this pop up menu, as shown in Figure 3.2.4.3. The string to be matched is typed in the text field next it.

Contains: When this option is on, objects and/or groups whose name contains the string entered in the text field are selected. The matching string can be anywhere in the object or group name.

Starts With: With this option, objects and/or groups whose name starts with the string in the text field are selected.

Ends With: With this option, objects and/or groups whose name ends with the string in the text field are selected.

Is: With this option, objects and/or groups are selected only if their names match the string entered in the text field exactly.

Is Not: With this option, objects and/or groups whose name does not precisely match the string in the text field are selected. This option is exactly the opposite of the previous one.

Does Not Contain: This option selects the objects and/or groups whose name does not contain the string entered in the text field. This is exactly the opposite of Contains.

Ignore Case: When this option is selected, the case of the strings in the object/group names and in the text field are ignored.

Rendering tab

This tab contains the Smooth Shading, Shadows, Rendering Backdrop, Render As Wire Frame, and Render As Surface criteria.

Smooth Shading: When this category is selected, the smooth shading attribute of an object must match the selected option (see subsection 4.25.5 in the form•Z User’s Manual) in this category in order to be picked.

Shadows: When this category is on, objects are selected according to their shadow Casting and/or Receiving attributes (see subsection 4.25.5 in the form•Z User’s Manual).

Rendering Backdrop: When this category is selected, an object must match the selected option in order to be picked. There are two options: Is Not Backdrop and Is Backdrop. Turning on one of these options selects all the objects that either do not have or have this attribute.
**Render As Wire Frame**: When this category is selected, an object must match the selected option in order to be selected. There are two options: **Yes** and **No**. Picking one of these options selects all the objects that are rendered as wire frame (**Yes**) or not as wire frame (**No**).

**Render As Surface**: This category works in the same manner as **Render As Wire Frame**, except that the render as surface attribute is considered.

**Script** tab

This tab contains the option of writing a **Script** for the selection of entities. That is, instead of presenting a particular property of an object by which it would be selected, the **Script** category allows the user to define a function, which determines whether a given object should be selected or not. The statements in the function must be written in the FSL script language, which is described in further detail in the **form•Z** SDK development manual. The object to be checked is passed as an argument into the function. The function returns a Boolean value. If set to FALSE, the object will not be selected. If set to TRUE, it will. The statements in the function should set the return value based on the user’s desired selection criteria.

For example, the statements could identify only those objects that have a volume less than a certain quantity. To retrieve information about an object using the FSL language, any number of **form•Z** api functions can be called. Which **form•Z** api functions are available in the script language can be found in the online **form•Z** SDK documentation, which includes details about how each function works. The statements for the function can be typed in the large text edit field in the **Script** tab. Below the field is the **Check Syntax** button. When pressed, the typed statements are checked for correct syntax. Any errors are identified and highlighted. Only when the **Check Syntax** button produces no error, will the **Script** selection category produce the expected result.

The statements of the function can also be debugged through the same environment available in the **Extensions** menu. When the **Debug Script** check box is selected and the **Selection Criteria : Modeling** dialog is closed, the script statements are shown in the debug dialog when the actual object is selected. This allows the user to step through the script code one statement at a time and determines whether the typed statements have the desired effect. Debugging scripts is explained in further detail in section 3.8.2 of the **form•Z** SDK documentation.

A few simple example scripts are shown below. Note, that only the statements for the body of the function need to be written. The function header and return statement are already defined and are shown as non editable text in the dialog. The examples below only outline the function statements. The function also initializes the return value (rv) to FALSE. Therefore, the statements only need to set rv to TRUE, if the object needs to be selected, but not to FALSE if it doesn’t.
To select objects with a volume between 100 and 1000 cubic inches:

double volume;

fz_objt_alys_get_volume(windex, objt, volume);
if (volume >= 100 && volume <= 1000) rv = TRUE;

To select objects which have a texture map control attribute and the mapping type of the object level texture group is cylindrical:

fzrt_boolean has_tctl;
fz_map_space_type_enum map_type;

fz_objt_attr_objt_has_tctl(windex, objt, has_tctl);
if (has_tctl == TRUE)
{
    fz_objt_attr_get_objt_tctl_parm(windex, objt, -1,
        FZ_ATTR_TCTL_PARM_MAP_TYPE, map_type);
    if (map_type == FZ_MAP_SPACE_CYLINDER) rv = TRUE;
}

To select objects, which are at least 24 inches high, measured along their local z axis:

fz_xyz_td bvol[8];
double dist;

fz_objt_alys_get_objt_bvol(windex, objt, bvol);
dist = FZ_MATH_3D_DIST_PT_PT(bvol[0], bvol[4]);
if (dist > 24) rv = TRUE;

To select all objects which use the Brick color shader in their surface style:

long rmtl_tag;
fz_rmtl_ptr rmtl_ptr;
fz_shdr_ptr shader;
fzrt_UUID_td type;

fz_objt_attr_get_objt_rmtl(windex, objt, FALSE, rmtl_tag);
fz_rmtl_tag_to_ptr(windex, rmtl_tag, rmtl_ptr);
fz_rmtl_get_parm(windex, rmtl_ptr, FZ_RMTL_PARM_COL_SHADER, shader);
fz_shdr_ptr_get_shdr_type(windex, shader, type);

if (type == FZ_SHDR_COLOR_BRICK_UUID) rv = TRUE;
General options

The following options apply in addition to the context of all four tabs.

**Apply To Unghosted:** When this option is checked, the selected constraints are applied to all unghosted objects in the active project.

**Apply To Ghosted:** When this option is checked, the selected constraints are applied to all ghosted objects in the active project.

**Add To Pick:** When this option is selected, the objects defined by the parameters specified in the Selection Criteria: Modeling dialog are added to the already selected objects.

**Remove From Pick:** This option actually represents a deselection method. When selected, the objects defined by the parameters specified in the Selection Criteria: Modeling dialog are removed from the currently selected objects.

**All:** When this button is selected, all categories and all attributes in each category are selected.

**None:** When this button is selected, all categories and all attributes in each category of both tabs are deselected. This will select all objects since no criteria are specified.
Select By... in drafting

When clicking on the Select By... item while a drafting window is active, the Selection Criteria: Drafting dialog (Figure 3.2.4.4) is invoked. As in the modeling equivalent, the Selection Criteria: Drafting dialog has a list of selection sets on its left. It works the same as in the modeling dialog. The drafting dialog contains two tabs, which correspond to the first two tabs of the modeling version.

Geometry tab

Element Types: When this category is selected, an element must match at least one of the selected element types in this category to be picked. When the Symbols... item is selected while the option key (Macintosh) or ctrl+shift (Windows) is pressed, the Symbol Instance Selection Criteria dialog is invoked. When the Text... item is selected while the option key (Macintosh) or ctrl+shift (Windows) is pressed, the Text Instance Selection Criteria dialog is invoked. These dialogs are discussed at the end of this section. The element types are discussed in section 5.0 in the Drafting Volume of the form•Z User’s Manual.

Attributes tab

Color: When this category is selected, the color attribute of an element must match at least one of the selected colors in order to be picked.

Layer: When this category is selected, the layer attribute of an element must match at least one of the selected layers in order to be picked. Layers which are not visible, or are ghosted or locked are not selectable, and therefore are not shown in this category.

Line Weight: When this category is selected, the line weight attribute of an element must match at least one of the selected line weights in order to be picked.

Line Style: When this category is selected, the line style attribute of an element must match at least one of the selected line styles in order to be picked.

Figure 3.2.4.4: The Selection Criteria: Drafting dialog with the (a) Geometry and (b) Attributes tab selected.
General options

**Apply To Unghosted**: When this option is checked, the selected constraints are applied to all unghosted elements in the current project.

**Apply To Ghosted**: When this option is checked, the selected constraints are applied to all ghosted elements in the current project.

**Add To Pick**: When this option is selected, the elements defined by the parameters specified in the **Selection Criteria: Drafting** dialog are added to the already selected elements.

**Remove From Pick**: With this option, the elements defined by the parameters specified in the **Selection Criteria: Drafting** dialog are removed from the currently selected elements.

**All**: When this button is selected, all categories and all attributes in each category are selected.

**None**: When this button is on, all categories and all attributes in each category are deselected.
Select By... for symbols

Selecting Symbols... while pressing the option key (Macintosh) or ctrl+shift (Windows) from either the Object Types category of the Selection Criteria: Modeling or Element Types category of the Selection Criteria: Drafting dialog, invokes the Symbol Instance Selection Criteria dialog (Figure 3.2.4.5), which is used to select symbol instances based on symbol attributes. The rules of the categories and attributes used for the drafting and modeling criteria also apply to the symbols criteria.

Library: When this category is selected, the library of the definition referenced by a symbol instance must be among the selected libraries in this category in order for the symbol instance to be picked.

Definition: When this category is selected, the definition referenced by a symbol instance must be among the selected definitions in this category in order to pick the symbol instance. The list of definitions is determined by the selected libraries in the Library category. That is, only those definitions which are part of the selected libraries are shown in the list. If no libraries are selected, or the Library category is not checked, the definitions of all libraries are shown.

Figure 3.2.4.5: The Symbol Instance Selection Criteria dialog.

Color Display: When this category is selected, the color display attribute of a symbol instance (see subsection 4.20.5.5 in the form-Z User’s Manual) must match at least one of the selected color display attributes in order for the symbol to be picked.

Layer Display: When this category is selected, the layer display attribute of a symbol instance (see subsection 4.20.5.5 in the form-Z User’s Manual) must match a selected layer display attribute in order for the symbol to be picked.

Detail: When this category is selected, the detail level of a symbol instance must match at least one of the selected detail levels in order for the symbol to be picked.

All: When this button is selected, all categories and all attributes in each category are selected.

None: When this button is selected, all categories and all attributes in each category are deselected.
Select By... for model text

Selecting **Text...** while pressing the option key (Macintosh) or **ctrl+shift** (Windows) from either the **Object Types** category of the **Selection Criteria** dialog, invokes the **3D Text Selection Criteria** dialog (Figure 3.2.4.6), which is used to select text instances based on text attributes. The rules of the categories and attributes used for the modeling criteria also apply to the text criteria. Text options are discussed in section 4.19 of the **form•Z** User’s Manual.

![3D Text Selection Criteria dialog](Figure 3.2.4.6: The 3D Text Selection Criteria dialog.)

Text objects can be selected by any of the text attributes. Numeric attributes can be selected by a range. There are two text edit fields for each numeric value. The left specifies the lower limit of the range and the right specifies the upper limit of the range. To define an exact match, enter the same value in both text fields.

Some text options are represented in the **Text Place** and **Text Edit** dialogs as check boxes or radio buttons. This dialog represents the values of these options as menus containing **On/Off** options. An **On** selection means the option in the **Text Place** or **Text Edit** dialog was selected, **Off** means it was not selected. The check box in this dialog simply means that the option is to be used as a search criteria.

*Menus* • *The Edit menu*
**Font:** The font to search for. The content of the **Font** menu depends on the fonts you have loaded on your machine.

**Size:** The font size to search for.

**Style:** The font style to search for. The content of this menu depends on the selected font. This option is only available if a font is selected.

**Which:** One of the following methods of string matching can be selected from this pop up menu, as shown in Figure 3.2.4.3. The string to be matched is typed in the text field below it. This option is only available when the **Pick** option is selected.

- **Contains:** When this option is selected, the search string anywhere within an object’s text is considered a match.
- **Starts With:** When this option is selected, an object’s text must begin with the search string to be considered a match.
- **Ends With:** When this option is selected, an object’s text must end with the search string to be considered a match.
- **Is:** When this option is selected, the entire text of an object must match the search string to be considered a match.
- **Is Not:** When this option is selected, any object’s text which does not entirely match the search string is considered a match.
- **Does Not Contain:** When this option is selected, any object’s text which does not contain the search string is considered a match.

**Search String:** The string to search for.

**Ignore Case:** When this option is selected, the case of the strings does not need match the case of the search string in order to be picked. If this option is not selected, the case must match.

**Match Whole Word Only:** When this option is selected, the search string must not be part of a larger word in the object’s text to be considered a match.
The **Text Type** tab

**Text Type:** When this option is selected, the text object’s text type must match the selected text type to be considered a match. This option is made up of two menus. The first specifies the whether the object was created as plain text or as object text. The second menu specifies the placement type of the text.

![Text Type tab](image)

*Figure 3.2.4.7*: The **Text Type** tab in the 3D Text Selection Criteria dialog.

The first menu contains the following selections:

- **Plain Text:** A text object created as plain text is considered a match.
- **Text As Object:** A text objet created as object text is considered a match.

The second menu contains the following selections:

- **At Point:** A text object created with the **At Point** placement type is considered a match.
- **Between Points:** A text object created with the **Between Points** placement type is considered a match.
- **On Line:** A text object created with the **On Line** placement type is considered a match.
- **Between Parallel Lines:** A text object created with the **Between Parallel Lines** placement type is considered a match.
- **Between Lines:** A text object created with the **Between Lines** placement type is considered a match.

**Preserve Height & Width:** When this option is selected, a text object created with a matching **Preserve Height & Width** option is considered a match.

**Scale Width Only:** When this option is selected, a text object created with a matching **Scale Width Only** option is considered a match.

**Scale Height & Width:** When this option is selected, a text object created with a matching **Scale Height & Width** option is considered a match.

**Adjusted To Line/Curve:** When this option is selected, a text object created with a matching **Adjusted To Line/Curve** option is considered a match.
The Text Options tab

![Image of Text Options tab]

*Figure 3.2.4.8:* The Text Options tab in the 3D Text Selection Criteria dialog.

**Topology Type:** When this option is selected, a text object with the same Topology Type as selected here is considered a match.

**Height Base:** When this option is selected, a text object with the same Height Base as selected here is considered a match.

  **Character:** When this option is selected, a text object with the same Height Base Character as selected here is considered a match.

**Height:** When this option is selected, a text object with a Height value in the range specified here is considered a match. Additionally, the height justification can be selected as a search criteria and its value specified.

**Depth:** When this option is selected, a text object with a Depth value in the range specified here is considered a match. Additionally, the depth justification can be selected as a search criteria and its value specified.

**Width:** When this option is selected, a text object with a Width value in the range specified here is considered a match.

**Leading:** When this option is selected, a text object with a Leading value in the range specified here is considered a match.

**Standing (Perp To Plane):** When this option is selected, a text object created with a matching Standing (Perp To Plane) option is considered a match.

**String Angle:** When this option is selected, a text object with a String Angle value in the range specified here is considered a match.

**Character Angle:** When this option is selected, a text object with a Character Angle value in the range specified here is considered a match.

**Incline Angle:** When this option is selected, a text object with an Incline Angle value in the range specified here is considered a match.
The **Facetted Rounding Options** tab

![Image of Facetted Rounding Options tab]

*Figure 3.2.4.9:* The **Facetted Rounding Options** tab in the **3D Text Selection Criteria** dialog.

**Normal:** When this option is selected and it’s menu set to **On**, a text object that has it’s **Facetted Rounding** type set to **Normal** is considered a match. If this option is selected and the menu is set to **Off**, a text object that has a **Facetted Rounding** type that is not **Normal** is considered a match.

**Rounded:** When this option is selected and it’s menu set to **On**, a text object that has it’s **Facetted Rounding** type set to **Rounded** is considered a match. If this option is selected and the menu is set to **Off**, a text object that has a **Facetted Rounding** type that is not **Rounded** is considered a match.

**Radius:** When this option is selected, a text object with **Facetted Rounding** type of **Rounded** and a **Radius** value in the range specified here is considered a match.

**# Of Edges:** When this option is selected, a text object with **Facetted Rounding** type of **Rounded** and a **# Of Edges** value in the range specified here is considered a match.

**Beveled:** When this option is selected and it’s menu set to **On**, a text object that has it’s **Facetted Rounding** type set to **Beveled** is considered a match. If this option is selected and the menu is set to **Off**, a text object that has a **Facetted Rounding** type that is not **Beveled** is considered a match.

**Offset:** When this option is selected, a text object with **Facetted Rounding** type of **Beveled** and a **Offset** value in the range specified here is considered a match.

**Adjust Radius/Offset:** When this option is selected, a text object with a matching **Adjust Radius/Offset** option is considered a match.
The Smooth Rounding Options tab

<table>
<thead>
<tr>
<th>Front</th>
<th>Back</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge Type</td>
<td>Edge Type</td>
</tr>
<tr>
<td>Radius</td>
<td>Radius</td>
</tr>
<tr>
<td>Major Radius</td>
<td>Major Radius</td>
</tr>
<tr>
<td>Minor Radius</td>
<td>Minor Radius</td>
</tr>
<tr>
<td>Rotation Angle</td>
<td>Rotation Angle</td>
</tr>
<tr>
<td>Left Distance</td>
<td>Left Distance</td>
</tr>
<tr>
<td>Right Distance</td>
<td>Right Distance</td>
</tr>
<tr>
<td>Bulge</td>
<td>Bulge</td>
</tr>
</tbody>
</table>

![Figure 3.2.4.10: The Smooth Rounding Options tab in the 3D Text Selection Criteria dialog.](image)

**Edge Type:** When this option is selected, a text object with the same Smooth Rounding Edge Type as selected here is considered a match.

**Radius:** When this option is selected, a text object with a Smooth Rounding type of Circular and a Radius value in the range specified here is considered a match.

**Major Radius:** When this option is selected, a text object with a Smooth Rounding type of Elliptical and a Major Radius value in the range specified here is considered a match.

**Minor Radius:** When this option is selected, a text object with a Smooth Rounding type of Elliptical and a Minor Radius value in the range specified here is considered a match.

**Rotation Angle:** When this option is selected, a text object with a Smooth Rounding type of Elliptical and a Rotation Angle value in the range specified here is considered a match.

**Left Distance:** When this option is selected, a text object with a Smooth Rounding type of Bevel and a Left Distance value in the range specified here is considered a match.

**Right Distance:** When this option is selected, a text object with a Smooth Rounding type of Bevel and a Right Distance value in the range specified here is considered a match.

**Bulge:** When this option is selected, a text object with Smooth Rounding type of Bevel and a Bulge value in the range specified here is considered a match.
Select By... for draft text

Selecting Text... while pressing the option key (Macintosh) or ctrl+shift (Windows) from either the **Element Types** category of the Selection Criteria: Drafting dialog, invokes the Draft Text Selection Criteria dialog (Figure 3.2.4.11), which is used to select text instances based on text attributes. The rules of the categories and attributes used for the drafting criteria also apply to the text criteria. Text options are discussed in section 5.7 of the form•Z User’s Manual.

Text elements can be selected by any of the text attributes. Numeric attributes can be selected by a range. There are two text edit fields for each numeric value. The left specifies the lower limit of the range and the right specifies the upper limit of the range. To define an exact match, enter the same value in both text fields.

Some text options are represented in the Text Place and Text Edit dialogs as check boxes or radio buttons. This dialog represents the values of these options as menus containing **On/Off** options. An On selection means the option in the Text Place or Text Edit dialog was selected, Off means it was not selected. The check box in this dialog simply means that the option is to be used as a search criteria.

![Draft Text Selection Criteria dialog](Figure 3.2.4.11: The Draft Text Selection Criteria dialog.)
**Font**: This item determines the font to search for. The content of the **Font** menu depends on the fonts you have loaded on your machine.

**Size**: This item determines the font size to search for.

**Style**: This item determines the font style to search for. The content of this menu depends on the selected font. This option is only available if a font is selected.

**Which**: One of the following methods of string matching can be selected from this pop up menu. The string to be matched is typed in the text field below it. This option is only available when the **Pick** option is selected.

- **Contains**: When this option is selected, the search string anywhere within an element’s text is considered a match.
- **Starts With**: When this option is selected, an element’s text must begin with the search string to be considered a match.
- **Ends With**: When this option is selected, an element’s text must end with the search string to be considered a match.
- **Is**: When this option is selected, the entire text of an element must match the search string to be considered a match.
- **Is Not**: When this option is selected, any element’s text which does not entirely match the search string is considered a match.
- **Does Not Contain**: When this option is selected, any element’s text which does not contain the search string is considered a match.

**Search String**: The string to search for.

**Ignore Case**: When this option is selected, the case of the strings does not need match the case of the search string in order to be picked. If this option is not selected, the case must match.

**Match Whole Word Only**: When this option is selected, the search string must not be part of a larger word in the object’s text to be considered a match.
The **Text Type** tab

![Image 1](image1.png)

*Figure 3.2.4.12: The Text Type tab in the Draft Text Selection Criteria dialog.*

**Text Type:** When this option is selected, the text object’s text type must match the selected text type to be considered a match. This specifies the placement type of the text. This menu contains the following selections:

- **At Point:** A text element created with the At Point placement type is considered a match.
- **Between Points:** A text element created with the Between Points placement type is considered a match.

The **Text Options** tab

![Image 2](image2.png)

*Figure 3.2.4.13: The Text Options tab in the Draft Text Selection Criteria dialog.*

**Height Base:** When this option is selected, a text element with the same **Height Base** as selected here is considered a match.

- **Character:** When this option is selected, a text element with the same **Height Base Character** as selected here is considered a match.

**Height:** When this option is selected, a text element with a **Height** value in the range specified here is considered a match. Additionally, the height justification can be selected as a search criteria and its value specified.

**Width:** When this option is selected, a text element with a **Width** value in the range specified here is considered a match.

**Leading:** When this option is selected, a text element with a **Leading** value in the range specified here is considered a match.

**String Angle:** When this option is selected, a text element with a **String Angle** value in the range specified here is considered a match.
**Deselect**

Clicking on this item deselects all the currently selected entities.

**Clear**

This item deletes all the unghosted (active) modeling objects or drafting elements from the current project, depending on the active window. Unlike the **Cut** command, the cleared entities are not placed in the Clipboard. When executed, the system prompts the user to confirm their intentions before deleting the objects or elements.

**Clear All Ghosted**

This item functions identically as the **Clear** command, but affects the currently ghosted objects or drafting elements only. The user is prompted to confirm before clearing the entities.
3.2.5 Key shortcuts

Key shortcuts, which are also called key commands, hot keys, or simply shortcuts, are a combination of keystrokes and/or mouse clicks that perform actions, which would otherwise be initiated from a menu item, a tool icon, or a dialog. form•Z offers key shortcuts to most of the actions that the program executes (which is more than 600). You are also able to customize the key shortcut assignments for all the actions, can save any number of key assignment sets, and can switch from one set to another whenever it facilitates the modeling task at hand.

form•Z is shipped with a default set of key shortcuts. This set can be edited, new commands can be added to it, and others deleted. Different sets of shortcuts can be saved and loaded at any time. The shortcut files are saved with a “.sct” extension. The set of defined shortcuts is saved in the form•Z defaults file, the content of which is shown in the table of Figure 3.2.5.1.

An action in form•Z is the execution of an operation from a tool or menu item, selecting the tool icons and the menu items themselves, invoking dialogs, selecting options from dialogs, and even operations that have no other counterpart but are only executed through a keystroke.

A valid shortcut in form•Z is a single keystroke, one or more modifiers plus a key or a mouse-click. The available modifier keys are:

- Macintosh: command, control, shift, and option;
- Windows: control, alt, and shift.

Shortcuts assigned to menu items are displayed on the right side of the respective menu item. Given that, in form•Z, up to five distinct key shortcuts may be assigned to the same action, if more than one shortcut has been assigned to a menu item, only the first one is displayed in the menu.

WARNING: Recall that the keyboard is also used to enter input through the Prompts palette, which may interfere with certain key assignments. For example, the space bar, which some users may want to assign to the pan operation, also enters a space in the Prompts palettes, which may take precedence. Following are the conditions you need to keep in mind.

- If the Prompts palette is open and tracking is on, the Prompts palette takes precedence.
- If tracking is off, all keys will be recognized as key shortcuts, assuming there is an action assigned to them.
- If the Prompts palette is closed there is no interference with the key commands.
**form.Z default key shortcut assignments**

<table>
<thead>
<tr>
<th>Macintosh</th>
<th>Windows</th>
<th>Macintosh</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radiosity Options</td>
<td>F5</td>
<td>F5</td>
<td>Wire Frame Display</td>
</tr>
<tr>
<td>Initialize Radiosity</td>
<td>F6</td>
<td>F6</td>
<td>Cut</td>
</tr>
<tr>
<td>Generate Radiosity Sol.</td>
<td>F7</td>
<td>F7</td>
<td>Undo</td>
</tr>
<tr>
<td>Exit Radiosity</td>
<td>F8</td>
<td>F8</td>
<td>Zoom In Increment</td>
</tr>
<tr>
<td>Display Options</td>
<td>F9</td>
<td>F9</td>
<td>Reset View</td>
</tr>
<tr>
<td>Pick Options</td>
<td>F10</td>
<td>F10</td>
<td>Zoom Out Increment</td>
</tr>
<tr>
<td>Object Snap Options</td>
<td>F11</td>
<td>F11</td>
<td>Select Previous</td>
</tr>
<tr>
<td>Status of Objects</td>
<td>F12</td>
<td>F12</td>
<td>Decrease Scale, Y</td>
</tr>
<tr>
<td>Tool Options</td>
<td>F13</td>
<td>F13</td>
<td>Increase Scale, X</td>
</tr>
<tr>
<td>Window Options</td>
<td>F14</td>
<td>F14</td>
<td>Increase Scale, Y</td>
</tr>
<tr>
<td>Move Negative y</td>
<td>arrow, down</td>
<td>arrow, down</td>
<td>Deselect</td>
</tr>
<tr>
<td>Move Negative x</td>
<td>arrow, left</td>
<td>arrow, left</td>
<td>Previous View</td>
</tr>
<tr>
<td>Move Positive x</td>
<td>arrow, right</td>
<td>arrow, right</td>
<td>Project Info</td>
</tr>
<tr>
<td>Move Positive y</td>
<td>arrow, up</td>
<td>arrow, up</td>
<td>End Drawing</td>
</tr>
<tr>
<td>Close Drawing</td>
<td>C</td>
<td>C</td>
<td>Zoom in by Frame</td>
</tr>
<tr>
<td>Cancel</td>
<td>esc</td>
<td>esc</td>
<td>Decrease Scale, Z</td>
</tr>
<tr>
<td>Rotate View</td>
<td>cmd + ’</td>
<td>ctrl + ’</td>
<td>Increase Scale, Z</td>
</tr>
<tr>
<td>Preferences</td>
<td>cmd + ,</td>
<td>ctrl + ,</td>
<td>Rotate Negative x</td>
</tr>
<tr>
<td>[-ZX] Front</td>
<td>cmd + -</td>
<td>ctrl + -</td>
<td>Rotate Negative z</td>
</tr>
<tr>
<td>Help</td>
<td>cmd + /</td>
<td>ctrl + /</td>
<td>Rotate Positive z</td>
</tr>
<tr>
<td>[+ZX] Back</td>
<td>cmd + 0</td>
<td>ctrl + 0</td>
<td>Rotate Positive x</td>
</tr>
<tr>
<td>45 45 View</td>
<td>cmd + 2</td>
<td>ctrl + 2</td>
<td>Rotate Negative y</td>
</tr>
<tr>
<td>120 20 View</td>
<td>cmd + 3</td>
<td>ctrl + 3</td>
<td>Rotate Positive y</td>
</tr>
<tr>
<td>220 45 View</td>
<td>cmd + 4</td>
<td>ctrl + 4</td>
<td>Arbitrary Plane Active</td>
</tr>
<tr>
<td>60 30</td>
<td>cmd + 5</td>
<td>ctrl + 5</td>
<td>Outline</td>
</tr>
<tr>
<td>[+XY] Top</td>
<td>cmd + 6</td>
<td>ctrl + 6</td>
<td>Face</td>
</tr>
<tr>
<td>[+XZ] Bottom</td>
<td>cmd + 7</td>
<td>ctrl + 7</td>
<td>Grid Snap</td>
</tr>
<tr>
<td>[+YZ] Right Side</td>
<td>cmd + 8</td>
<td>ctrl + 8</td>
<td>Hole/Volume</td>
</tr>
<tr>
<td>[-YZ] Left Side</td>
<td>cmd + 9</td>
<td>ctrl + 9</td>
<td>Object</td>
</tr>
<tr>
<td>Set View</td>
<td>cmd + ;</td>
<td>ctrl + ;</td>
<td>Point</td>
</tr>
<tr>
<td>CANCEL</td>
<td>cmd + .</td>
<td>ctrl + .</td>
<td>Segment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Group</td>
</tr>
<tr>
<td>Interactive Shaded</td>
<td>cmd + J</td>
<td>ctrl + J</td>
<td>Perpendicular Switch</td>
</tr>
<tr>
<td>Select All Unghosted</td>
<td>cmd + A</td>
<td>ctrl + A</td>
<td>XY Plane Active</td>
</tr>
<tr>
<td>Redo</td>
<td>cmd + B</td>
<td>ctrl + B</td>
<td>YZ Plane Active</td>
</tr>
<tr>
<td>Copy</td>
<td>cmd + C</td>
<td>ctrl + C</td>
<td>ZX Plane Active</td>
</tr>
<tr>
<td>Duplicate</td>
<td>cmd + D</td>
<td>ctrl + D</td>
<td>Uniform Scale Decr.</td>
</tr>
<tr>
<td>Edit Cone Of Vision</td>
<td>cmd + E</td>
<td>ctrl + E</td>
<td>Uniform Scale Incr.</td>
</tr>
<tr>
<td>Fit All</td>
<td>cmd + F</td>
<td>ctrl + F</td>
<td>Ortho/Diagonal Snap</td>
</tr>
<tr>
<td>Grab Image</td>
<td>cmd + G</td>
<td>ctrl + G</td>
<td>Snap to Face Center</td>
</tr>
<tr>
<td>Hidden Line Display</td>
<td>cmd + H</td>
<td>ctrl + H</td>
<td>Angle/Slope Snap</td>
</tr>
<tr>
<td>Import</td>
<td>cmd + I</td>
<td>ctrl + I</td>
<td>Snap to End Point</td>
</tr>
<tr>
<td>RenderZone Display</td>
<td>cmd + J</td>
<td>ctrl + J</td>
<td>Snap to Face</td>
</tr>
<tr>
<td>Shaded Render Display</td>
<td>cmd + K</td>
<td>ctrl + K</td>
<td>Snap to Intersection</td>
</tr>
<tr>
<td>New Project [Model]</td>
<td>cmd + L</td>
<td>ctrl + L</td>
<td>Snap to Midpoint</td>
</tr>
<tr>
<td>Open</td>
<td>cmd + M</td>
<td>ctrl + M</td>
<td>No Object Snap</td>
</tr>
<tr>
<td>Print Page</td>
<td>cmd + N</td>
<td>ctrl + N</td>
<td>Ortho Snap</td>
</tr>
<tr>
<td>Quit/Exit</td>
<td>cmd + Q</td>
<td>ctrl + Q</td>
<td>Snap To Point</td>
</tr>
<tr>
<td>Surface Render Display</td>
<td>cmd + R</td>
<td>ctrl + R</td>
<td>Radial Snap</td>
</tr>
<tr>
<td>Save</td>
<td>cmd + S</td>
<td>ctrl + S</td>
<td>Snap to Segment</td>
</tr>
<tr>
<td>Quick Paint Display</td>
<td>cmd + T</td>
<td>ctrl + T</td>
<td>No Directional Snap</td>
</tr>
<tr>
<td>Underlay</td>
<td>cmd + U</td>
<td>ctrl + U</td>
<td>Move Negative z</td>
</tr>
<tr>
<td>Paste</td>
<td>cmd + V</td>
<td>ctrl + V</td>
<td>Move Positive z</td>
</tr>
<tr>
<td>Close Drawing thru Point</td>
<td>shift + C</td>
<td>shift + C</td>
<td></td>
</tr>
<tr>
<td>End Drawing thru Point</td>
<td>shift + E</td>
<td>shift + E</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3.2.5.1:** Default key shortcuts for the Macintosh and Windows.
Key Shortcuts...

When this item is selected from the **Edit** menu, it invokes the **Key Shortcuts Manager** dialog, which is shown in Figure 3.2.5.2 for both Macintosh and Windows.

The **Key Shortcuts Manager** dialog contains a number of button commands and four boxed areas. They will be referred to as **actions list** (left side), **action title** (upper right), **action description** (right, second from top), and **shortcut assignments** (middle right).

**Actions list**: This is a hierarchical list of the actions that **form-Z** executes and to which a shortcut can be assigned. Each level in the hierarchy groups actions that logically belong together.

The hierarchical nodes of the list can be closed or opened by clicking on the **arrows** (Macintosh), or **plus or minus** signs (Windows), displayed in front of them. Opening or closing a node while pressing the **shift** key, opens or closes the node and all its siblings and children. Opening or Closing a node while pressing the **option** key (Macintosh) or **ctrl+shift** (Windows) opens or closes the node and all of its children.

**Figure 3.2.5.2**: The **Key Shortcuts Manager** dialog for (a) Macintosh and (b) Windows.
To assign a shortcut or to apply another operation to an action, it needs to be selected. You can select an action by clicking on it with the mouse or by using the up and down arrow keys. A selected action is called the active action. Only those items in the list which are preceded by a square icon can be selected. These icons are color coded and represent the action, as follows:

- Blue is used for actions that select tool icons.
- Red is used for actions that invoke dialogs.
- Yellow is used for actions that execute operations.

**Action title:** This box displays an extended title of the active action. If the active action is a menu item, the title also includes the name of the menu where it is located. If the active action is a tool, its icon is also displayed next to the title. This box is updated every time the active action changes.

**Action description:** This box displays a summary description of the active action. It is updated every time the active action changes.

**Shortcut assignments:** This box displays all the key shortcuts, if any, that are currently assigned to the active action. This is updated every time the active action changes. Any shortcut can be edited, deleted, etc. using one of the following buttons. To select one, click on it.

**Add ...** This button is used to add a new shortcut to the active action. Clicking on it invokes the Add Shortcut dialog shown in Figure 3.2.5.3. Up to a maximum of five shortcuts can be assigned to a single action, but the same shortcut can only be assigned to a single action. If an attempt is made to assign to the active action a shortcut that is already assigned, a warning message is posted. In response to the message you may instruct the program to go ahead with the new assignment, which will result in the cancellation of the previous assignment. If the active action already has five shortcuts assigned to it, this button is dimmed.

The Add Shortcuts dialog contains a list of modifier keys to its left and two groups of options to its right. These are used to construct a key shortcut combination, which is displayed at the lower left corner of the dialog.

**Command, Control, Option, Shift** (Macintosh): **Control, Alt, Shift** (Windows): Selecting one or more of these modifier keys makes them part of the key shortcut.
**Key Stroke**: When this option is selected, an alphanumeric character is added to the shortcut. This character is selected by either typing it in the **Key** text field or by selecting it from a pop up menu (Macintosh) or a list (Windows), which is invoked from the **Browse** button.

**Mouse Click**: When this option is selected, a **Single**, **Double**, or **Triple** mouse click is added to the key shortcut. You also need to select the button with which the click is made. Only a single button is supported in MacOS. Up to five buttons are supported in OS X. On Windows the number of buttons is determined by the number of buttons in your mouse. Note that a shortcut can either have a key stroke or a mouse click, but not both.

**Edit...**: This button is used to edit an existing shortcut. Clicking on it invokes the **Edit Shortcut** dialog, shown in Figure 3.2.5.4 for both Macintosh and Windows. Except for its title, this dialog is identical with the **Add Shortcut** dialog, but works differently. It displays the shortcut that is already assigned to the active action, while the **Add Shortcut** dialog displays a default shortcut. As with the **Add...** command, if a shortcut that is edited becomes the same as one already assigned to another action, a warning message is posted. If the active action has not been assigned any shortcut yet, this button is dimmed.

**Delete**: Clicking on this button deletes the selected shortcut. If the active action does not have any shortcut assigned to it, this button is dimmed.

**Clear...**: Clicking on this button clears all the shortcuts defined in the current set, after posting a warning message. If the current set does not contain any shortcut assignments or after it has been cleared, this button remains dimmed.

**Reset...**: Clicking on this button resets the shortcut list to the state it was when the **Key Shortcuts Manager** dialog was last entered. A warning message is displayed before proceeding.

**Save...**: Clicking on this button invokes the standard Save File dialog from which the current set of shortcuts can be saved to a disk file, after specifying a name and location.

**Load...**: Clicking on this button invokes the standard File Open dialog, from which a shortcut file (.sct) can be opened.

**Load Defaults...**: Clicking on this button loads the system set of default shortcuts after posting a warning message.

*Figure 3.2.5.4: The Edit Shortcut dialog for (a) Macintosh and (b) Windows.*
Since version 5.5, for the arrow keys there are two sets of defaults, one supporting the nudge commands and the other the view navigation commands. Users can select to use one or the other, as follows: When the Load Defaults... button is selected in the Key Shortcuts Manager dialog, the Default Arrow Key Shortcut Selection dialog, shown in Figure 3.2.5.5, will be invoked. This allows for the selection of either the arrow key defaults for the nudge commands or the view navigation commands.

Use Arrow Keys For Nudge Commands (Pre-Version 5.5 Defaults): When this option is on, the nudge commands are assigned by default to the arrow keys.

Use Arrow Keys For View Commands (Current Defaults): When this option is on, the view navigation commands are assigned by default to the arrow keys.

Figure 3.2.5.5: The Default Arrow Key Shortcut Selection dialog.
When a key shortcuts file is loaded from the **Load...** button on the **Key Shortcuts Manager** dialog or at startup, the shortcuts file is scanned for conflicts with the new key shortcut defaults. If the file only contains the pre-version 5.5 defaults for the arrow keys and no other key shortcuts conflict with the view navigation tools, the **Default Arrow Key Shortcut Selection** dialog (Figure 3.2.5.5) will be invoked. Otherwise, the **Key Shortcut Conflicts** dialog (Figure 3.2.5.6) will be invoked. This dialog provides a selection mechanism for each conflicting key shortcut.

This dialog shows a list containing each conflicting key shortcut:

- **Shortcut** column shows the key shortcut.
- **From File** column shows the name of the command the shortcut is assigned to in the shortcuts file.
- **New Default** column shows the name of the view navigation command the shortcut defaults to.

Clicking in the **Use File** column will toggle a check mark in that column. The presence of a check mark indicates that the key shortcut from the shortcuts file is to be used. The default for this column is that none of the items are checked and that the view navigation shortcuts will override the shortcuts in the shortcuts file.

**Use All From File:** Clicking on this button places a check mark in the **Use File** column of the list indicating that the key shortcuts from the shortcuts file will be used for all conflicting shortcuts.

**Use None From File:** This removes all check marks in the **Use File** column of the list indicating that the default view navigation key shortcuts will be used for all conflicting shortcuts.

---

**Figure 3.2.5.6:**
The **Key Shortcut Conflicts** dialog.
**List Shortcuts...**: Clicking on this button invokes the **Shortcuts List** dialog (Figure 3.2.5.7) where the current set of defined shortcuts is displayed.

The shortcuts list is organized in four columns labeled Category, Key, Command, and Description. The first column displays the category of the action. These categories are the same top level categories displayed in the actions list of the **Key Shortcuts Manager** dialog. The Key column displays the shortcut that has been assigned to the action, which is displayed in the Command column. Finally, the last column displays a brief description of the action.

The list can be sorted by the entries in any one of the four columns, by clicking on the title of that column. Clicking while holding the **option** key (Macintosh) or **ctrl+shift** (Windows) produces inverse sorting. Initially, the list is sorted by the key. To sort by modifiers select **Sort By Modifier** (see below).

The width of the columns can be changed by dragging the dividing lines with the mouse. Double clicking invokes the previously described **Edit Shortcut** dialog. Pressing **delete** (Macintosh) or **backspace** (Windows) deletes the selected shortcut after a warning.

![Shortcuts List dialog](image)

**Figure 3.2.5.7**: The **Shortcuts List** dialog (Macintosh).
**Sort By Modifier:** When this option is selected, the key column is sorted by the modifiers as a primary sort, and by the key as a secondary sort.

**Print...**: Clicking on this button invokes the standard Print dialog from where the list of shortcuts can be printed.

**Keyboard...**: Clicking on this button invokes the **Keyboard Layout** dialog for finding out what keys execute what actions. This is the same dialog used by the Help environment (see section 1.8 in the form•Z Manual).

**Prompts palette shortcuts**

There are four shortcuts that deserve special mention. They are used in the Prompts palette for terminating line and arc drawing by closing or ending the drawn shape. These are equivalent to the double (end) and triple (close) click actions to terminate drawing. Their default key assignments are as follow, but these can freely be changed by a user:

- **Close** (c): Terminates drawing and connects the last point of the shape to the first point.
- **Close Through Point** (shift+c): Adds a point to the shape at the current cursor location and closes the shape by connecting it to the first point (equivalent to a triple-click).
- **End** (e): Terminates drawing leaving an open shape with the end at the last point.
- **End Through Point** (shift+e): Adds a point to the shape at the current cursor location making it the end of the shape (equivalent to a double-click).

**Scroll wheel functions**

Some special functionality is attached to the scroll wheel, available whenever a mouse with a scroll wheel is used.

**Scrolling the contents of palettes and dialogs**

To scroll the contents of palettes that contain scroll bars, place the cursor over the palette and use the scroll wheel to scroll up and down.

To scroll the content of dialogs, place the cursor over the area that is scrollable (indicated by the presence of a scroll bar) and use the scroll wheel to scroll up and down.

**Zooming in and out**

To zoom a view displayed in a graphics window, place the cursor in the window and roll the scroll wheel. The image zooms in about the cursor position when the wheel is rolled up and it zooms out when it is rolled down. This functionality is also available in dialogs with graphic previews.
3.2.6 Direct view navigation

A series of special key and mouse driven view navigation commands are available. These are extensions of the navigation tools, however, they are much easier to use as they are tied to keys and/or mouse buttons and actions, which facilitates view navigation while working, without the need for icon selection and extra clicks.

The actions are listed below. Note that the items that contain “interactive” in their name will only work when assigned to a shortcut that uses a mouse button. The three variations of the Fit All operation can be assigned to a mouse button or a key. The remainder will only work well when used as a pure key shortcut, such as the arrow or other keys. Defaults have been assigned to a number of the shortcut commands and are shown in parentheses on the following list:
Interactive

Hand Interactive (*single click* of *middle* button): This works as the Hand window tool. That is, the image on the screen moves as the mouse is moved, as shown in Figure 3.2.6.1(a). Note that, in a zoomed perspective, the movement is bounded by the perspective focal length.

Pan Interactive (*shift* + *single click* of *middle* button): This works as the Hand window tool, except that the line of sight is always moved, as shown in Figure 3.2.6.1(b). This difference is only realized for perspective zoomed images.

Set View Interactive about Center of Interest (*cmd* + *single click* of *right* button [Mac] or *ctrl* + *single click* of *right* button [Windows]): This works as the Set View window tool with the Center Of Interest option on, as shown in Figure 3.2.6.1(c).

Set View Interactive about Center of Scene (*opt* + *single click* of *right* button [Mac] or *ctrl* + *shift* + *single click* of *right* button [Windows]): This works as the Set View window tool with the Center Of Scene option on.

Set View Interactive about Point on Reference Plane (*cmd* + *opt* + *single click* of *right* button [Mac] or *ctrl* + *shift* + *alt* + *single click* of *right* button [Windows]): This works as the Set View window tool with the Point On Reference Plane option on.

Spin Interactive (*cmd* + *opt* + *single click* of *left* button [Mac] or *ctrl* + *shift* + *alt* + *single click* of *left* button [Windows]): This interactively changes the view spin (head tilt), as shown in Figure 3.2.6.1(d).

Depth Interactive (*cmd* + *single click* of *left* button [Mac] or *ctrl* + *single click* of *left* button [Windows]): This interactively moves in/out along the line of sight, as the cursor is moved up/down, as shown in Figure 3.2.6.1(e).

Focal Interactive (*opt* + *single click* of *left* button [Mac] or *ctrl* + *shift* + *single click* of *left* button [Windows]): This interactively changes the perspective focal length, as the cursor is moved up/down, as shown in Figure 3.2.6.1(f).

Figure 3.2.6.1: Interactive view navigation key commands for: (a) Hand, (b) Pan, (c) Set View, (d) Spin, (e) Depth, and (f) Focal.
The Edit menu

Keyed

Fit All (opt + double click of middle button [Mac] or ctrl + shift + double click of middle button [Windows]): This is the same as the Fit All window tool with Include Lights and Include Views selected. In addition, it adjusts the center of interest (COI) to the center of what is fit to.

Fit All Objects (double click of middle button): This is the same as the Fit All window tool with Include Lights and Include Views OFF. In addition, it adjusts the COI to the center of what is fit to.

Fit All Selected (cmd + double click of middle button [Mac] or ctrl + double click of middle button [Windows]): This is the same as the Fit All window tool with Include Selected enabled. In addition, it adjusts the COI to the center of what is fit to.

Move View Right (right arrow), Move View Left (left arrow), Move View In (up arrow), Move View Out (down arrow), Move View Up (shift + up arrow), Move View Down (shift + down arrow): These move the EYE to the respective direction, as shown in Figure 3.2.6.2(a).

Look Right (cmd + right arrow [Mac]; ctrl + right arrow [Windows]), Look Left (cmd + left arrow [Mac]; ctrl + left arrow [Windows]), Look Up (cmd + up arrow [Mac]; ctrl + up arrow [Windows]), Look Down (cmd + down arrow [Mac]; ctrl + down arrow [Windows]): These rotate the COI about the EYE in the respective direction, as shown in Figure 3.2.6.2(b).

Spin Clockwise (opt + right arrow [Mac]; ctrl + shift + right arrow [Windows]), Spin Counterclockwise (opt + left arrow [Mac]; ctrl + shift + left arrow [Windows]): These change the view spin (head tilt) in the specified direction, as shown in Figure 3.2.6.2(c).

Rotate View Right (ctrl + right arrow [Mac]; ctrl + alt + right arrow [Windows]), Rotate View Left (ctrl + left arrow [Mac]; ctrl + alt + left arrow [Windows]), Rotate View Up (ctrl + up arrow [Mac]; ctrl + alt + up arrow [Windows]), Rotate View Down (ctrl + down arrow [Mac]; ctrl + alt + down arrow [Windows]): These rotate the EYE about the COI in the specified direction.

Figure 3.2.6.2: View navigation key commands:
(a) Move View Left/Right,
(b) Move View Up/Down,
(c) Move View In/Out,
(d) Look Up/Down,
(e) Rotate View Left/Right,
and (f) Rotate View Up/Down.
3.2.7 Saving and recalling preferences

With the next to the last item in the Edit menu you can save and recall your custom defined set of system parameters, and you can tell form•Z what options to apply next time you launch it.

Preferences...

When this item is selected it invokes the Preferences dialog, shown in Figure 3.2.7.1. This dialog is structured hierarchically and displays a list of categories on its left side. When an item from the list is selected by clicking the mouse on it, the corresponding preference settings for the selected subject are shown on the right portion of the dialog. Each of these categories can also be accessed individually in its own dialog by assigning a key shortcut in the corresponding item in the Key Shortcuts Manager dialog (see section 3.2.5).

At the top level, the list on the left side of the Preferences dialog, contains two categories: System and Project. Some of the preferences belong to the former and some to the latter and for each category they are saved and recalled differently.

To save Project level preferences, first make the desired changes to project level settings, such as in the Working Units or Window Setup dialogs. Then save the project under a name of your choice. This project now becomes a “template” project and you can instruct form•Z to pick up the settings of this project for future projects. You do this by selecting Use Template Project under the Project: General category and by using the Choose Template File... button to select the project to be used as a template. After the selection the name of the template project is displayed in the template box under the Project: General category. To retain this setting, you should select Use Preference File (see below) or Use Previous Session Settings from the System: General category.
To save a preference file, after you make the desired changes, under the **System: General** category, select **Use Preference File**. This activates the **Save Preferences...** and **Load Preferences...** buttons. You click on **Save Preferences...** and, in the standard **Save As** dialog that is invoked, type the name under which you wish your preferences to be saved. After you are done, the name of your preference file should show up in the preferences box, under the **System: General** category. If you now Quit/Exit form•Z and launch it again, the preferences you saved at both the system and project levels will appear in your new projects.

**System: General**

This category tells form•Z which preference file to use and what action to perform at start-up.

**Preferences File:** This group determines what preference settings will be applied when form•Z is launched. When the selected option is changed, you have the opportunity to change preferences immediately or at the next form•Z launch.

- **Use Defaults:** When this option is selected, the system will apply its standard defaults.
- **Use Previous Session Settings:** With this option selected, form•Z will use whatever the state of the parameters were at the closing of the most recent session. That is, each time you quit the program, your current parameters are automatically saved by the system, and can be used again the next time you run the program if this option on.
- **Use Preference File:** With this option selected, the system uses the parameters saved in the file whose name appears in the **Current Preferences File** field.

**Startup Command:** This group determines what command is performed when form•Z is launched as follows:

- **New Project [Model]; New Project [Draft]:** Creates a new project and opens a modeling or a drafting window, respectively.
- **New Imager Set:** Creates a new project and opens an Imager set window.
- **Open:** Invokes the Open dialog for the selection of an existing file.
- **None:** Does nothing.

**Multi Processing:** The raytrace rendering mode of RenderZone Plus can use multiple processors concurrently. That is, instead of rendering one scanline at a time, raytracing can render two scanlines simultaneously using two processors. This speeds up rendering by a factor of 1.6 to 1.8. If more than two processors are available, more scanlines can be rendered simultaneously. The options in this group, allow you to allocate the desired number of processors.

- **Use All Processors (n):** When this option is on, all available processors are used. The number of processors available on a machine is shown in parentheses.
- **Use n Processors:** When this option is on, n processors are allocated for raytrace rendering. Sometimes it may be desirable to free some processors for other tasks, in which case a value less than the number of available processors should be entered.
System: Language

This category contains all language related settings and is intended to facilitate localizations by enabling the ability to switch between different languages. Note that the defaults for the language preferences are determined by the operating system settings. On OS X they are found in the “International” system preferences. On Windows they are found in the Regional and Language Options control panel. This category is shown in Figure 3.2.7.2.

![Figure 3.2.7.2: The Preferences dialog with the System: Language options displayed.](image)

**Language**: This menu determines what language is used throughout the form\-Z interface. This menu contains a list of all languages installed with the form\-Z Installer. The default language is determined from the operating system’s language. If the default system language was not installed by the form\-Z Installer, English is used as the default language. Note that changes to this setting will not take effect until the next time form\-Z is launched.

**Field Separator**: This character is used to delimit lists of numbers such as found in the prompts palette.

**OS Default**: When this option is selected the default is determined by the settings on your computer. The default is the comma (,) for localization settings where a period (.) is used as the decimal separator. The default is a semi colon (;) for localization settings where a comma (,) is used as the decimal separator.

**Custom**: When this option is selected the character in the corresponding text box is used as the field separator.

**Default Units**: This group determines if the default work units are English or Metric.

**OS Default**: When this option is selected the default working units type is determined by the settings on your computer.

**English**: When selected, English will be the default unit type.

**Metric**: When selected, Metric will be the default unit type.
System: Interface

Selecting Interface in the list displays the respective options, which are organized into two tabs: Dialogs and Palettes, as shown in Figure 3.2.7.3.

The Dialogs tab

Preview Dialogs: This slider bar determines the size of the preview area in dialogs with previews. When the slider bar is to the left the preview will be at the smallest possible size (193 pixels) and when it is to the right it will be at its largest (593 pixels). The default setting is determined based on the size of your monitor such that dialogs with previews are shown as large as possible without exceeding your monitors extents.

Fonts: The options in this group control the text size and font used in the form·Z dialogs. Note that the changes in the font settings are not visible until the next dialog is opened. The default font settings vary depending upon the language and operating system.

Font Name: This is the list of available fonts for displaying form·Z dialogs.

Font Size: This is the size of the font in pixels.

Figure 3.2.7.3: The Preferences dialog with the System: Interface options selected and the Dialogs tab displayed.
Menus

Reset To Defaults: This button resets the font name and size to the system defaults.

Dialog Centering: This group of options determines the position where dialogs are placed.

Automatic: When this option is on, which is the default, the program determines where to place a dialog; it is centered about the respective icon. The position changes when the tool palette containing the respective tool is repositioned.

Custom Location: When this option is on, the locations of the dialogs are determined by the values in the H and V fields, next to this option. Note that, by this method, the position of a dialog is independent of the position of its tool icon.

Order Of Buttons: The options in this group determine the order of the standard OK, Cancel and Reset buttons at the bottom of the dialog. Reset, Cancel, OK is the default MacOS and OS X layout and OK, Cancel, Reset is the default Windows layout.

Use Sheet Dialogs For Alerts: This option only applies to OS X. When on, alert messages are displayed in sheets that slide out from the top of the active window rather than in an independent alert window.

List Colors: This group contains items that control the colors used for the background of lists in Palettes and Dialogs. The alternating colors of the list background make lists easier to read. These options are available for both the Dialogs and Palettes tabs.

Title Color: This is the color that is used for the background of the title bar of the list.

Even Color: This is the color that is used for the first and all odd rows of the list.

Odd Color: This is the color that is used for the second and all even rows of the list.

Use Logical Sorting: This option determines how lists are sorted in form•Z. Lists appear throughout form•Z in a number of interfaces such as the Objects, Lights and Layers palettes. When this option is enabled, a better more logical sorting method is used. When this option is off, a more traditional computer language based sorting method is used. The former yields better results for items that contain numbers or special language characters. This option is on by default. It is should only be necessary turn this option off if a situation is encountered where a list takes an extreme long time to sort because of its size or other criteria.
The **Palettes** tab

This preference category contains options for all of the **form·Z** palettes. The content of the tab is shown in Figure 3.2.7.4.

![Preferences Dialog]

**Figure 3.2.7.4:** The **System: Palettes** options in the **Preferences** dialog.

**Use Tabs:** When this option is selected, palettes which contain complex options are displayed using the standard “tab” interface. That is, the parameters are placed into groups, only one of which is displayed at time. A group of parameters can be displayed by selecting the tab which contains the name of the group. This option is on by default. When this option is off, the groups of parameters are displayed vertically in the palette. Figure 3.2.7.5 shows the **Use Tabs** option on and off.

![Tool Options]

**Figure 3.2.7.5:** The **Use Tabs** option (a) on and (b) off.

**Fonts, Font Name, Font Size:** These options are as for the **Dialogs** tab.
System: Scratch Disk

Selecting **Scratch Disk** in the list displays the respective options, as shown in Figure 3.2.7.6. These options provide control over the location of the temporary (or scratch) files of *form•Z*. These include the files that contain the undo and redo records and rendering buffers. These files are hidden so that they are not accidentally deleted while the program is running. The default scratch location is set to the system default temporary directory. You may want to specify a different location for the scratch files if the hard disk where the *form•Z* application is located is very full or if you want to use a RAM disk to increase performance.

![Preferences dialog with System: Scratch Disk options displayed.](image)

**Figure 3.2.7.6:** The Preferences dialog with the System: Scratch Disk options displayed.

**Use System Temporary Folder:** When this option is selected, the scratch files are stored in a folder named “form•Z Scratch” in the system’s temporary directory. The location of the System Temporary directory varies depending on the operating system. Please consult your operating system’s documentation for the exact location.

**Use Application Folder:** With this option on, the scratch files are kept in the folder that contains *form•Z*.

**Custom:** When this option is on, a user-defined location for the scratch files can be specified. When you click on the **Choose Location...** button, the Standard Folder Selection dialog appears (see Figure 3.2.7.7). This dialog is similar to the dialog used to open files. Once you have selected the folder where the scratch files will be saved, click on the **Choose** button. This dialog also contains a **New** button, which can be used to set up a new folder. A scratch disk on RAM can be selected through the **Custom** option. Note, however, that such a scratch disk should first be set at the operating system level before running *form•Z*.

![Select Scratch Location dialog.](image)

**Figure 3.2.7.7:** The Select Scratch Location dialog.
Location: This information field shows the full path of the location where the scratch files will be saved. Because the scratch files are already open when the application is running, their location will change only when form·Z is launched again.

Current Location: This information field shows the current location of the scratch disk. This is useful since the new setting entered in the Location field does not take effect until form·Z is relaunched.

System: Recent Files

Selecting Recent Files in the list displays the respective options, as shown in Figure 3.2.7.8. These options affect the Open Recent pop up menu (see section 3.1).

![Preferences dialog with System: Recent Files options displayed.](image)

Figure 3.2.7.8: The Preferences dialog with the System: Recent Files options displayed.

Maximum Number Of Files: This text field determines the maximum number of files that are remembered and shown in the Open Recent menu. When this maximum is reached, the oldest file is “forgotten” so that a newer file can be remembered. Old and new refers to when the file was last opened or saved within form·Z, not in terms of when the file was created. The default is 10, the minimum is 1, and the maximum is 100.

Show Full Path In Menu: When this option is on, which is the default, the full path for accessing the file is shown in the menu. The full path is the complete hierarchy of the file system directory necessary to access the file. When this option is off, only the file name is shown in the menu. This option is useful for distinguishing files with the same name but different locations. If the path is too long to fit in the menu, the middle of the path is truncated.

Sort Menu By Name: When this option is selected, the Open Recent menu is sorted in alphabetic order by the file name. When this option is off, the menu is sorted by order of use with the most recent file at the top of the menu and the “oldest” at the bottom.

Clear Open Recent Menu, Clear Import Recent Menu, Clear View Recent Menu: Clicking on one of these buttons clears the respective Open Recent files list. These buttons are dimmed and inactive if the respective files list is empty.
System: Updates

The Update Manager is designed to keep users informed of new updates and versions of form•Z. The Update Manager checks with the auto•des•sys server to see if there are any patches available for form•Z as well as for any new versions of the plugins that are loaded on form•Z at the time the update check is run. The Update Manager will download the patch and any selected plugins as well as the Patcher application and the corrections log file. The Update Manager will update form•Z with the patch file and create a backup copy of the current version of form•Z.

The Updates tab in the Preferences dialog, shown in Figure 3.2.7.9, is where options are set for checking for updates.

Check For Updates At Startup: When this option is on, form•Z will check for updates at the time it is launched.

Check For Updates: This pop up menu, which is only available when Check For Updates At Startup is on, contains three items: Daily, Weekly, and Monthly. Selection of one of them determines how frequently form•Z will check for updates when it starts up. For example, if Daily is selected, form•Z will check the first time it is launched in a given day.

Check Now: Clicking on this button will check to see if there are any new updates available now. Clicking Check Now will cause the Preferences dialog to close and any changed options in the Preferences dialog will be saved.
The **Update Availability** dialog, shown in Figure 3.2.7.10, is displayed when **form•Z** is done checking for updates. If there is an update available, the group **There Is A New Patch Available** will be displayed. Inside that group, the first item is a string describing the update.

If there are any updates to loaded plugins available, the group **The Following Plugins Are Available** will be displayed. Inside that group, a list of plugins preceded by checkboxes is displayed. All plugins that are checked will be updated.

The **Download Updates** button will start the update process. **form•Z** will proceed to download the updates to the scratch disk. A progress bar will be displayed to show the progress of the download. Once the download and extracting of the updates is complete, the prompt **Updates Will Not Take Effect Until The Next Time form•Z Is Launched. Click OK To Quit And Relaunch form•Z Now.** will appear, as shown in Figure 3.2.7.11. Clicking **OK** will cause **form•Z** to quit and update immediately, while clicking **Cancel** will hold off on updating **form•Z** until the next time **form•Z** quits. In either case, after **form•Z** quits, the patcher application will update and relaunch **form•Z**.

If there is a new version of **form•Z** available that is not an update, the group **There Is A New Version Available** will be displayed.

Inside that group, the first item is a string describing the new version. The **Open Version Information Page...** button will launch the default web browser with the web page that has information about the newest version of **form•Z** available.

If there are no updates available, **There Are No New Patches Available** will be displayed.

The **Update Availability** dialog will not be displayed if there are no updates available and the update check was run automatically at startup.
**Project: General**

This category tells form-Z which preferences should be applied to a new project. Clicking on it displays the respective options, as shown in Figure 3.2.7.12.

![Preferences dialog with the Project: General options displayed.](image)

**Figure 3.2.7.12:** The Preferences dialog with the Project: General options displayed.

**New Projects:** The options in this group determine where the initial values of new projects will be coming from.

**Use Defaults:** When this option is selected, the form-Z project defaults are used for all new projects. All of the defaults are fixed by form-Z, except for the default working units.

**Default Working Units...:** Clicking on this button invokes the Default Working Units dialog shown in Figure 3.2.7.13. This is a partial version of the Project Working Units dialog invoked from the Options menu. Its options are discussed in section 3.7.1.
Figure 3.2.7.13: The Default Working Units dialog with (a) the Numeric Options and (b) the Angle Options tabs displayed.

Use Template Project: When this option is selected, the contents of a form-Z template file are used as the starting content for all new projects. A template file is simply a previously saved form-Z (.fmz) file that contains your preferred starting environment. The text field of this option shows the path and name of the current template file.

Choose Template File...: Clicking on this button invokes the standard Open dialog for selecting the desired form-Z template file.

Ask For Each Project: When this option is selected, the New Projects dialog, shown in Figure 3.2.7.14 is invoked each time a new project is created. This dialog contains the same options found in the New Projects group of the general Preferences dialog (see above).

Figure 3.2.7.14: The New Projects dialog.
**Menus**

- **The Edit menu**

**Project File Options:** These two options control actions associated with saving project documents. Both options are also available in the *Save As...* dialog (see subsection 3.1.3).

**Keep Backup:** When this option is selected, a backup document is made each time you *Save* or *Save As* with the same name. The name of the backup file is constructed using the name of the original file and the suffix “.fzb.”

**Compress:** When this option is selected, your project document will be compressed each time you save. This option results in smaller files but requires additional time when saving.

**Continuous Window Tool Control (Zoom By Frame/Pan/View Tools):** As the clarification in the parenthesis indicates, this option affects the mode in which the window tools that require graphic input are executed. When selected and one of these tools is executed, the system designates exclusive control to that tool and makes all the other tools inactive, which is indicated by graying them out. This allows the continuous execution of the operation without reselecting the tool. When off, each time the tool is selected it executes a single operation. By default this option is on.

**Show Window Zoom Percentage:** With this option on, the current percentage of zooming relative to the current working scale is shown in a box at the bottom of each window to the left of the horizontal scroll bar. A percentage of 100% indicates that the current zooming is at the display scale. A smaller percentage indicates how much the window is zoomed out, and a larger percentage indicates how much the window is zoomed in. This number is updated to reflect the current zoom percentage when a new view is selected, or a Zoom or View Navigation tool is used. Clicking in this area invokes the *Zoom Percentage* popup menu (Figure 3.2.7.15), which is used to select zoom percentages.

![Figure 3.2.7.15: The Zoom Percentage popup menu on the window's lower margin.](image)

**Memory Display:** This popup menu (Figure 3.2.7.16) controls the memory display box at the bottom of each project window. It has three options. *Off* turns off the memory box.

**Show Available Memory:** When this item is selected (default on Macintosh), a numeric display of the amount of memory that is available to *form-Z* is shown. When running on MacOS, this is the amount of memory allocated to the program (as set in the file information box in the Finder), less the amount that has been consumed. On Windows, this is the total memory available for the system. In both cases this number includes both physical and virtual memory.
**Show Memory Meter:** When this is selected (default on Windows), a graphic representation of the amount of free memory is shown in a format similar to a progress bar. The more memory is used, the more area of the bar appears solid from left to right. The bar is drawn in green to represent physical memory, and red to represent virtual memory (if available). As the memory consumption increases into virtual memory, the red area increases also, warning of a loss of performance.

**Save Prompts In TEXT File:** Selection of this option causes the content of the prompts palette to be written into a text file. This is useful if you need to go back and look at information in the prompts palette further back than the 20 lines it saves in the scroll area. The TEXT file created by this option is named `form·Z Prompts`, and it is always saved in the `form·Z` application folder. This file is not deleted when you restart `form·Z`, therefore you can go back and see what was done in previous `form·Z` sessions. Because this file will continue to grow in size, you should periodically throw it away to save disk space. This file can be viewed by virtually any text editor or word processor.

**Always Open File Format Options Dialogs:** When this option is on, the file format options dialog for the selected export format is automatically invoked when using the **Save As...** item to save the project. If this option is off, the **Options...** button in the **Save Project As...** dialog needs to be clicked whenever it is necessary to select options other than the defaults, when exporting to a file format.
**Project: Auto Save**

Selecting this category displays its options, as shown in Figure 3.2.7.17. They allow you to turn on the **Enable Auto Save** option and to select the conditions under which it will operate. Note that the project needs to be saved at least once after it has been created in order for this option to take effect. If the project still carries the **Untitled** label, automatic saving can not take effect.

![Figure 3.2.7.17: The Project: Auto Save options in the Preferences dialog.](image)

**Enable Auto Save**: When this option is selected, form•Z saves the active project according to the criteria specified by the following options. This option is off by default.

**Every n Minutes**: When this option is on, the frequency at which the project is saved is determined by intervals of \( n \) minutes, where \( n \) is the number entered in its numeric field.

**Every n Operations**: With this option on, the frequency of the saves is set by the number of operations executed. It saves every \( n \) operations, where \( n \) is the number entered in its field.

**Save To Project File**: When this option is on, the automatic save writes the data directly to the project file.

**Save As Copy**: This option creates a copy and leaves the previously saved project as is. The files are saved using the project’s name with “.sav” as the file extension instead of the “.fmz” extension. The following options allow you to specify where the copy will be saved.

**Use Project Folder**: When this option is selected, the auto save file is saved to the folder in which the respective project file is located.

**Use Application Folder**: With this option, the auto save file is saved to the folder in which the form•Z application resides.
**Use Scratch Disk:** This option directs the auto save files to the same location that is selected for the scratch disk. Note that Macintosh operating systems will delete the contents of the temporary folder at restart or log-out. If the scratch disk preferences are set to use the temporary folder using this option for Autosave files may not be desirable.

**Set Location...:**
Clicking on this button invokes the standard folder selection dialog (Figure 3.2.7.18), through which the folder for the auto save location may be selected, or a new folder may be created and selected. The currently selected auto save location is displayed in the Location text area below the button.

**Incremental File Names:**
When this option is selected, with each automatic save, a new file is written. Each new file’s name is created by appending the current date and time to the project’s name. File names may get concatenated, depending on the maximum file name length allowed in the system under which form-Z is running.

*Figure 3.2.7.18: The folder selection dialog is used to set a location for the Auto Save.*
Project: Fonts

Selecting this category displays its options, as shown in Figure 3.2.7.19. These allow you to determine the time at which text fonts are loaded, as well as the locations where the fonts can be found.

![Figure 3.2.7.19: The Project: Fonts options in the Preferences dialog.](image)

**Load Fonts At Launch:** Selection of this option causes all the font information to be loaded at startup, instead of the first time text is applied. This is off by default.

**Default Font:** This is a pop up menu from which a default font may be selected. The default font is used:

- when a project contains plain or drafting text with a font that is not available on the computer;
- when text objects are edited that use a font unavailable on the computer;
- when placing text for the first time; the default font appears selected in the Text Editor dialog.

**Stick Font File Paths...:** This area contains a list of folders/directories that will be searched when loading Stick fonts in the drafting environment.
**Add...**: This button is used to add a folder/directory to the list. Clicking on it invokes the Select Postscript Location dialog, which is similar to the standard Open dialog. After selecting the desired location, click on the Use Current button to add it to the list.

**Remove**: Clicking on this button removes the highlighted item from the list. If no item is selected or the list is empty, this button is dimmed and inactive.

**Absolute Path, Relative To form** • Z Application: Selecting one of these radio buttons applies the respective type to the highlighted path.

**Search Nested Folders**: When this option is on, the file search extends to all the folders inside the listed folder and all the folders inside them, until the desired file is found. When this option is off, the file search takes place only at the level of the folder listed in the selected search path.
**Project: File Search Paths**

Selecting this category displays its options, as shown in Figure 3.2.7.20. These options are used to specify the search paths (list of directories), when looking to locate a symbol library or an image file used in a project. The search paths specified in this dialog are used when a symbol library or an image cannot be found in the *expected location*, which is where an image was at the time it was applied or a symbol library was at the time it was defined.

![Figure 3.2.7.20: The Project: File Search Paths options in the Preferences dialog.](image)

By default this dialog initially appears with four preset search paths, as discussed below. These paths can be deleted, new paths can be added, and the attributes of paths can be changed using button commands and options available in the dialog.

File search paths can be either *absolute* or *relative*. An absolute path is a complete description of the location of the directory from the top of a volume to its location. A relative path is a location based on another location. For example, if a search path is set to be relative to entity A, the search is applied to the folders contained in the folder that contains A. There are two relative options that can be selected for file paths (see below).

At its top, the *File Search Paths* area displays a list of the search paths. A check mark in the column on the left indicates that a path is *active*. Clicking in this column toggles the path between active and inactive. Inactive paths are not searched, but they remain in the list for future use. Search paths can be highlighted by clicking on them. When highlighted they can be removed or changed from absolute to relative or vice versa. Search paths also can be repositioned in the list by dragging them. Their order in the list is significant speed-wise, as the search always starts at the bottom of the list and moves upward.
Add: Clicking on this button invokes the standard folder dialog for selecting a folder. The new directory is then added to the list.

Remove: Clicking on this button removes the highlighted path from the list.

Absolute Path, Relative To form•Z Application, Relative To Project File: Selecting one of these radio buttons applies the respective type to the highlighted path.

As mentioned above, by default, the paths list initially contains four search paths, all of which are relative; two are relative to form•Z and two to the project file. They are: “form•Z Symbols,” “form•Z Textures,” “Project Symbols,” and “Project Textures.” Note that this default setup assumes that there are folders labeled “Symbols” and “Textures,” as is typical with graphics applications.

Search Nested Folders: When this option is off, the file search takes place only at the level of the folder listed in the selected search path. When on, the file search extends to all the folders inside the listed folder, and all the folders inside them, until the desired file is found.

Project: Undo

Selecting this category displays the Undo options, as shown in Figure 3.2.7.21. These options are the same with the options in the Undo Options dialog invoked from the Undo Options... item in the Edit menu. They are discussed in section 3.2.1.

![Preferences](image)

**Figure 3.2.7.21:** The Project: Undo options in the Preferences dialog.
**Project: Warnings**

Selecting this category displays its options, as shown in Figure 3.2.7.22. These options determine when warnings will be posted by the program.

![Figure 3.2.7.22: The Project: Warnings options in the Preferences dialog.](image)

**Show Error Messages Immediately**: When this option is on, each time the program encounters an irregularity and needs to inform the user about it, it will present an alert box with a message. This message is also entered into the **form•Z Messages** dialog, which can be invoked by selecting the **Error Messages...** item from the **Help** menu. When this option is off, the message is only entered into the Messages dialog.

- **Warn Before Clearing Ghosted**,
- **Warn Before Clearing Unghosted**,
- **Warn Before Clearing Rendering Memory**,
- **Warn Before Generating 20 Or More Objects**,
- **Warn When Fonts Are Missing Or Not Found**,
- **Warn Before Clearing Control Parameters**: When these items are on, a warning will be issued before the respective action is taken. When they are off, no warning will be issued. Default for all options is on.
**Project: Modeling: Animation**

Selecting this category displays its options, as shown in Figure 3.2.7.23.

![Preferences dialog](image)

*Figure 3.2.7.23:* The **Project: Modeling: Animation** options in the **Preferences** dialog.

**Enable Animation:** When this option is on, the animation features in form•Z are active and can be used. They are inactive and unavailable otherwise. This option is on by default.

The animation features of form•Z are discussed in Chapter 7 of this User's Manual.
**Project: Modeling: Model Type**

Selecting this category displays its options, as shown in Figure 3.2.7.24. These options determine what type of a model will be produced when applying operations that mix facetted and smooth objects. This affects the insertions, Booleans, sections, trim, and split operations.

![Figure 3.2.7.24: The Project: Modeling: Model Type options in the Preferences dialog.](image)

**Enable Smooth Modeling**: When this option is on, the form-Z smooth modeling features are available. This includes smooth modeling options of tools such as the Object of Revolution, Helix, and Sweep Along Path. It also includes tools such as Smooth Rounding, Fillet, and Loft. This option also enables support for trimmed nurbs objects. When this option is off only the classic facetted modeling features are available. Note that this option is disabled if the smooth modeling module is not installed. The smooth modeling module is installed by default with the form-Z Installer and can be optionally not installed by selecting the Custom Install option and deselecting the smooth modeling module from the custom features list.

**Set Tools To Facetted**: When this is selected, all tools which have options for Facetted and Smooth modeling are set to use the facetted options.

**Set Tools To Smooth**: When this is selected, all tools which have options for Facetted and Smooth modeling are set to use the smooth options.

**Execute Operations Between Smooth And Facetted Objects**: The options in this group determine what type of object will be created when mixing the two types in certain operations.

**Always Smooth, Always Facetted**: Selecting one of these options results in a smooth or facetted object, respectively.

**Facetted If Object Has More Than n Faces**: With this option selected, the resulting object will be smooth only if it contains at most n faces. If it has more than n, it will be generated as a facetted object. The value of n is entered by the user in the field provided in the dialog.
Project: Modeling: Radiosity

Selecting this category displays its options, as shown in Figure 3.2.7.25. Note that this category is only available in form-Z RenderZone Plus.

![Figure 3.2.7.25: The Project: Modeling: Radiosity options in the Preferences dialog.](image)

Warn Before Entering And Exiting Radiosity: With this option on, a warning is posted when:

- Generate Shadows is on in the Radiosity Options dialog, but no light has its shadows attribute on.
- A distant light is active but does not cast shadows.
- Radiosity is exited by selecting Exit Radiosity, closing a window, or switching to another project.

Save Solution In Project File: When this option is on (default), the current radiosity solution is included in the project file, when Save or Save As... is selected from the File menu. When such a file is opened, and Generate Radiosity Solution* is selected, the radiosity solution is restored, and the radiosity process continues from where it was interrupted.

Preserve Solution During Working Session: When this option is on (default is off), the current radiosity solution is not discarded when selecting Exit Radiosity but is saved to a file. If no change is made to an object or light and Generate Radiosity Solution* is selected, the radiosity process continues from where it was interrupted. However, if such a change were made, the radiosity solution also has to change and cannot continue, regardless of whether this option is on or off. This option also affects whether a radiosity solution will be preserved to be continued later when switching active projects. That is, given that only one radiosity solution may be in progress at any time, which is the solution of the active project, when switching the active project, the radiosity solution of the previous project (when there is one in progress) cannot be continued. It has to be discarded or saved. With this option on, the solution is saved and is restored when the project becomes active again. If this option is off, the current radiosity solution is always discarded when selecting Exit Radiosity or when switching projects.
**Project: Modeling: Textures**

Selecting this category displays its options, as shown in Figure 3.2.7.26. These options are only present in form•Z RenderZone Plus.

**Texture Size**: This group of options can be used to limit the size of textures in order to optimize memory consumption. Note that these options apply to image, transparency, bump, reflection, and environment maps. They do not apply to background images.

- **Limit Size To n Pixels**: When this option is selected, all textures whose width or height is larger than the selected pixel resolution are sampled down to the selected resolution.

- **Round Textures Above n Pixels**: When this option is selected, all textures whose width or height is larger than the selected pixel resolution are scaled up or down to the closest resolution, which is a power of 2. For example, a 280x280 pixel texture is sampled down to 256x256 pixels, whereas a 480x480 pixel texture is scaled up to 512x512 pixels.

- **Full Texture Size**: When this option is selected, no adjustment is made.

**Limit Textures At Preview To n Pixels**: When textures are rendered in a preview window, the textures are generated at a resolution of \( n \) pixels. Preview textures are also rendered in the Texture Map Controls, Decals, and Surface Style Parameters dialogs. When a low resolution is selected, less memory is used and the texture generation is faster, but the textures displayed in the preview will be at a lower quality.

**Post Texture Warning**: When this option is selected, form•Z will issue error messages when errors related to textures (missing files, insufficient memory, etc.) occur. When this option is off, no message is posted and the rendering proceeds by substituting the default form•Z texture. This option is on by default.
Project: Drafting: General

Selecting this category displays its single option, as shown in Figure 3.2.7.27.

![Preferences dialog]

**Figure 3.2.7.27:** The Project: Drafting: General options in the Preferences dialog.

**Store New Image Elements In Project:** When this option is selected, which is the default, images added to a drafting project as image elements are stored with their full resolution directly in the project. If this option is off, these images are not stored in the project but the project is linked to a file that contains the image. In either case a low resolution image is stored in the project. If this option is on and the image requires more than 256 K of space, a warning is issued offering the option to either store the image or link to it. Which option is preferable depends on how much memory and disk space you have available. Images stored with a project redraw and print faster.

**Script: General**

Selecting this category displays its options, as discussed in section 3.8.1 of the form-Z SDK documentation.
Where are the preference files stored and how are they called

The preference settings are stored in a number of files that are based on the type of information they contain. The preference settings applied to new projects (such as the default surface styles, default layers, etc.) are placed in template files (see below). All the preference files maintained by form-Z are stored in the system standard preferences folder in a folder named “autodessys”. The files are stored inside a folder named “formZ”, “formZ Render Client”, or “formZ Render Server” for each respective application. The location of the standard preference folder varies depending on the operating system. For most operating systems, a separate preference file is maintained for each user with their own login name.

The following files are automatically created and maintained by form-Z:

- **form-Z Settings.zpf**: This file stores the current settings found the System: General group from the Preferences dialog (see below). This tells form-Z what preference file to use and what action to perform at start-up.
- **form-Z Defaults.zpf**: This file stores the preferences from the last form-Z session. form-Z automatically updates this file when it quits/exits. This file is loaded when the Use Previous Session Settings option is selected from the System: General group.
- **form-Z Session.zpf**: This file stores the system persistent information such as recent files.
- **form-Z Defaults.fpo**: This file stores the current settings found in the Plugins And Scripts dialog. This tells form-Z which plugins and scripts are currently enabled.
- **form-Z Defaults.sct**: This file stores the current key shortcuts as set in the Shortcuts dialog. This tells form-Z which shortcuts are currently defined.
- **form-Z Defaults.mnu**: This file stores the current icon layout as set in the Menu Manager dialog.

The above preference files were introduced by form-Z 4.0 and they are different from previous versions. Consequently, provisions had to be made about how to handle preference files from earlier versions of form-Z. They are automatically converted to the new format when they are opened. The following changes occur:

1. The converted preference file is saved with “.400” added at the end of the file name (for example “My Pref.zpf” becomes “My Pref.400.zpf”).
2. The key shortcuts from the preference file are saved into the standard “form-Z Defaults.sct” file (see above).
3. The icon menu layout from the preference file is saved into the standard “form-Z Defaults.mnu” file (see above).
4. The project settings are saved to a form-Z project file with the preference file name (for example “My Pref.zpf” yields “My Pref.fmz”). This file is set to be the current template file. Use Template Project is selected from the Project: General preferences.
3.3 The **Window** menu

The **Window** menu, shown in Figure 3.3.0.1, consists of eight groups of items.

- The first group contains commands that control the opening and closing of associated windows.
- The second group contains items used for splitting the **form-Z** window into several portions.
- The third group contains options for window frames which are used to divide windows into multiple workspaces.
- The fourth group contains extended cursor options and the auto scroll function.
- The fifth group contains commands that allow you to turn on and off graphic references, namely the grids and axes.
- The sixth group contains the ruler options.
- The seventh group contains the snapping options, as well as the underlay options.
- The eighth group is a list of all the open windows. Its length varies and depends on the number of open windows. The names of the windows in this list also function as commands. When selected, they bring the corresponding window to the front of the screen and make it the active window. This is identical to clicking the mouse in an inactive window to make it active. The active window is indicated on the list with a check mark in front of its name.

The **Window** menu complements the commands of the **File** menu that are used to open and close projects. After a project has been opened, additional windows for that same project can only be opened through the commands in the **Window** menu. The additional windows may be modeling or drafting windows.

*Figure 3.3.0.1:* The **Window** menu (a) with a single window and (b) with six windows open.
3.3.1 Opening and closing windows

The items in the first group of the **Window** menu open and close associated windows. All the windows of the same type and of the same project are associated (see subsection 2.1.1). They are views into the same project and any changes that are made in one window are visible immediately in all the associated windows. Each window has its own graphic control and view parameters that are independent from any other window. These parameters include the viewing angle, active reference plane, size, scale, zoom/pan status, grid and snap increments, and display and rendering parameters. When a project is saved, all its windows are saved with their current graphic control and view parameters. When a project is reopened all the windows are opened with their graphic and view parameters set to the state they were at the time they were saved.

The functionality of the associated windows allows different views of the same world at different levels of detail. The first window opened in a project is opened when either one of the two **New** commands is executed from the **File** menu. This window covers the most of the screen, by default. The commands in the **Window** menu open additional windows that are gradually smaller. Multiple views work better when a large screen monitor or several monitors are connected to the computer. This allows the windows to be larger without overlapping one another. The first group of the **Window** menu also includes a command that automatically opens four small windows with three aligned projections and a 3D view.

**New Model Window**

This command opens a new window into the modeling environment for the active project. If the project already has one or more modeling windows opened, the new window becomes associated with them. This window is placed on top of all the graphics windows and becomes the new active window.

**New Draft Window**

This command opens a new window into the drafting environment for the active project. If the project already has one or more drafting windows opened, the new window becomes associated with them. This window is placed on top of all the graphics windows and becomes the new active window.
**Tile Windows**

This command pulls out the Tile Windows submenu, which contains four items (see Figure 3.3.1.1). This command is only applicable to the modeling environment, therefore, it appears dimmed and is inactive if the active window is a drafting window.

**Open**

This item opens four small associated modeling windows, each occupying one quarter of the screen. The tile windows are arranged on the screen in a way such that the graphic areas of the windows do not overlap. The lower right window is the last of the four to be opened and becomes the active window.

All four tile windows are initially opened with the same graphic parameters as the active window but different viewing parameters. The views of these windows are intended to give a complete set of views into the modeling project. The upper left window is a top view, the lower left is a front elevation, and the lower right is a right elevation. The upper right window is a 3D view identical to the view in the active window at the time the command is selected. If the active window displays a projection, then a 3D axonometric view is displayed by default. These windows function as any other modeling window. An example of tile windows is shown in Figure 3.3.1.2. Each project can only have one set of tile windows. Thus when tiled windows are open, this item appears dimmed.

**Close**

This closes all the tile windows of the currently active project. Tile windows can also be closed individually. When there are no tile windows open, this item is dimmed.

**Arrange**

If the tile windows have been moved or their sizes have been changed, this item rearranges and resizes them to their default size and layout. This item is dimmed when no tile windows are open.

**Align And Scale Views**

If the views and/or scales of the displays in the tile window have been changed, this item will return them to the initial default views and scales. This item affects the three orthographic projection windows only. The upper right tile window that is initially set to a 3D view is not adjusted. For the three projection windows the system attempts to align the views and zooming in a way such that their axes align with each other. This command works best when the tile window projection views have not been changed to 3D views.

---

Figure 3.3.1.1: The Tile Windows menu (a) without and (b) with tile windows open.
Close
This item closes the active window. Its function is identical to clicking in the close box of the active window. If the active window is the last open window of a project, then the project is also closed. If the project has not been saved since changes were made to it, the system prompts you to save the project before it is closed.

Close All
This item closes all the windows of all the open projects. This effectively closes all the projects. If a project has not been saved since changes were made to it, the system prompts you to save the project, before it is closed. After this command is executed, a new project must be opened or an existing project must be opened to continue work with the system. The Close All item does not terminate the work session. If you wish to terminate the session you should select the Quit/Exit item. Close All can also be executed by clicking on the close box of any window while pressing the option key (Macintosh) or ctrl+shift (Windows).

Figure 3.3.1.2: Tile windows.
3.3.2 Window frames

The tile windows discussed in the previous subsection allow you to split the screen into four windows, which are all views into the same project. While only one of these windows may be active, because they are associated, whatever change is made in one of them is reflected in all the others. You can also start drawing in one and continue in another. **Window frames** make this functionality easier and more flexible. Window frames divides a window into multiple portions instead of dividing the screen by creating new windows. This wastes less screen space for margins of multiple windows and makes it easier to switch between views, since there is no need to click to select a new window.

The portions of a subdivided window are referred to as **frames**. Each frame may display a different view of the project, including orthographic and 3D views. When window frames are first generated, they default to one 3D and three orthographic (top, front, and right) views. These can be changed the usual way, which is by selecting a view from the **Views** menu. You can also adjust the size of the frames and/or create new frames up to a maximum of 10. With window frames, you can conduct operations within and between the frames. The frame windows have the same controls as the regular windows and can be closed, resized, collapsed, etc. the same way. Special operations are also available for the frames.

Each frame has a border, one of which is shown in the highlight color indicating that it is **active**. Clicking inside a frame makes it the active frame. Each frame also contains a name at the middle of the upper edge, which includes a designation of the type of view displayed in the frame.

The number of frames can be decreased by closing or increased by splitting them, as follows:

![Figure 3.3.2.1: Window frames (a) as initially generated and (b) after they are resized and split.](image)
Resizing frames: A frame is resized by moving its border. To do so, place the cursor on a segment of a frame, which changes it to the resize cursor (=image). With this cursor showing, click and drag the border of the frame, which follows the motion of the mouse. When you release the mouse button, the frames on both sides of the border are resized to reflect the new border position. If you move the border to the opposite border so that there is none of the frame remaining, the frame is closed. The adjacent frame expands to fill the area.

Making more frames: A frame can be split by moving its border while pressing option (Macintosh) or ctrl + shift (Windows) at the same time. A maximum of 10 frames are allowed. The split frames initially display duplicate views until a new view is selected.

Any interactive modeling operation can be conducted completely within a single frame or between more than one frame. Between frames means that the interactive execution of an operation will continue from one frame to another. This includes the ability to start drawing in one frame and continue into another. By setting up the reference planes in the panes to different planes in 3D space, you can effectively draw in 3D space by moving from between the frames as you draw. During an interactive action, you do not need to click in a pane to make it active. As the cursor enters the area of the pane, it automatically becomes the active plane. Pressing the control key (Macintosh) or ctrl+shift (Windows) moves all colinear frame edges at once. This is useful for resizing multiple frames and keeping the frame edges aligned.

Window Frames

When this item is selected for the first time the Window Frame Options dialog is invoked for selecting the default window frame configuration. When this item is selected when window frames are active, the window is restored to a single view which is the view of the active frame. Subsequent selection of this item re-enters the window frames in the state they were in when they were deactivated.

Window Frame Options...

This item invokes the Window Frames dialog (Figure 3.3.2.2), which also is invoked the first time Window Frames is selected. It contains a graphic preview of how the window frames will be divided and sized and indications of the view types. Clicking inside a frame highlights it, makes it active, and shows its settings in the View menu.

Flip: The Flip button on the right flips the frames vertically. The Flip button on the bottom flips them horizontally.

Figure 3.3.2.2: The Window Frames dialog.
**Layout:** This menu contains eight preset settings and a **Custom** configuration, which is dimmed if the current configuration is one of the preset. **Custom** is displayed after the layout of the frames is graphically changed in the project window and the **Window Frames** dialog is invoked. One of the preset items in the **Layout** menu can still be selected to return the frames to a preset layout.

**View:** The item selected from this menu sets the type of view to be displayed in the highlighted frame.

**Keep Snap Settings Identical In All Frames:** When this option is on, the Grid, Direction, and Object Snap settings are kept identical in all frames of the window. When this option is off (default), each frame has its own independent snap settings.

**Align And Scale Frames**

This item is only available when window frames or the cone of vision frames are displayed. It is dimmed otherwise.

Selecting this item aligns and applies the same scale to the display of each and all the frames currently shown. This is the same operation with that available for the cone of vision environment and the tile windows. Note that if there are more than three frames with projection views than a best attempt is made to align the views, however, this will frequently leave some of the frames unaligned.
3.3.3  The extended cursor

The option to use a standard small cursor or an extended cursor is available in form•Z. You can switch from one type of cursor to the other through an item in the Window menu.

**Extended Cursor**

When this item is selected, a check mark is displayed in front of it, and an extended cursor replaces the regular cursor. The reverse occurs when this item is deselected. This item is off by default.

The extended cursor can also be turned on and off through a check box in the Extended Cursor Options dialog, which can be invoked directly from the Extended Cursor* item or from the Extended Cursor Options... item. The extended cursor can be used in both the modeling and the drafting environment.

When used in modeling and a 3D view is displayed, the extended cursor appears as a 3D entity consisting of three lines, which are parallel to the axes of the reference plane. When an orthographic view is displayed, the extended cursor is a 2D entity consisting of two lines which are parallel to the axes of the projection. In drafting, the cursor is similar to that of the orthographic modeling views.

**Extended Cursor Options...**

In modeling, when one of the object snaps is turned on, the extended cursor may or may not snap to the respective entities, depending on the option selected from the Extended Cursor Options dialog, shown in Figure 3.3.3.1.

**Show Cursor At:** When Mouse Position is selected from this pair of options, the full cursor is displayed at the position of the mouse, even when an object snap option is turned on and the actual input snaps to an entity. When Snapped Position is selected and object snap is on, the full cursor is displayed at the snap position.

**Cursor Size:** When Window Extents is selected from this pair of options, the cursor is automatically extended to the limits of the window. When Custom is selected, the desired length of the cursor extensions, in pixels, is entered in the text field.

**Show Cursor Marker:** When this option is selected, a small square, which is aligned parallel to the active plane, is drawn around the location of the cursor. When this option is not selected, the cursor is marked only by the intersection of its extensions. Default is on.
3.3.4 Auto scrolling

form•Z has the ability to auto scroll when the mouse cursor is moved beyond the border lines of the window, as you perform a dynamic action, such as drawing a shape. Auto scrolling is on when the Auto Scroll item in the Window menu is selected, which is indicated by a check-mark. Clicking on Auto Scroll toggles its status from on to off and vice versa. When on, you can temporarily disable auto scrolling by pressing the control key (Macintosh) or ctrl+shift (Windows), assuming you are using the default key shortcuts.

• Auto scrolling is available in both modeling and drafting. In modeling, it is available for all types of views, except panoramic and unzoomed perspective views.

• Auto scrolling takes effect as soon as your mouse cursor reaches any of the window borders and the speed of scrolling accelerates the farther you move the cursor from the border line. Scrolling will occur both when the mouse cursor is simply moved and when it is used to execute an operation, such as drawing a new shape or moving an existing object.

• Auto scrolling is disabled when the mouse cursor is on a palette, including the tool and floating palettes. This has a practical implication when, for example, you want to auto scroll in the area of the windows tool palette (left portion of the lower margin). While auto scrolling will not occur as long as the cursor is on the palette, it will as soon as the cursor is moved beyond the palette. Similar conditions apply to the left side, where the modeling and drafting tool palettes are positioned by default.
3.3.5 Axes and grids

The reference grid, the axes of the reference plane and/or of the world space can be turned on or off through three items in the **Window** menu or from options in the **Window Setup** dialog, shown in Figure 3.3.5.1. This dialog is invoked by **option** (Macintosh) or **ctrl+shift** (Windows) clicking on the three items, or directly by clicking on the **Window Setup...** item.

**Show Plane Axes***

When this item is selected (default), a checkmark is displayed in front of it, and the reference plane axes are plotted on the screen. The axes are hidden and the check mark removed when this item is deselected.

**Show World Axes***

This item works as **Show Plane Axes***, but applies to the axes of the world coordinate system.

**Show Grid***

When this item is selected (default), a check mark is displayed in front of it and the reference plane grid is plotted on the screen. The reverse occurs when this item is deselected.

**Window Setup...**

Clicking on this item invokes the **Window Setup** dialog, which contains three options similar to the above menu items and some more, as follows:

**Show View Information**: When selected, information about the current view is shown at the top of the window or the top of each frame when window frames are enabled. This includes the view type, the view name (if the view corresponds to a view from the Views palette) and the frame i.d. when window frames are enabled. When this option is off, the information is not shown. This option is off by default as it can degrade the performance of some rendering modes such as **Interactive Shaded**.

**Show Axis Marks**: When this option is on, the position of the mouse cursor is projected on the axes of the reference plane and is marked with little arrow heads, as it moves in the reference plane space. The position is projected on the horizontal and vertical axes, unless the Perpendicular switch is on, in which case it is projected on the perpendicular axis. This is off by default.
**Reference Grids**: This pair of options determines how the reference grid is drawn.

**Line Grid**: When this option is on, which is the default, the reference grid is drawn with lines.

**Dot Grid**: When this option is on the reference grid is drawn with dots. Note that this grid is slower to refresh than the line grid.

You have the option to design the reference grid by indicating where its two types of lines, the *major* and * minor*, will be drawn. This is done through the following options.

**X, Y, and Z Module**: The values entered in these fields represent the distances between each pair of major lines, expressed in real units. The starting basis of these increments is the origin of the Coordinate space. If the **XYZ Grid Lock** box is off, different values may be entered for each direction, allowing the grid squares to be any rectangular shape. If it is on, whatever value is typed in one of the fields is automatically applied to all the others. The default for **Module** is 8'-0" or 1.000 m, and **Grid Lock** is on.

The minor lines are the lines that divide the distance between the major lines. They are defined through the following parameters:

**X, Y, and Z # Divisions**: The values entered in these fields represent the number of divisions between the major lines. The number of minor lines actually drawn is one less than the number of divisions. Again, the divisions in the three directions can be entered independently, unless the **XYZ Grid Lock** is on. The number of divisions can be any integer greater than zero. The default is 4 for English and 2 for metric.

**Auto Grid Scaling**: When this option is on, the layout of the grid is automatically adjusted, according to the current display scale. That is, the **X**, **Y**, and **Z** distances of the major grid lines are automatically adjusted to values which, on the screen, produce one inch major grid squares. This option is off by default.

**All Windows**: When this option is selected, all the parameters currently shown in the dialog box are applied to all the windows of the active project. Similarly, each time a window option is changed, it affects all the windows in the active project. When this option is off, which is the default, the window option affects only the active window.
3.3.6 Rulers

Rulers may be displayed in both modeling and drafting windows. When used in modeling, they are more meaningful when displayed with one of the orthographic projections, namely the **Top**, **Bottom**, **Right Side**, **Left Side**, **Back**, and **Front** views, selected from the **Views** menu. Recall that when arbitrary reference planes are used, these views are selected from the **Plane Projection** submenu in the **View** menu.

**Show Rulers**

When this item is on, a check mark is displayed in front of it, and one or more rulers appear on the screen. This item is off by default.

The rulers may also be turned on and off through the **Ruler Options** dialog (Figure 3.3.6.1), which is invoked from the **Show Rulers** or **Ruler Options...** item.

The **Ruler Options** dialog has five sections that allow you to turn rulers on/off, to determine which of the four possible rulers will be displayed, to select which coordinates they will use and how the text will be displayed, to set up their increments, and to apply them to only the active or to all the windows of the project.

**Show Rulers**: This check box performs the same function as the **Show Rulers** item in the **Windows** menu, and the two are correlated. When this option is on, the **Show Rulers** item is also turned on, and vice versa. When on, the rulers are displayed in the active window. Which of the four available rulers are displayed depends on the following selection.

**Ruler Display**: This group of options determines which of the four rulers will be displayed.

**Top, Left, Bottom, Right**: When one of these options is selected, a ruler is displayed along the top, left, bottom, or right edge of the active window, respectively. The window can be a modeling or a drafting window. Any number and any combination of these four rulers may be selected simultaneously.

**Coordinate Mode**: These two radio buttons allow you to select either **Absolute** or **Relative** coordinates. With absolute coordinates, the origin of the rulers is aligned with the origin of the coordinate system continuously. With relative coordinates, the origin of the rulers is adjusted each time a new point is drawn, and is aligned with the previous point. This is illustrated in Figure 3.3.6.2. The default is **Absolute**.

![Figure 3.3.6.1: The Ruler Options dialog.](image)
Menus - The Window menu

Reference: These two radio buttons allow you to select as base for the rulers either the World or the Reference Plane coordinate system. Note that this option is not available for rulers displayed in drafting windows, since drafting windows do not have reference planes and work only in world coordinates. The default is World.

Show Text: When this option is selected (default) numeric labels will also be displayed with the rulers. No labels are included when this option is off.

Show Unit Indicators: This option is only available when the Show Text option is on. When selected, which is the default, unit indicators are also displayed with the numeric labels. For example “m,” “mm,” or the feet and inches signs are printed after the number, according to the currently selected unit of measurement. When this option is off, plain numbers are printed. Examples of the last two options are shown in Figure 3.3.6.3.

Ruler Increments: The options in this group are used for setting the increments of the rulers: they may be the same as those of the reference grid, which are set in the Window Setup dialog, they may be set independently from the reference grid, or they may be generated automatically.

Align With Grid Module: When this option is selected, the layout of the rulers follows the layout of the grid. The rulers are divided in the same way that the grid is divided. Text labels are displayed at points corresponding to grid major divisions, provided they fit. If they do not fit, they are displayed as frequently as possible.

Set Increments: When this option is on, the values entered in its numeric fields determine the increments of the major and minor divisions of the rulers. The Module and Division parameters work as for the grid, and can be set independently for the Horizontal and Vertical rulers. The significance of this option is that rulers may be set independently from the reference grids, which allows you to work with two different references at the same time.

Subdivisions: This option becomes available when the Align With Grid Module or the Set Increments option is selected.
When **Subdivisions** is selected, the values entered in its fields determine how the ruler divisions are subdivided in horizontal and vertical rulers when larger window display scales are used. Depending on how large the display scale is, subdivisions will be subdivided again and again, whenever the whole number of subdivisions entered in the **Subdivisions** field fits in the space of the immediately higher level of subdivision. The distance between subdivision lines can be no less than three pixels.

When this option is not selected, there are no ruler divisions smaller than those determined by the **Grid Module** settings or the **Set Increments** options. The default setting for the **Subdivision** option is two for both directions. This option is not available when the **Ruler Increments** mode is set to **Automatic Settings**.

**Automatic Settings**: When this option is selected, the ruler layout is generated automatically. Ruler markings are generated based on the origin, active unit of measurement, and the unit format as set in the **Project Working Units** dialog. Depending on the display scale, units are grouped in multiples of ten, and are divided according to the increments listed in the following table:

**Working Units: English**:

<table>
<thead>
<tr>
<th>Numeric Options:</th>
<th>Architectural</th>
<th>Engineering</th>
<th>Fractional</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foot is divided by:</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>Inch is divided by:</td>
<td>2</td>
<td>10</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>

**Working Units: Metric**: All divisions are based on the decimal system (by 10).

With automatic settings, ruler divisions are generated by first dividing each unit into halves, then subdividing these halves to generate the total number of divisions listed in the table. If this rule cannot be applied, since each ruler increment must be greater than or equal to 3 pixels, then the system divides each ruler unit by half the number of divisions shown in the table. For example, if a one foot module is displayed in the window ruler, and the screen resolution does not allow 12 divisions, when the Architectural format is used the foot will be divided into 6 units.

The following two options apply to all three ruler increment modes.

**Origin**: This option allows you to select an offset for the rulers. When on, the alignments of the rulers are offset by the values entered into the **Horizontal** and **Vertical** fields.

**Use Plot Scale**: When this option is selected, the layout of the rulers is set according to the plot scale value entered in the **Plot/Print Setup** dialog. This value is also displayed next to this option. When this option is on, the text labels are also sized according to the plot scale.

**All Windows**: The rulers are window level options. Each graphics window of a project normally has a different ruler setup. However, this option allows you to treat the rulers as project level options. When selected, the ruler parameters defined in the **Ruler Options** dialog will be applied to all the open windows of the active project. This option is off by default.
3.3.7 Snapping options

Snap Options...

This item invokes the Snap Options dialog, shown in Figure 3.3.7.1. It consists of four tabs: Grid, Angle, Radial, and Object. These groups of options correspond to the types of snaps available in form•Z and each can also be invoked as an independent dialog through Windows tools. They are discussed in subsections 2.2.4, 2.2.5, and 2.2.6.

Figure 3.3.7.1: The Snap Options dialog with the (a) Grid, (b) Angle, (c) Radial, and (d) Object tabs displayed.
3.3.8 Importing and placing underlays

An underlay can be any of the available image formats (except Illustrator). These image files can be created in form·Z or by an external source such as another application or an image scanner. It is imported into a form·Z window by selecting the **Underlay...** item and by setting the desired parameters in the **Underlay** dialog. The underlay image is placed into the active window, which may be a modeling orthographic or 3D view, or a drafting window. Different underlays may be placed in different windows of the same project.

Underlays with aerial photographs may be placed into graphics windows so that site plans may be traced over them (Figure 3.3.8.1) or they can be used as background photos in front of which a model may be generated. The underlay does not become part of a project but rather serves as a reference picture. When a project with an underlay is saved, a link to the underlay image file is stored with the project and the underlay is automatically reloaded when the project is opened. If the underlay file has been moved from its original location and is not in the same location as the project file when the project is opened, a prompt is posted asking to identify the location of the underlay file.

**Underlay...**

This item invokes the **Underlay** dialog shown in Figure 3.3.8.2. This dialog can be used either to load an underlay for the first time, or to simply change the parameters of the current underlay.

**Show Underlay**: When this check box is off, all the other options under it are dimmed and inactive. When turned on, additional options can be selected, including the loading of an underlay, which is done by clicking on the **Select Underlay File** button. In modeling, underlays can be placed either in 3D views or in projection views, each of which is controlled by a different set of parameters. Placing an underlay in a drafting window is controlled by the **In Projection Views** parameters.
**In 3D Views:** When this option is selected, the currently loaded underlay will be placed in the 3D views of a project. The size and shape of the underlay is controlled by the following option. When the active window is a drafting window, this option is dimmed and unavailable.

**Match Image Size To Underlay Size:** When this option is off, the size and shape of the underlay image will be adjusted to the size and shape of the window where a 3D view is displayed. If the proportions of the underlay are different than the proportions of the window, the underlay will be distorted to fit the shape of the window. This distortion can be avoided by turning this option on, which causes the window size and shape to be adjusted to that of the underlay. Note that this is done automatically by setting the **Image Size** options in the **Image Options** dialog to **Use Custom Size**, and setting the **Width** and **Height** parameters to numbers that match the size of the underlay.

**In Projection Views:** When this option is selected, the currently loaded underlay will be placed in a modeling window displaying a projection view, or in a drafting window. The size and shape of the underlay in such views are controlled by the parameters under this option.

**Scale:** This field determines the size of the underlay image when it is imported into a modeling projection or a drafting window. The default scale is 1/16” = 1'-1" (or 1:200 in metric). For scanned images, the scale should be set to the scale of the original scanned drawing. This field has no effect when the underlay is imported into a modeling 3D view.

**Horizontal Origin, Vertical Origin:** The values entered in these fields determine the alignment of the underlay within the coordinate system of a modeling projection or a drafting window. These parameters have no effect when the underlay is placed into a 3D view.

**Center:** This option determines the method of alignment when an underlay is placed into a modeling projection or a drafting window. When selected (default), the center of the underlay image is placed on the origin defined by the previous two parameters. If not selected, the upper left corner of the image is placed on the origin. This parameter has no effect when the underlay image is placed into a modeling window with a 3D view.

**Display On Top Of Grid And Axis:** This option determines whether the underlay is displayed on top of or under the reference grid and its axes. The default is off, which places it under the grid.

**Select Underlay File...:** This button invokes the standard Open File dialog from which an image file can be selected in the usual manner for placement as an underlay in the active window. The text field under this button displays the name and the full path of the current image. If there is no image specified, the word “None” is displayed. If you close the **Underlay** dialog with **Show Underlay** on and no image loaded, an error message will be posted.

Note that underlays are significantly different entities from those produced when importing images by using the **Import...** or **Paste Image** items. Also note that underlays are not automatically rendered as backgrounds. To do so, the image used as an underlay should also be loaded as a background when **RenderZone** is execute or the **Project Background** item should be selected from the **Background** menu in the **RenderZone Options** dialog.
3.3.9 The window list

The window list, found at the bottom of the **Windows** menu contains the names of all the open windows. The names on the list correspond to the names in the title bar of each window, and are also appended with `[Model]` or `[Draft]` depending on which environment they are in. These windows may belong to different projects and may be modeling or drafting windows. The window list is updated each time a new window is opened or an open window or project is closed.

Selecting a window name brings the corresponding window to the front of the screen and makes it the active window. This is identical to clicking the mouse in an inactive window to make it active. The active window is indicated in the list with a check mark in front of its name. This method of window selection is especially convenient when a window is completely covered by other windows.
3.4 The Heights menu

The **Heights** menu, shown in Figure 3.4.0.1, is used to set the *height mode* and to select the *active height*. Since heights are only used by the modeling environment, this menu is only present when a modeling window is active.

The **Heights** menu consists of three groups of items. The first group contains a single command, which sets the system to an *interactive* height input mode. In this mode you determine the height of the newly generated objects at the time you draw them, rather than by using a preselected height.

The second group contains a list of fourteen preset heights, any one of which can be deleted or changed and new numeric items can be added to the list. Any of the numeric items can be selected to set the active height value, which can be positive or negative. The heights are used for a variety of modeling operations and their signs are frequently quite significant.

The third group contains the **Custom...** and the **Edit Menu...** items, which can be used to customize the **Heights** menu.
**Graphic/Keyed**

When this item is selected, the height of a 3D extrusion or enclosure is defined interactively, either graphically through the use of the mouse or by numeric input. As soon as the base shape is drawn, the object is generated with its height rubber banded. It follows the motion of the mouse, which is locked to the perpendicular direction. The height definition is completed by the next click of the mouse. The graphic height can be positive or negative. During the graphic height definition process, the actual height (Z) value is displayed in the Z field of the Coordinates palette and in the Prompts palette.

When this command is selected and the Prompts palette is used to define the base shape by entering numeric values through the keyboard, as soon as the input of the base shape is completed, the system prompts you to also enter the height numerically. Keying in a height and pressing return completes the creation of the object. An example of the values entered for a 28' x 34' x 8' cuboid is shown in Figure 3.4.0.2.

![Figure 3.4.0.2: Selecting the Graphic/Keyed item from the Heights menu and using the Prompts palette with numeric input to create a cuboid.](image)

**Custom...**

This item invokes the Custom Height dialog (Figure 3.4.0.3). It contains a numeric field for entering a specific height value and the Add To Menu... button. Clicking on the button enters the value that appears in the numeric field into the Heights menu. If the value in the field is the same with a height value already in the menu, the Add To Menu... button appears dimmed.

![Figure 3.4.0.3: The Custom Height dialog.](image)

*Menus • The Heights menu*
Edit Menu...

This item invokes the Heights Menu dialog shown in Figure 3.4.0.4. It consists of a numeric field where values can be entered and edited, a scrollable height list, and button commands that allow you to change the height values in the Heights menu, to save the menu values, or to load previously saved values.

Clicking on a value in the height list selects it and enters it in the numeric field. The value in the numeric field can be changed using the standard text editing methods.

Add: This button is active only when the value in the numeric field is different from all the values that are already contained in the height list. Clicking on it enters the value in the height list (and in the Heights menu).

Remove: Clicking on this button removes the selected value from the height list (and the Heights menu). When values are added to or removed from the height list, the height list is automatically sorted to retain its numeric order.

Save...: Clicking on this button invokes the standard save dialog, which allows you to save your height definitions into a file under a name entered in the save dialog.

Load...: Clicking on this button invokes the standard open file dialog which allows you to select the name of a previously saved heights file.

form-Z The default heights are determined by the Data Scale and Unit Type (English, Metric) options in the Default Working Units dialog. This results in eight possible default configurations as shown in Figure 3.4.0.5.

Figure 3.4.0.5: The content of the preset height files: (a) Mini English, (b) Small English, (c) Medium English, (d) Large English, (e) Mini Metric, (f) Small Metric, (g) Medium Metric, and (h) Large Metric.
3.5 The Views menu

The Views menu (Figure 3.5.0.1) contains commands that manipulate the viewing parameters for the modeling and drafting environments. The modeling Views menu offers a variety of items for setting and changing the type of 3D view (axonometric, isometric, oblique, perspective, or panoramic), as well as the position of the viewer, the viewing angle, the sun position and other viewing parameters. These items are discussed in sections 3.5.1 through 3.5.6.

The drafting Views menu offers simple 2D rotation and scale control for drawings. The drafting Views menu is discussed in section 3.5.7.

The Views menu is complemented by additional graphic methods for manipulating the viewing parameters which are provided by the View in the window tool bar (see section 2.2.8). The Views palette and Views Dialog are frequently used with the Views menu.
3.5.1 Setting the viewing angles

The modeling objects contained in a project may be viewed from any position in the 3D world space. That position may be derived either by rotating the objects, while the position of the viewer remains constant, or by rotating the position of the viewer, while the objects’ positions remain constant. These two methods are actually equivalent and they simply represent two different ways of comprehending the operations which involve view transformations in 3D space.

To mentally trace the effects of the viewing rotations, begin with a cube which sits on the XY plane and is viewed from its top (Figure 3.5.1.1(a)). The screen coordinate system always remains fixed to that position. The object and its coordinate system are rotated around the axes of the screen coordinate system, in Z, Y, X order. As an object is rotated, the viewer’s position remains fixed on the positive Z axis of the screen. This is illustrated in Figure 3.5.1.1, where the cube is first rotated by 30° around the screen Z axis, and then by 60° degrees around the X axis. Note that all the predefined viewing angles use a 0° value for the Y rotation. This preserves the verticality of lines parallel to the Z axis, in axonometric views.

![Figure 3.5.1.1: Application of viewing angles to a cube. (a) Positive XY projection of the cube. (b) 30° rotation about the screen Z axis. (c) 60° rotation about the screen X axis.](Image)
The first group of items in the **Views** menu contains preset viewing angles. When one of them is selected, it applies the indicated viewing angles to the active window. How the viewing angles are expressed depends on the type of 3D view currently selected.

\[
\begin{align*}
Z &= 30^\circ & X &= 60^\circ & \text{\(\#1\)} \\
Z &= 45^\circ & X &= 45^\circ & \text{\(\#2\)} \\
Z &= 120^\circ & X &= 20^\circ & \text{\(\#3\)} \\
Z &= 220^\circ & X &= 45^\circ & \text{\(\#4\)} \\
Z &= 60^\circ & X &= 30^\circ & \text{\(\#5\)}
\end{align*}
\]

This list of viewing angles appears when either an axonometric or a perspective view is displayed. The numbers represent the degrees of rotation. These rotations are for viewing purposes only and do not affect the actual position of the model in 3D space. Figure 3.5.1.2 shows the effect of each of the preset view angles.

**Custom View Angles**

This command is only available when axonometric or perspective views are displayed. It allows you to specify rotational values through the keyboard. When selected, it presents the **Custom View Angles** dialog, shown in Figure 3.5.1.3. This dialog contains three text fields representing the **Z**, **Y**, and **X Angles** of rotation. Rotational values can be typed in any or all the fields. Clicking on the **OK** button closes the dialog, and applies the customized rotational values to the view in the active window. The setting of the custom view angles is independent of the windows and the project files. These values are retained by **form-Z** from session to session, provided that the sessions are terminated by the **Quit / Exit** command.

Note that the **Custom View Angles** dialog can also be used as a query dialog to read the parameters of a view that may have been set using graphic methods for setting the view.

**Figure 3.5.1.2:** The preset axonometric/perspective views.

**Figure 3.5.1.3:** The **Custom View Angles** dialog.
These preset angles are available when **Isometric** is selected. Isometrics are not “natural” views but rather artificially constructed views that preserve the actual dimensions of objects. The angles in the preset views represent inclinations of the X and Y directions, relative to the view direction. The Z direction is always vertical.

Additional parameters, namely rotation about Z and scaling factors can be set in the **Isometric View Parameters** dialog (see below). The preset isometric views are always with the view direction aligned to the world x axis and scaling factor of 1.0. Figure 3.5.1.4 illustrates the preset isometric views.

- **Inclination = 30°**: 1
- **Inclination = 45°**: 2
- **Inclination = 60°**: 3
- **Inclination = 120°**: 4
- **Inclination = 135°**: 5
- **Inclination = 150°**: 6

These angles are available when **Oblique** is selected. Similarly to the isometric the oblique views are artificial views. The preset views have their X always parallel to the world x axis. The angles in the preset views represent the inclination of the Y direction from the view direction. Additional parameters can be set in the **Oblique View Parameters** dialog (see below). Figure 3.5.1.5 illustrates the preset oblique views.
3.5.2 Selecting an orthographic projection

The viewing position can be such that one of the orthogonal (Cartesian) planes of the 3D world space becomes coincident to the plane of the screen. This is equivalent to plotting an image, by ignoring one of the three dimensions, and is commonly called an orthographic projection. Orthographic projections can be relative to any of the three Cartesian planes or an arbitrary plane. They can view the plane from either its positive or negative side. The positive side of a projection orients the positive direction of the perpendicular axis towards the viewer. The reverse is true for the negative side. The projection commands automatically adjust the active reference plane to the projection plane, whenever necessary. This adjustment is reflected by the Reference Plane icon of the window tool palette of the active window (see subsection 2.2.1). A command is provided for viewing each of the orthogonal and arbitrary plane projections, from either side.

\[
\begin{align*}
[+XY] &: \text{Top} & [+YZ] &: \text{Right Side} & [+ZX] &: \text{Back} \\
[-XY] &: \text{Bottom} & [-YZ] &: \text{Left Side} & [-ZX] &: \text{Front}
\end{align*}
\]

These commands make the indicated orthogonal plane (XY, YZ, or ZX) coincident with the screen plane and display an orthographic projection of the modeling environment on the plane. In axonometric views, the images of the modeling objects appear as 2D shapes. In addition, the grid is extended to fill the contents of the window. This extension is temporary and only affects the grid while it is viewed in that projection. In a perspective view, the image reflects the three-dimensionality of the objects. Figure 3.5.2.1 shows the six different projection views.

*Figure 3.5.2.1:* Orthographic projections. Rows: (1) positive and (2) negative projections. Columns: (a) on XY (Top and Bottom), (b) on YZ (Right and Left), and (c) on ZX (Back and Front) plane.
**Plane Projection**

*form•Z* provides the tools for defining reference planes which may have any position and orientation in 3D space. They are called *arbitrary reference planes* and are very similar in nature to the Cartesian reference planes. The arbitrary planes are discussed in detail in subsection 2.2.3. The **Plane Projection** item pulls out a submenu that contains a complete set of orthographic projections. These are the same items available for the Cartesian planes and work the same way, but apply to the active arbitrary reference plane.

*Figure 3.5.2.2:* Hidden line arbitrary plane projection views: (a) 3D view of object and arbitrary plane, (b) \[+X] Top plane projection, and (c) \[-X] Bottom plane projection.
3.5.3 Types of 3D views and their parameters

With today’s display technology, the 3D world that we model can only be shown on the inherently two-dimensional surface of the monitor screen. The images on the screen that we frequently call three-dimensional, are only two-dimensional images that contain a visual illusion of the third dimension. This 2D image is derived from the 3D models by applying certain geometric transformations, which reduce the three coordinates of the model to two and make the resulting representation plottable on a 2D surface.

form•Z offers five methods for transforming the 3D representation of a model to a 2D image: **axonometric, isometric, oblique, perspective**, and **panoramic**. Four of these view types are illustrated in Figure 3.5.3.1. A panoramic view is shown in Figure 3.5.3.9. They are all produced by items in the **View** menu.

![Figure 3.5.3.1](image)

*Figure 3.5.3.1:* (a) Axonometric, (b) isometric, (c) perspective, and (d) oblique views.

When you click on one of the view items while you press **option** (Macintosh) or **ctrl+shift** (Windows), the **View Parameters** dialog, shown in Figure 3.5.3.2, is invoked. It mostly contains parameters shared by all the view types and also view type specific parameters, in its lower left portion. The content of this area changes with each view item. On the right side of the dialog are **Camera** options, which are special types of views, specifically useful for animations. The common options of views are discussed below:
Figure 3.5.3.2: The modeling **View Parameters** dialog.

**View Name**: A name for the current view is displayed in this field, which is the same name displayed in the Views palette. A default name is generated by the program and can be changed.

**View Type**: The current view type is displayed in this pop up menu and another one can be selected. When the dialog is invoked from one of the view items, that type of view is displayed in the menu. If it is changed, the selected view item in the **View** menu will also change. The latter can only be observed after you exit this dialog.

**Eye Point**: This is the position of the viewer’s eye, which may be expressed with either Cartesian or Polar coordinates (**XYZ** or **Polar**). Which type is used is determined by the item selected from the pop up menu next to it. Depending on this selection, the numeric fields of this option are adjusted to accept the appropriate parameter.

**Center Of Interest**: This is the point towards which the attention of the viewer is directed. It is again defined by its X, Y, and Z coordinates.

**Clip Hither/Yon**: This option turns on the hither and yon clipping planes, which are used to clip the view of objects when they are rendered using a rendering method other than quick paint.

**Hither** (close) and **Yon** (far): The values entered in these fields represent the distances of the respective planes from the eye point. These planes are always perpendicular to the sight line.
Camera View: When this option is selected, the view becomes a camera. Camera views are different than regular views in that they contain additional information about the dimensions of the picture plane. In regular views, the image resolution is determined by the window’s image options and the view is modified when the window dimensions are changed. The resolution of a camera is determined by the values set for its parameters, which simulate physical characteristics of real cameras. The image options and window size do not affect the camera views.

Two methods for setting cameras are available. One selects a camera from a list of standard cameras and its parameters are set automatically according to the selected standard. The other method allows you to individually set the parameters of a camera.

Standard: When this option is on, a camera is selected from its pop up menu, which contains a variety of standard settings, as shown in Figure 3.5.3.3.

![Standard pop up menu](image)

Figure 3.5.3.3: The Standard pop up menu.

Custom: When this option is on, the following parameters are set individually.

Width, Height: These values represent the number of pixels in the width and height of the image, respectively. The defaults are 640 and 480, respectively.

Aspect Ratio: This is the image width/height ratio, which is automatically updated when width or height is changed. When the lock (🔒) is selected, the ratio is fixed and changing Width or Height causes the other to be automatically updated.

Pixel Ratio: This is the width/height ratio of each pixel in the image. Default is 1.0.
**Axonometric**

This item is on by default, which is indicated by a check mark in front of it. An axonometric view is derived by essentially rotating the scene to a specific orientation and projecting it onto the screen.

Clicking on this item while pressing *option* (Macintosh) or *ctrl-shift* (Windows) invokes the View Parameters dialog (Figure 3.5.3.2). The type specific area displays the one option that affects the axonometric view (Figure 3.5.3.4).

**Spin:** This value, expressed in degrees, is the tilt at which a scene is viewed.

**Isometric**

Selecting this item produces an isometric 3D drawing. In contrast to the axonometric and perspective drawings, isometric drawings preserve the dimensions of the original model when measured in directions parallel to the orthogonal axes.

The isometric specific parameters are displayed when the View Parameters dialog is invoked from this item (Figure 3.5.3.5).

**Inclination Angles:** The values entered in the *X* and *Y* fields determine the angle by which the *X* and the *Y* orientations of a model will be inclined. Both angles are in degrees measured from the view direction. For angles less than **90°**, *X* is to the left, and *Y* is to the right. Angles greater than **90°** and negative angles are acceptable. Inclination angles can also be selected from the preset angles that appear in the top group of items of the View menu. These preset angles change to a set which applies to isometric drawings when *Isometric* is on. When inclination angles are selected from these preset items, the Eye Point, Center Of Interest and Scale values remain as they are currently set in the View Parameters dialog.

**Preserve Angles:** When this option is on (default), the values entered in the *X* and *Y* fields are locked to a total of **90°**. That is, whenever the value in one of the fields is changed, the other is automatically adjusted so that the sum of the values in the two fields is **90°**.

**Scale:** The values entered in the *X*, *Y*, and *Z* fields of this group of parameters represent scaling factors that can be applied to the respective directions. By default they are all set to **1.0**, which applies no scale. Typically, a scaling factor ranging between **0.50** and **0.90** is applied to the *Y* (or depth) of an isometric drawing, while the *X* and *Z* directions are not scaled.
Oblique*

This item produces an oblique 3D drawing. Like the isometric, the oblique drawings preserve the sizes of the original model when measured in directions parallel to the orthogonal axes. The **Oblique Parameters** are set in the **View Parameters** dialog (Figure 3.5.3.6).

**Scale**: This is a scaling factor by which the Y (depth) direction of a model is scaled.

**Inclination**: This value, which is in degrees, determines the inclination of the Y (depth) direction of the model relative to the view direction. For values less than 90°, the depth of the model appears to the right. For angles greater than 90° (and less than 180°), the depth appears to the left. For negative angles, the depth direction is toward the bottom of the screen, which produces a view of the bottom of the model. Observe that the X direction is the line perpendicular to the view line, and lines of the model that are parallel to the X direction will align with this direction. Inclination angles for the oblique drawing can also be selected from the **View** menu, when **Oblique** is selected. When a preset inclination parameter is selected from these menu items, the **Eye Point**, **Center Of Interest** and **Scale** parameters remain as in the **View Parameters** dialog.

Perspective*

When this item is selected a perspective transformation is applied to the coordinates of a modeling scene, before it is plotted or rendered. The perspective images are the most “realistic” and simulate the way physical entities look in the real world. The **Perspective Parameters** are displayed and set in the **View Parameters** dialog that is invoked from this item (Figure 3.5.3.7).

**Spin**: This is the tilt of the viewer, as for axonometric.

**Angle**: This is the angle of the cone of vision whose center is the line of sight. It controls the width of the field of vision and is defined as an angle in degrees.

**Focal Length**: This is the viewing area of a camera’s lens. It is the distance (in mm) of the focal point from the picture plane. Wide angle lenses have small focal lengths (i.e. 28) and zoom lenses have large focal lengths (i.e. 200). The focal length and viewing angle parameters are correlated with each other. That is, the larger the view angle the smaller the focal length, and vice versa.

**Keep Vertical Lines Straight**: Selecting this item produces a straight up perspective, also known as architectural perspective. This is a constructed 3D view that cancels the perspective convergence of the vertical lines and displays them always parallel to the vertical orientation. Note that this type of perspective always assumes that the XY plane is the ground plane.
**Panoramic**

When this item is selected, a panoramic view of the current scene is generated. This 3D view behaves differently than the other view types selectable from the View menu. When active, the icons in the tool palette are grayed out and cannot be selected. This is because the usual geometric mapping procedures cannot be made to work with a panoramic view. Consequently, the panoramic view is not an interactive environment but rather strictly a display mode. The program returns to its usual operations as soon as another 3D view type is selected.

A panorama can be thought of as a special view which is created by spinning the viewer around an axis. For example, imagine standing in the center of a room and turning around 360 degrees, recording the image along the way. In the real world, panoramic photographs can be taken with special cameras. The panoramic images generated by form•Z can be saved as regular images or they can be used for QuickTime VR panoramic movies (see section 3.6.10).

When invoking the **View Parameters** dialog from this item, the Panoramic Parameters shown in Figure 3.5.3.8 are displayed. Two of these parameters (Spin and Angle) are as for perspective views. The third is unique to this item.

**Smoothness**: The value in this field, expressed in pixels, determines the intervals at which the image strips that comprise a panoramic image will be generated.

![Figure 3.5.3.8](image)

**Figure 3.5.3.8**: The Panoramic Parameters in the **View Parameters** dialog.

Panoramic images are actually composed of small strips of individual 3D images, whose width is determined by the **Smoothness** parameter. Note that, in panoramic views, straight horizontal lines appear curved. If the smoothness parameter is set to a small value, such as 5 pixels, the curvature appears smooth. As the smoothness increases, curved lines appear more segmented. This is illustrated in Figure 3.5.3.9, where two panoramic views of the same scene are drawn with different smoothness: 5 and 50 pixels. Note how smoother the former is. Also note that the **Smoothness** parameter affects the rendering times.

![Figure 3.5.3.9](image)

**Figure 3.5.3.9**: The effect of the **Smoothness** parameter when generating panoramic images. **Smoothness** (a) = 5 pixels and (b) = 50 pixels.
3.5.4 View parameters and animations

View Parameters...
When this item is selected it invokes the **View Parameters** dialog, shown in Figure 3.5.3.2. Recall that this dialog is also invoked from **Axonometric**, **Isometric**, **Oblique**, **Perspective** or **Panoramic** in the **View** menu, as discussed in the previous subsection. The **View Parameters** dialog can also be invoked while the Cone of Vision environment is active.

Clip Hither/Yon
This item toggles on and off the **Clip Hither/Yon** option found in the **View Parameters** dialog. The preview area is redrawn to show the new clipping state. This item has a check mark in front of it when **Clip Hither/Yon** is on.

Reset Hither/Yon
This resets the positions of the hither and yon planes so that all the objects in front of the eye point are between these two planes. It does not adjust the viewer position. Any objects behind the viewer are ignored.

Reset View Angle
This resets the perspective view angle to the default 60°. The item is available only after the view angle has been changed. It is dimmed otherwise.

Reset View Spin
This resets the view spin of the cone of vision to a 0° spin. The item is available only after the view has been spined. It is dimmed otherwise.
Save View...

This item can be used to save the current view. When selected, it invokes the Name View dialog (Figure 3.5.4.1). To name the view, enter the desired name to replace the default name, and click on the OK button. A saved view is also listed in the Views palette. This is equivalent to saving a view in the palette.

Viewing parameters of the active window in both modeling and drafting can be saved. In modeling, the current zoom/pan location, the view type and its parameters are saved. In drafting, the current zoom/pan and view rotation are saved. There is no limit to how many views can be saved, which are saved with the project file and are automatically retrieved.

Views...

Selecting this item invokes the Views dialog (Figure 3.5.4.2), which is the same dialog invoked from the Views palette. It allows you to Save, Delete, Rename, make a view Active, move to Top and Bottom, Sort, and completely Clear the views list. These buttons work as discussed in subsection 2.3.3. The Views dialog is discussed in subsection 2.9.2.

Load Project Views...: When you click on this button, the standard Open File dialog is presented, where you can select the name of the project whose views you wish to transfer to the active project. The views are transferred as soon as you click on Open in the Open dialog.
3.5.5 Setting sun positions

Sun Position...

This item allows you to define the position of the sun light either by specifying the sun’s altitude and azimuth, or by selecting the geographic position of the site and the month, day, and time of the desired sun position. The sun position is used by the Quick Paint* and Surface Render* modes to calculate the shading and, if the shadows option is selected, the shadows for rendering. These modes are affected by the sun light only, even when more lights are defined in the Lights palette. For interactive shaded and the higher end rendering modes (Shaded Render* and RenderZone*) the sun is as any other distant light.

This item invokes the Sun dialog (Figure 3.5.5.1) for setting sun parameters.

![Sun dialog](image)

Angle of Site North: This field is used to align the project’s coordinate system with Earth’s compass directions. By convention, the XY plane is aligned with the Earth’s surface and the positive Z axis projects from the Earth’s surface into the sky. The Angle of Site North field determines the orientation of North on the XY plane. This is the direction of a North arrow typically drawn on architectural plans or maps.

By Site, Date And Time Zone: This group of options allows you to set the sun position by picking a site, a date and a time zone.

Site From City: This field tells you what geographical location is currently selected. Another geographical location can be selected from the Geographic Position dialog, invoked from the Choose Site... button (see next paragraph). Three information fields under the city name display the latitude, longitude, and time zone of the selected location. The time zone is the time it would be at the location when it is 12:00 noon at Greenwich Mean Time (GMT), as expressed on a 24-hour clock. For example, a location 2 hours ahead (east) of GMT would be 14:00 and a location 4 hours behind (west of) GMT would be 8:00. The time zone information is needed because the time zone divisions do not strictly follow the 15° longitudinal divisions.
**Choose Site...**: This button invokes the **Geographic Position** dialog (Figure 3.5.5.3) that allows you to select a site location. This dialog is discussed towards the end of this section.

**Site From Longitude, Latitude And Time**: The options in this group offer an alternative for defining a geographical location, which is through its **Longitude**, **Latitude**, and **Time Zone**. Entering values in these numeric fields defines the position.

**Date And Time**: These fields are used to specify the **Month**, **Day Of Month**, and **Time**. The month and the day are selected from pop up menus (see Figure 3.5.5.2). The time is entered by typing it in together with AM or PM. There is also a check box for indicating that a one hour **Daylight Savings** adjustment is in effect. You can also set the date to the seasonal peaks, by selecting the **Summer/Winter Solstice** or the **Vernal/Autumnal Equinox**, or to the **Current Time** and date. The latter uses your computer’s clock and, to work properly, your computer’s clock should be set correctly.

*Figure 3.5.5.2: The Month and Day menus.*
By Altitude And Azimuth: This offers an alternative method for defining a geographical location. It contains two angle fields, Sun Altitude and Sun Azimuth, and a pop up menu. Altitude is the angle of the sun with the horizon. Azimuth is the angle of (the projection of) the sun from the east-west axis. The values entered in these fields define a geographical location, provided they are supplemented by the selection from the pop up menu.

From: This pop up menu allows you to select North or South. By convention, the 0° azimuth is due south from the northern hemisphere and due north from the southern hemisphere. The From pop up menu allows you to select whether the azimuth is measured from north or south. Sites north of the Tropic of Cancer (23.5° N.L.) will always measure the azimuth from the south and sites south of the Tropic of Capricorn (23.5 S.L.) will always measure from the north. Sites between the tropics will measure from north or south depending on the day of the year.

The sun position set through the Sun dialog will update the position of the light that is designated as the sun. When the sun position is moved the values in the Altitude and Azimuth fields will reflect that position, next time the Sun dialog is invoked. The Sun dialog is also available from within the cone of vision environment.

Selecting and editing geographical locations

To select a geographical location click on the Choose Site... button in the Sun dialog. This will invoke the Geographic Positions dialog (Figure 3.5.5.3), which displays the form-Z library of geographic locations. The dialog also allows you to edit, add locations to, and delete locations from the library. The form-Z library of locations contains over 500 cities, which includes all the capitals of the world and all the cities with population over 400,000. The library is stored in a file named “form-Z Geography,” and should be in the same folder with the form-Z application.

List By: In the Geographic Positions dialog, the cities are listed in alphabetical order. Under List By, you have the option to list them by City name or by Country name.

Figure 3.5.5.3: The Geographic Position dialog.
You select a city by clicking on its name. Typing the first letter of a city/country will scroll the list to the first city/country that starts with that letter. When you select a city, its name appears in the City field and the Latitude, Longitude, and Time Zone fields are updated to show the geographic position of the city.

Add...: This button invokes the Add City dialog (Figure 3.5.5.4). At its top is the Country field where the name of a country or the word <None> appears, when no country is currently selected. To select a country, click on the Set... button, which invokes the Country dialog, shown in Figure 3.5.5.5. This dialog contains a list of countries that can be selected and includes the New... button, which allows you to add new countries to the list. Clicking on it invokes the Name Of New Country dialog (Figure 3.5.5.6). Enter the name of the new country and click OK.

To complete the entry of a city, enter a name in the City field and the proper values in the Latitude, Longitude, and Time Zone fields. As soon as you click OK, the system adds the city you created to the geographic locations library and makes it the current city.

Edit...: This button allows you to change the location information. It invokes the Edit City dialog (Figure 3.5.5.7) which is identical to the Add City dialog except for its name.

Delete: This button is used to delete a city. Select a city and then click on the Delete button. Before deleting the city, the system will issue a warning alert.
3.5.6 The cone of vision

The **cone of vision** provides graphic control over all the parameters that affect the view of the modeling environment. They are the **viewer position**, **center of interest**, **line of sight**, **view spin**, and the **hither** and **yon** clipping planes. Axonometric, isometric and oblique views also use the **view extents** and perspective and panoramic views use the **view angle**.

The cone of vision provides an accurate graphic representation of the viewing parameters, which can be interactively edited. The cone of vision has three different graphic representations based on the type of view being edited. Axonometric, isometric and oblique views use a cone that looks like a rectangular cube which reflects the parallel nature of their projections. The perspective cone appears like a truncated pyramid reflecting its convergent characteristic. The panoramic cone is cylindrical. The three cone of vision types are shown in Figure 3.5.6.1 with their parts labeled.

When the cone of vision is invoked from a perspective view to which zooming has been applied, two cones are displayed: one corresponds to the normal (100%) view and the other to the zoomed view.

**Figure 3.5.6.1:** The cone of vision and its parts for: (a) axonometric, isometric and oblique, (b) perspective, and (c) panoramic.
**Edit Cone of Vision**

When this item is selected, the system opens the **Cone of Vision** window, shown in Figure 3.5.6.2. This is a special type of a graphics window that is subdivided to four areas, each occupying one quarter of the screen. Three of the four areas function as associated windows and each displays a different orthographic projection of the modeling environment. These views are the same as the default projections of the tile windows (subsection 3.3.1). The upper left portion is a *top*, the lower left a *front*, and the lower right a *right* view. These views display the modeling scene as well as the cone of vision, which is shown in black. The parts of the cone of vision can be moved in any of the three orthographic projections.

The upper right area displays the 3D view defined by the current viewing parameters of the cone of vision and is referred to as the **preview area**. It is marked with a frame in the highlight color that indicates the border of the active window. Any part of the image that is beyond the frame is beyond the window limits and is not visible. The **Cone of Vision** window and its subdivisions are shown in Figure 3.5.6.2. This window functions like a dialog; it remains the only active window until it is closed.

*Figure 3.5.6.2: The Cone of Vision window for a perspective view.*
When the Cone of Vision window is first opened the upper left projection is active. Any other projection, except for the preview area, may be activated by clicking on it. The active area is marked by a three-pixel-wide, black rectangular frame. The Cone of Vision window has most of the features of a common window, but it cannot be resized or moved. In addition, the controls that are available affect only the active area and not the complete window or the preview area. The close box in the title bar of the window is the only control that affects the entire window. Clicking on it closes the Cone of Vision window, and returns control to the graphics window that was active when the Cone of Vision window was opened. The view shown in the preview area becomes the current view in the active window.

Most of the window tools are available in the Cone of Vision window but, like the other window controls, they affect only the active area and not the complete window. They have no affect on the preview area either. The window tools are updated each time another area is activated, and they always reflect the status of the graphic parameters of the active area. A few of the window tools can not be used in the cone of vision environment; they are the Reference Plane tools, the Perpendicular switch, and the View tools. This is because the views in the projection areas are fixed and can not be changed. These icons are inactive and are grayed out. The remaining tools in the window tool palette can be used freely. Also, many of the general menu bar commands remain available and can be used while the Cone of Vision window is active.

All the parts of the cone of vision can be moved in the active projection. Moving parts of the cone causes one or more of the viewing parameters to change. To move a part of the cone, the mouse is first clicked on the desired part. This picks the part, and the affected portions of the cone are rubber banded and follow the motion of the mouse. As this occurs, the image in the preview area is constantly updated, allowing you to visually inspect the result of the movement. Options are available that control whether the image is shown in detail or whether only bounding boxes are drawn. These options affect the redraw speed. The image regeneration speed is affected by the speed at which the mouse is moved and by the complexity of the model. The system will not update the complete image when the mouse is moved quickly. When the system detects movement in the mouse, it stops the drawing process and starts redrawing the new image. The objects of the model are drawn in the order in which they were generated. Keeping the mouse idle for a few seconds allows the system sufficient time to completely update the image.

When you are satisfied with the new view, one more click of the mouse freezes the view at its current state. The process of editing the view may continue by picking the same or another part of the cone of vision, in the same or another area. Picking a part of the cone in another area first activates that area, then picks the intended element, and then begins to move it.

The system recognizes which part of the cone of vision the user intends to pick from the position of the mouse relative to the pickable parts of the cone. No mode or modifier is required to be set. The movable parts and how they are picked are shown in Figure 3.5.6.3 and discussed below.
(a) The **viewer position** is picked by clicking in the small circle that represents eye of the viewer. The viewer position is connected to the center of interest and the two together define the line of sight (see below). When the viewer position is moved, the line of sight is anchored at the center of interest. The viewer position may be moved along the length of the line of sight, or it may be rotated around the center of interest, which affects the direction of the line of sight. The complete cone of vision follows the motion of the viewer position. This is due to the relationship between the viewer position and the parameters of the cone. The line of sight is always the central axis of the cone of vision and is perpendicular to the yon and hither planes at their centers.

(b) The **center of interest** is picked by clicking the mouse on the small cross that represents it. When it is moved, the line of vision remains anchored at the viewer position. It may move along the direction of the line of sight, or it may rotate about the viewer position. Its movement affects the complete cone of vision in the same fashion as the movement of the viewer position.

(c) The **line of sight** is the line from the viewer position to the center of interest. It is selected by clicking the mouse on or close to it. When the line of sight is moved, both the viewer position and the center of interest move together. Thus, the line of vision is always moved to positions that are parallel to its original position, relative to the projection of the active area. The motions of the line of sight affect the complete cone of vision.

The movement of the viewer position (a), center of interest (b), and the line of sight (c) can be constrained to move along the line of sight by pressing the **shift** key while moving the mouse. This makes it easier to move the viewer position and/or center of interest closer or farther to the scene, with out changing the view direction.
(d) The **view extent** or **view angle** is picked by clicking the mouse on one of the side edges of the cone of vision and behaves differently for each type of cone. In the cone for axonometric, isometric and oblique, all four edges move parallel. This motion increases or decreases the view extent. Reducing the view extent results in a larger image and increasing results in a smaller image. In the perspective cone, all four edges move together symmetrically, following the motion of the selected line. This motion increases or decreases the view angle. Reducing the view angle results in a larger image and increasing the view angle results in a smaller image. In the panoramic cone, the view boundaries are dynamically rotated around the axis of the panoramic cylinder. Note that changing the view angle also changes the height of the cylinder. Recall, that the circumference of the cylinder represents the width of a panoramic image while the height of the cylinder represents the height of the image. As the view angle changes, so does the width of the image, and the height of the cylinder needs to be adjusted so that the proportions of the width and height of the image may be maintained.

(e) The **hither plane** is picked by clicking the mouse on a segment of the top face of the cone, which represents the position of the hither plane. The hither plane moves only along the direction of the line of sight and cannot be moved beyond the current position of the yon plane or beyond the eye position. The hither plane represents the nearest visible point along the line of sight. When the **Clip Hither/Yon** option is selected, the rendered and hidden line images are clipped to this plane. Objects or portions of objects that are in front of this plane are invisible when clipped.

(f) Similarly, the **yon plane** is picked by clicking the mouse on a segment of the bottom face of the cone, which represents the position of the yon plane. The yon plane moves only along the direction of the line of sight, and cannot be moved closer to the viewer position than the hither plane. The yon plane represents the farthest visible point along the line of sight. When the **Clip Hither/Yon** option is selected the rendered and hidden line images are clipped to this plane. Objects or portions of objects that are behind this plane are invisible when clipped.

(g) The **view spin** is changed by clicking the mouse on any one of the corner points of the yon plane. When picked, the cone is dynamically rotated around the line of sight. This has the effect of tilting the viewer’s head and looking at the modeling environment sideways. The wide black line on one of the edges of the hither plane represents the bottom of the image on the screen. This line is parallel to the horizon of the image and can be used to visually orient the view spin.

The project’s lights are also shown and can be manipulated graphically through the **Cone of Vision** window. The lights which have their visibility attribute on are shown in the projection areas. The lights can be graphically edited in the same manner as in a regular project window.

As already mentioned, many of the menu bar commands remain available to the **Cone Of Vision** environment and can be used, with their action being frequently adjusted to the semantics if the cone of vision. For example, the **Undo** and **Redo** items affect the view manipulations that have occurred within the cone of vision environment. All the display and the view type setting commands affect the 3D preview area only and none of the three projection areas. The **Align And Scale Frames** can be used to align and scale the views of the cone of vision three projection areas.
3.5.7 The drafting view menu

When a drafting window is active, the view menu appears as shown in Figure 3.5.0.1 (b). It has one group of items at the top containing default rotations and the View Parameters... item in the fourth group for editing drafting view parameters. The remaining items in the menu are exclusively used in modeling and are dimmed and inactive when a drafting window is active.

Each drafting window has its own view parameters as does each drafting pane. In drafting space, the view parameters are displayed and applied to the drafting window. In layout space, if a pane is active, the view parameters are displayed and applied to the active pane. If a pane is not active in layout space, the view parameters are displayed and applied to the layout space window.

A view in drafting consists of the view center, view rotation and view scale. The center is the coordinates of the draft or paper space that are positioned in the center of the view. The rotation is the angle that the view is rotated around the view center. The view scale determines how large or small the drafting elements appear in the view.

<table>
<thead>
<tr>
<th>Rotation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>30°</td>
<td>1</td>
</tr>
<tr>
<td>45°</td>
<td>2</td>
</tr>
<tr>
<td>60°</td>
<td>3</td>
</tr>
<tr>
<td>180°</td>
<td>4</td>
</tr>
<tr>
<td>270°</td>
<td>5</td>
</tr>
</tbody>
</table>

When selected, these items change the rotation in the active window or pane to the rotation specified in the menu item. The view center and view scale are not changed. The drafting window or pane is redrawn to reflect the new rotation. These are illustrated in Figure 3.5.7.1

Figure 3.5.7.1: Views with Rotation = (a) 0°, (b) 45°, (c) 180°, (d) 270°, (e) 90°, (f) −30°.
**Custom Rotation**

This menu item invokes the **Custom View Rotation** dialog as shown in Figure 3.5.7.2. The **Rotation** field displays the current rotation and can be changed to any desired angle. When the dialog is closed, the angle is changed and the drafting window or pane is redrawn to reflect the custom rotation.

**View Parameters**

This menu item invokes the drafting **View Parameters** dialog as shown in Figure 3.5.7.3. When the dialog is closed, the new parameters are applied to the drafting window or pane and it is redrawn to reflect the new parameters.

**View Name:** This is the name of the current window which can not be edited. When the **View Parameters** dialog is invoked from the **View** dialog or palette, this field contains the name of the saved view and it can be edited.

**Center:** This group of items contain the positions of the X and Y coordinates of the center of the view. The coordinates specified in these fields will be placed at the center of the view’s window or pane.

**Rotation:** This field specifies the view’s rotation about the view center and can be changed to any desired angle.

**Scale:** This field specifies the scale of the view. The larger the scale, the larger the image of the drafting elements will appear in the view’s window or pane. This scale works the same as the scale in the **Display Menu** as described in section 3.6.
3.6 The Display menu

The modeling Display menu (Figure 3.6.0.1(a)) consists of ten groups of items. The drafting Display menu (Figure 3.6.0.1(b)) consists of four groups of items.

- In both modeling and drafting, the first group contains items for displaying and changing the scale of the active window.
- The second group contains two interactive display items.
- The third group contains a variety of rendering types.

Modeling:

- The fourth group contains rendering types which are installed as plugins.
- The fifth group contains two RenderZone related items.
- The sixth group contains four Radiosity items. The items in the fifth and sixth groups are only available in form•Z RenderZone Plus.

![Display menus](image)

*Figure 3.6.0.1: The Display menus for (a) drafting and (b) modeling.*
All the rendering items in groups 2 through 5 invoke **Options** dialogs when you press **option** (Macintosh) or **ctrl+shift** (Windows) while clicking on the item.

- The sixth group contains two network rendering items.
- The seventh group contains two items that invoke **Options** dialogs that apply to displays and renderings.
- The eighth group contains items that produce and play animations,
- The ninth group contains items that manage the rendering memory and the appearance of surface shapes.
- In both modeling and drafting, the last group contains a single item that invokes the **Image Options** dialog.

**Drafting:**

- The second group contains the single plotting command available in drafting and an item for invoking the **Display Options** dialog.
- The third group in drafting contains the single **Draft Layout Mode** item.
- The fourth group contains the **Image Options** item.

**form·Z** offers many methods for displaying the images of 3D models. **form·Z RenderZone Plus** includes one more. All methods are affected by optional parameters that are set through dialogs. Generally, the speed and the quality of a display mode are inversely analogous.

When a display command is selected, it is executed immediately. The contents of the active window are re-displayed using the selected method of plotting or rendering. The display commands also work as modifiers and, when selected, they set the **active display mode**, accordingly. This is the mode that is used every time the window or an area of the window is redrawn. Since the effects of the operations are immediately shown on the active window, the speed of the different display methods is a major factor in deciding which display method to use. The default mode is the wire frame display that is set by the **Wire Frame** command.

**form·Z** uses a **rendering record** and a **display buffer** for storing information relative to plotting and rendering. The rendering record contains sorting and splitting information that is derived for a modeling scene and is used by all the plotting/rendering methods except wire frame and quick paint. The display buffer is essentially a copy of the pixel map used to display an image on the screen. The display buffer is used with hidden line and scan line rendering. **form·Z** makes a special effort to avoid the recalculation of the screen image. When a dialog is opened or a palette is moved either the display buffer is simply redrawn on the screen or the information in the rendering record is used to produce the image, without recalculating it, whenever possible.
3.6.1 Setting the display scale

When new projects are opened, the scale of the initial window defaults to 1/16"=1'-0" (or 1:200 for metric). When a new associated window is opened, its scale is set to the scale of the active window. Associated windows may have different scales.

There are two methods for changing the scale of a window. The first method produces a scale that depends on the current scale, which is either doubled or halved. The second method allows you to define any scale by entering a scaling factor in a dialog.

- 1/8" = 1'-0" Scale
- √ 1/16" = 1'-0" Scale
- 1/32" = 1'-0" Scale

The top three lines of the Display menu work as a group that consists of two commands and an information field. The second line, which is always displayed with a check mark in front of it, is not an executable command but rather displays the scale of the active window. The first and the third lines are commands that can be used to change the scale of the active window. They offer a frequently convenient but limited method for changing scales.

The first item always displays a scale that is twice as large as the current scale. The third item displays a scale which is half of the current scale. When either one is selected that scale becomes the current scale of the active window. The scaling factors shown in these three items are updated to reflect the change in scale. These items provide the ability to make quick scale changes by either doubling or halving the current window display scale.

**Custom Display Scale...**

This item represents a general method for changing the scale of the active window. When selected, it invokes the Custom Display Scale dialog shown in Figure 3.6.1.1. It consists of a single field where a scaling factor can be entered. Clicking on OK closes the dialog and applies the new scale.

When the scale of the window is changed, the new scaling factors are immediately reflected in the top three lines of the Display menu. Thus the values of the scale commands constantly change as scaling changes are made. The values shown above are based on the default scale.

In addition to the display scale, the current zoom factor determines the final size of the entities as they are displayed. This zoom factor is shown in a pop up menu at the bottom of the window. At 100%, the display scale is applied to the real world size of the entities before they are drawn. The zoom factor applies an additional scaling to this process. The zoom factor can be changed in a number of different ways. It can be chosen from the pop up menu. It also can be changed by executing any of the Zoom, Fit, and Reset window tools.
3.6.2 Interactive displays

The second group of commands in the Display menu produce interactive displays. Currently there are two such commands, Wire Frame and Interactive Shaded, which is based on OpenGL. More interactive rendering methods are expected to become available through plugins in the future. Interactive display modes are real time renderings that allow you to view and work on a model.

Wire frame is the default interactive display. Both interactive displays are controlled by options that make it possible to show a variety of additional information about objects. These are selected from their dialog boxes, which typically consist of two tabs: the content of one tab is specific to the display method and the other contains options that are shared by all the interactive display modes.

**Wire Frame**

Clicking on this item displays a wire frame line drawing in the active window. A wire frame shows all the lines or edges of objects. Since no effort is made to eliminate the lines that are behind surfaces, it is the fastest display method. Examples of wire frame plots are shown in Figure 3.6.2.2.

Pressing option (Macintosh) or ctrl+shift (Windows) when clicking on **Wire Frame** invokes the Wire Frame Options dialog (Figure 3.6.2.1) whose options affect the wire frame plots. It consists of two tabs, Wire Frame and Interactive. They are both shown in Figure 3.6.2.1.

![Wire Frame Options dialog](image)

**Figure 3.6.2.1:** The Wire Frame Options dialog with (a) the Wire Frame and (b) the Interactive tabs selected.
**Wire Frame tab**

**Show Back Faces**: When this option is on, the back facing faces of objects are plotted or rendered. They are not if this option is off. An example is shown in Figure 3.6.2.2(b).

**Show Color**: When selected, all objects are shown with the color that is assigned to them. Faces that may have been assigned different colors are also shown in the color assigned at the object level. When off, all objects are plotted in black when the background color is light, or in white when the background is dark. The default is on.

**Hide Ghosted**: When this option is on, ghosted entities are not shown. They are shown in the project’s ghost color (grey by default), when this option is off. This option can be applied to either **Objects** or **Layers** or both, depending on the selection from the two options under it.

**Smooth Objects**: The two options in this group control whether the facets, the wires, or both are displayed for parametric objects (analytic primitives, nurbz, and patches).

**Facets**: When this option is on, facets are also shown for smooth objects, in addition to their iso lines, as illustrated in Figure 3.6.2.3. When shown, the color intensity of facets can be set using the slide bar next to this option. By default, the intensity of facets is lighter than that of the iso lines.

**Iso Lines**: This slider bar is used to set the intensity of the color of the iso lines.

Note that, while Facets can be turned on/off, the Iso lines are always on. However, their color intensity can be set low enough to make them virtually invisible.

**Radiosity Display**: These two options apply to the previewing of radiosity solutions, when **Wire Frame** is the active display mode (for more details see section 6.8 of the **form-Z RenderZone Plus** User’s Manual). This group of options is only present in the **RenderZone Plus** version of **form-Z**.
**Draw Mesh:** When this option is on, the radiosity mesh is displayed with the preview.

**Draw Mesh Ghosted:** When this option is on, the radiosity mesh is displayed in the project’s ghost color, while the original objects are drawn in their colors.

**Renderer:** This menu controls what low-level renderer is used to perform the **Wire Frame** rendering. Since the **Wire Frame** renderer itself supports plugins, future plugins could present other options in this menu. This menu has two items by default, as shown in Figure 3.6.2.4.

**Native Graphics:** When this item is selected, the **Wire Frame** renderer uses the system’s native graphics capabilities in conjunction with display buffers. When this item is selected, its options, under the title **Buffer Redraw**, are also displayed. They control how the screen is updated, as follows:

- **Automatic:** When this option is on, the system determines how frequently the screen is updated from the buffer, using a variety of criteria.

- **Interval:** When this option is selected, the system uses the interval typed in its field to determine how frequently the screen is updated.

**OpenGL:** When this item is selected, the **Wire Frame** render uses the system’s OpenGL graphics capabilities. If OpenGL is not installed, this item is dimmed. When the **OpenGL** item is selected, the content of the **Renderer** box changes to a single button labeled **OpenGL Info...**, as shown in Figure 3.6.2.5.

OpenGL is faster than **Native Graphics** when a capable hardware accelerator display card and driver are installed on your computer. **OpenGL** performance will deteriorate faster than **Native Graphics** as the number of edges in the scene increases. **OpenGL** performance will vary depending on the graphics card’s capabilities and memory. **OpenGL** can not represent perspective views with the **Keep Vertical Lines Straight** option or panoramic views. When either of these are active, **OpenGL** can not be selected.

**OpenGL Info...:** Clicking on this button, displays the **OpenGL Info** dialog (Figure 3.6.2.6), which lists information about your OpenGL installation. This dialog has two text information fields.

- **OpenGL Version:** This is the version of OpenGL installed on your machine. This is a part of the operating system and is usually installed by default. Hardware vendors may supply special versions of OpenGL for their specific needs.

- **OpenGL Vendor:** This is the maker of the OpenGL hardware and/or driver installed on your machine.

---

*Figure 3.6.2.4:* The **Renderer** menu.

*Figure 3.6.2.5:* The **Renderer** box with the **OpenGL** item selected.

*Figure 3.6.2.6:* The **OpenGL Info** dialog.
**Interactive** tab

**Show Points**: When this option is on, the points (vertices) of the displayed objects will be marked with a diamond. An example is shown in Figure 3.6.2.7.

**Show Key Points**: When this option is on, the key points for the object curve are shown as small dots on the screen. See section 2.2.6 in this Manual).

**Show Marked Points**: The Set/Clear Point Marker tool can be used to mark points by clicking on them, as discussed in subsection 4.21.8. These marked points are used by the Skin tool to place source shapes on paths. When this option is selected, these marked points are displayed with a diamond around them. This option is off by default.

**Show Objects As Bounding Volumes**: When this option is selected, the objects appearing in the modeling window are displayed as bounding boxes. When this option is off, which is the default, the faces of the object are plotted (Figure 3.6.2.8). Bounding volumes display much faster, especially when used for high resolution topology.

**Figure 3.6.2.7**: Plotting wire frames with (a) Show Points on and (b) Show Marked Points on.

**Figure 3.6.2.8**: Objects shown (a) without and (b) with Show Objects As Bounding Volume on.
**Show Face Normals**: The normals are vectors perpendicular to a surface. When this option is on, the normal vectors of all faces are also displayed, as illustrated in Figure 3.6.2.9. This option is off by default.

**Use Face Color**: When this option is on, the normal vectors are displayed in the color of their face. When this option is off, they are displayed in the object color.

**Size n Pixels**: The value entered in this field determines the size of the normal vectors in screen units (pixels).

Note that, when objects have dense faces, the normals may be hard to read. In such cases displaying the objects with their back faces eliminated improves their readability, as shown in Figure 3.6.2.9(b) for the sphere. Another method for improving the readability of the normal vectors is to select individual faces. When faces are selected, their normals are also highlighted, which makes it easier to see which normal corresponds to the selected face.

**2D Surface Objects**: These two options control whether certain structural details (first point and direction) will also be marked when surface objects are plotted.

**Show First Point**: When this option is selected, the first points of surface objects are marked with a diamond shape when plotted. An example is shown in Figure 3.6.2.10(a).

**Show Direction**: When selected, the directions of surface objects are shown by displaying arrows placed at the middle of each segment. With consecutive small segments, arrows are placed at intervals (Figure 3.6.2.10(b)). These options are quite useful for operations that are sensitive to directions.
Objects Axes And Centroids: This group of options specifies whether or not the local axes and centroids of objects will be shown and how. The local axes and centroids are generated at default positions at the time objects are created. Their positions and orientation follow the object as it is transformed. These can also be changed through the Objects dialog (invoked from the Objects... item in the Options menu, see subsection 3.7.2) or by using the Query tool.

Show Object Axes: When this option is on, the axes of each object in the modeling scene are also displayed (Figure 3.6.2.11(a)). When they are displayed, point snapping to the axis origin is also available.

Show Centroid: When this option is on, the centroid of each object is displayed as a 3D cross (Figure 3.6.2.11(b)). The centroid of an object can be set in three different ways, in the Objects dialog. When displayed, point snapping to it is also available.

Size n Pixels: The value n entered in this field specifies the length of the local axes to be displayed. It is expressed in pixels (default is 30).

Lights: The options in this group control how lights are displayed, when they are visible, which is set in the Lights palette.

Show Light Color: When this option is selected, each light is drawn in the light’s color. For all lights, the light direction vector is drawn in the color, whereas the light’s outline, such as the inner and outer cones of cone lights or the sphere of point lights, are drawn in black. For area lights, the arrow of the light direction, originating from the center of each face, is drawn in the face color, whereas the edges of the area light object are drawn in black. If this option is off, which is the default, the lights are drawn in black.

Show Direction Of Area Lights: When this option is on, arrows are drawn from the center of each face of an area light. Otherwise they are omitted. Note that area lights and this option are only available in form-Z RenderZone Plus, where the option is on by default.

Cameras: The options in this group control the size and the visibility of the viewing cones, when views are visible, which is set in the Views palette.

Camera Size: The value entered in this field, which is in pixels, determines the size of the camera graphics drawn on the screen. The default is 30 pixels.

Show View/Camera Cones: When this option is on, the camera view cones are displayed, as shown in Figure 3.6.2.12(a).
Plain Text Objects: Recall that 3D text can be generated as plain text or text objects. The latter are treated as all the other objects when plotted or rendered. The options in this group affect the plotting of plain text only.

Outline: The items in this pull out menu control how the outlines of text characters will be shown (Figure 3.6.2.13(a)). When Off is selected, no outline is plotted. When Black/White is selected the text outline is shown black on a light colored background and white on a dark colored background. When Color is selected (default), the outline is drawn in the text’s color.

Fill: These items control how the text characters will be filled (Figure 3.6.2.13(b)). Selecting Off, Black/White, or Color results in no fill, black or white fill, depending on the background color, or a fill in the text’s color. The default is Off.

The next three mutually exclusive options control how plain text is displayed.

Use Text Object’s Smoothness: When this option is on (default), text is displayed at the resolution it was generated with.

Bounding Box: When this option is on, text is displayed as bounding rectangles. This is useful for achieving fast redraw speeds during project development, when a lot of text is placed.

Override Text Smoothness: With this option on, the internal text resolution is ignored and the text is displayed with a locally defined resolution, which is set by the Outline Smoothness slider under this option. Again, this useful for achieving quicker redraw speeds during project development.

Symbol Instance Objects: This option controls how 3D symbols are displayed in wire frame.

Show As Bounding Volume: When this option is selected, the 3D symbols are represented by a 3D volume bounding cuboid, which allows faster regeneration of the screen. By default, this option is off and the 3D symbols are plotted with their complete detail.

All Windows: The plotting/rendering parameters are window level options (as opposed to project level options). Their settings affect only the active window, and different windows that may be open at the same time may have different plotting/rendering settings. This option allows you to treat the plotting/rendering parameters as project level parameters. When this option is selected, the current settings of the plotting/rendering options will be applied to all the windows of a project. This option is off by default.
**Interactive Shaded***

Clicking on this item displays an OpenGL based shaded rendering in the active window. Clicking on this item while pressing *option* (Macintosh) or *ctrl+shift* (Windows) invokes the **Interactive Shaded Options** dialog (Figure 3.6.2.14). It contains two tabs: **Shaded** and **Interactive**. The latter tab is as for **Wire Frame***. The **Shaded** tab contains the following options:

**Show Edges:** When this option is on, the edges of the faces are also visible.

**Show Object Color:** This option is available only when the edges are shown. When on, the edges are assigned the color of the object. Otherwise they are black on a light background and white on a dark background.

**Shadows:** When this option is selected, the interactive shaded renderings are displayed with shadows for shining **distant, point and cone** lights with shadow casting enabled. Interactive shaded rendering only supports hard shadows. All shadows are rendered as hard shadows regardless of the setting for the shadow type in the **Light Parameters** dialog. Figure 3.6.2.15 shows an example of an interactive image without and with shadows. Renderings with shadows are generally slower, however, the ultimate performance depends on the capabilities of the computer’s graphics card.

![Figure 3.6.2.14: The Interactive Shaded Options dialog.](image)

*Figure 3.6.2.14: The Interactive Shaded Options dialog.*

![Figure 3.6.2.15: Interactive Shaded rendering (a) without and (b) with shadows.](image)

*Figure 3.6.2.15: Interactive Shaded rendering (a) without and (b) with shadows.*
**Accurate:** When this option is off, all shadows are shown with a constant intensity based on the ambient light. When this option is enabled, the shadows are displayed with accurate intensities. That is, the shadows from each light source are affected by other lights in the scene and the distinction between shadows is often visible. Figure 3.6.2.15 illustrates the effect of this option. This option is slower but yields a more accurate result. This option is off by default.

![Interactive Shaded rendering with shadows (a) without and (b) with the accurate option.](image)

**Figure 3.6.2.15:** Interactive Shaded rendering with shadows (a) without and (b) with the accurate option.

**Transparencies:** When this option is selected, objects can be rendered transparent. Objects with surface styles that use the Simple transparency will be rendered with the transparency of the surface style. If the Textures option below is enabled (RenderZone Plus version) and a transparency shader is used in the surface style, then the transparent texture is displayed. This option is off by default.

**Illuminate Back Side Of Surfaces:** When this option is selected, faces of surface objects that face away from light sources are illuminated as if they were facing the light (i.e. reversed). Thus the face directions of surface objects do not affect the rendered image. This is consistent with Shaded render and RenderZone renderings. This option is on by default.

**Hide Ghosted:** As for Wire Frame.

**Textures:** When this option is on, Interactive Shaded displays the textures for objects that use surface styles that include image based or procedural textures. The texture is determined by the color component of the surface style. If the surface style contains a texture in its transparency component, this texture will be included if the above Transparencies option is selected. If the surface style contains a texture in its bump component, this texture will be included if the below Bump Mapping option is selected. This option, only available in RenderZone Plus, is off by default.
Image Textures n Pixels: This menu determines the maximum desired size for image based textures. If the maximum dimension of an image file (used by a surface style's color map, transparency map or bump map) exceeds this size the texture is reduced to the specified dimension. If the maximum dimension of an image file is less than the desired size, the dimensions of the file are used for the texture. Note that textures are rendered in OpenGL based on square power of 2 dimensions (16,32,64,128, etc). Textures generated from image files are rounded to the nearest power of 2 dimension. Image files with a power of 2 dimension yield the best image quality and performance.

Procedural Textures n Pixels: The value entered in this field determines the resolution level of for procedural textures. Note that procedural textures by their nature can only be accurately rendered in the original renderer (RenderZone) for example. The Interactive Shaded shows the procedural texture by rendering a small tile of the procedural pattern and then repeating it across the surface. This technique yields nearly perfect results for some procedural textures, however, many patterns will not give accurate results. The purpose of this option is to enable the ability to preview the scene quickly taking advantage of the capabilities of the graphics card. Note that selecting the higher numbers from this menu can add significant time to the initial rendering of the scene as each texture must be rendered at the selected resolution. This texture information is retained so that subsequent renderings do not experience this delay. This information is also retain in the form-Z project file (.fmz) so that the first rendering after opening a file that contains this information will also not have to wait for the textures to be rendered.

Bump Mapping: When this option is on (it is off by default), Interactive Shaded will render bump maps for objects that use the surface styles that contains a texture in its bump component.

Fog: When this option is on (it is off by default), Interactive Shaded will render the scene with a fog depth effect. There are two methods for setting the range of the fog.

Distance From View Point: When this option is selected, the position of the beginning of the fog is defined as a distance from the view point.

Distance From Hither: When this option is selected, the start of the fog is defined relative to the front clipping plane. The distance is expressed as a percentage

These parameters are correlated. When one changes, the value of the other is automatically adjusted. If, after the values are set in this dialog, the positions of the hither and yon planes are changed, whether new values are calculated on the basis of the hither plane or the view point depends on the option selected in this dialog.

The fog effect is the strongest when its base is set close to the hither plane (Distance From Hither = 0%). The farther from the hither plane (and the viewer) it is set, the weaker the effect becomes.
**Color:** The color in this box is mixed with the pixels of the image when producing the fog effect. This color can be changed by clicking on it.

**Background Mix:** This parameter indicates what percentage of the fog’s color will be mixed with the background. A mix of 0% renders the background as if no fog effect were specified. A mix of 100% renders the background with the fog color. Values between these two mix the fog effect color and the background at the specified percentage.

**Lights:** The options in this group determine how OpenGL, which is currently behind the Interactive Shaded display mode, will handle lights.

**Unsupported Lights:** This pair of options determines how types of lights that are not supported by OpenGL are handled, when they are encountered in a scene. These lights are the projector, custom, and area lights, which can be either approximated by another type or ignored.

**Ignore:** When this option is on, which is the default, the unsupported lights are not rendered.

**Approximate:** When this option is on, the lights are replaced with a supported light type that has parameters best approximating the effect of the original light. That is, projector lights are rendered as cone lights; area and custom lights are rendered as point lights. This option does not affect radiosity based renderings since the lighting calculations have already been incorporated into the radiosity solution.

There are two options that determine how lighting calculations are performed, which affects how many lights may be included in a scene.

**Use Native Lights:** When this option is selected, OpenGL handles all lighting calculations internally. This gives better performance (is faster), however, OpenGL has a limit on the number of lights it can handle. This limit varies depending on the implementation of OpenGL that is installed on your machine (8 lights is a common limit). The number of supported lights for the machine is displayed next to this option. When this option is on and an attempt is made to render a scene with more lights than the OpenGL limit, an error message will be posted. Note that the ambient light counts as one of the lights.

**Use Software Lights:** When this option is selected, the lighting calculations are performed by form-Z, which takes longer, but there is no restriction on the number of lights in the scene.

**Highlight Edge Size:** This option determines the width of the lines used for highlighting selected entities at the Point, Segment, Outline, Face, and Hole topological levels. This width is expressed in pixels. The default is 1 pixels. The highlight lines are in the highlight color, which is red by default. Picking at the Object or Group level paints the complete objects in the highlight color.

**OpenGL Info...:** Clicking on this button invoked the **OpenGL Info** dialog, as for Wire Frame (see Figure 3.6.2.6).

**Project Rendering Options...:** This item invokes the **Project Rendering Options** dialog as discussed in subsection 3.6.6 in this Manual.
3.6.3 Quick paint, hidden line, and surface rendering

Quick Paint*

This item is intended to provide fast previews of images and not final renderings. It applies quick sorting methods that work best when objects do not intersect and the sizes of their surfaces are evenly distributed. For example, quick paint works well with most of the meshes as illustrated in Figure 3.6.3.1. It is fast and is used to preview the results of operations. A more accurate rendering methods should be used for a final image. This item is affected by the Quick Paint Options dialog (Figure 3.6.3.2), invoked from it or from the Display Options dialog.

Show Color: When this is on (default), the colors assigned to the objects or their faces are used to render them. The colors of faces take priority over the object colors. When the option is off the scene is rendered using white color on dark background and black on light background, which results in images similar to those produced by hidden line.

Show Edges: This option applies only when the Show Color option is on. It is dimmed otherwise. When selected (default), the edges of the object are also shown.

Show Facets Of Smooth Objects: When this option is selected, the respective rendering is generated using the facets of the model rather than the edges of the smooth faces. This option is off by default.

Render With Shadows: With this option on, shadows are also rendered, using the current sun.

Show Transparent Objects: When this option is on, objects that have been assigned transparent surface styles will be displayed according to the following selection. When off, transparent objects are treated as invisible and are not displayed. A transparent surface style is one where the transparency parameter in the Surface Style Parameters dialog shows a value greater than zero.

As Solid/Surface: When this option is selected, transparent objects are treated as common opaque objects and are displayed accordingly.

As Wire Frame: When this option is on, transparent objects are displayed as wire frames.

Image Quality: These two radio buttons control the extent of sorting the program will execute. Note that, even though Better will resolve more sorting cases than Normal, it will still not handle intersecting surfaces.

All Windows: As for Wire Frame*.
Hidden Line*

Selection of this item generates a hidden line plot in the active window. The hidden line elimination method produces displays where only the visible lines of a scene are shown. The hidden line method resolves the intersections of surfaces, eliminates all back facing surfaces, and displays only the portions of the visible edges that are not obscured by other visible surfaces. The removal of hidden lines is a computationally intensive process that makes the hidden line method substantially slower than wire frame. An example of a hidden line plot is shown in Figure 3.6.3.4. Pressing the option key (Macintosh) or ctrl+shift (Windows) and selecting Hidden Line* invokes the Hidden Line Options dialog (Figure 3.6.3.5).

Show Object Color: When selected, the colors assigned to an object are used when plotting a hidden line image. When deselected, all lines are shown in black (default).

Include Open Shapes: When selected, the shapes that are open and do not define a surface are plotted. Such shapes cannot hide other solid objects, but may be (partially or completely) hidden by them. When deselected (the default), open shapes are skipped, and not plotted.

Show Transparent Objects: As for Quick Paint*.

Show Intersections: When this option is selected, the intersection lines of transparent objects displayed as wire frames are also displayed. They are not when this option is off.

Hide Edges With Angle Greater Than: When this option is selected and a hidden line plot is derived, edges whose adjacent faces form an angle greater than the value entered in the field to the right are hidden, except when they are on the object’s silhouette. This option is useful for line illustrations of curved objects, when the edges of smooth surfaces are not wanted. The angle can be any number between 0° and 180°. Examples are shown in Figure 3.6.3.6.

Show Facets Of Smooth Objects: This option is the same as for Quick Paint. Examples are shown in Figure 3.6.3.7.
**Save Image**: The three items in this pop up menu set the automatic saving of images after rendering is completed.

- **Off**: When this is selected, which is the default, the rendering is not saved.
- **Manual**: When this item is selected, the file name, format, and location are determined by invoking the standard file save dialog prior to the execution of each rendering.
- **Auto**: When this item is selected, the file name is generated automatically by combining the project’s name and the name of the current view. The file is then saved at a specified location. The format and location of the file can be selected in the **Image Autosave Options** dialog (Figure 3.6.3.8), invoked by pressing the **Options...** button next to the menu. The options in this dialog are as follows:

**Format**: The file format in which the rendering will be saved is chosen from this menu. **Options...** invokes the dialog of the respective format.

**Destination**: The image file can be saved **With Project** or in a Custom defined location, which is done by clicking on the **Set Destination Folder...** button to invoke the standard folder location dialog.

**Generate Unique Image File Names**: When this option is selected, the file name is constructed so that existing files will not be overwritten by new image files. This is done by appending incremental numbers to the file name.

**File Info**: This section displays the file name and location of the image file to be saved.

**All Windows**: This last option in the **Hidden Line Options** dialog is as for **Wire Frame**.
**Surface Render***

This item produces colored renderings with shades and possibly shadows that depend on the current position of the sun light source. *Surface rendering* paints the faces of a model from the farthest to the closest. Each face is painted on top of those painted previously and hides them. This is commonly known as the “painter’s algorithm.” Examples are shown in Figure 3.6.3.9. Pressing the **option** key (Macintosh) or **ctrl+shift** (Windows) and selecting **Surface Render*** invokes the **Surface Rendering Options** dialog shown in Figure 3.6.3.10.

![Figure 3.6.3.9](image)

**Figure 3.6.3.9:** Surface rendering (a) with edge lines and (b) without edge lines.

**Show Color:** When selected, the colors assigned to the object or its faces are shown. When off, the object is shown white with black edges or black with white edges, depending on the background color of the graphics window.

**Show Edges:** When selected (default), the edges of the faces are also plotted, using a **Quick** or an **Accurate** method. The former method plots the edges of the faces at the time each face is rendered. This may cause part of an edge to be covered by subsequently rendered faces. The latter method uses two sortings and two passes. All the faces are first rendered, and then their visible edges are plotted.

**Render With Shadows:** When this option is selected, the shadows of the objects are also calculated and rendered, using the current light position.

**Show Transparent Objects** and **Show Intersections:** As for **Quick Paint***.

**Show Facets Of Smooth Objects:** This option is the same as for **Quick Paint**.

**Save Image:** As for **Hidden Line***.

**All Windows:** As for **Wire Frame***.

![Figure 3.6.3.10](image)

**Figure 3.6.3.10:** The **Surface Rendering Options** dialog.
3.6.4 Shaded rendering

Shaded Render*

This item produces the highest quality images in the regular version of form•Z. It actually utilizes a subset of the RenderZone* rendering procedures up to the level of z-buffer. It can produce smoothly shaded images, apply transparencies, antialiasing, and cast both soft and hard shadows from multiple lights. It does not however produce special rendering effects such as texture mapping, reflections, background and depth effects, etc., which are produced by form•Z RenderZone Plus.

Selecting this item while pressing option (Macintosh) or ctrl+shift (Windows) invokes the Shaded Rendering Options dialog shown in Figure 3.6.4.1. It consists of two tabs, Shading and Geometry, as well as a number of options outside the tabs. The latter will be discussed first and then the options in the tabs.

![Shaded Rendering Options dialog](image)

**Figure 3.6.4.1:** The Shaded Rendering Options dialog with (a) the Shading and (b) the Geometry tabs selected.

Internally, the Shaded Render* command executes a number of distinct rendering procedures, ranging from flat to z-buffer rendering. The level of rendering procedure used by form•Z depends on the options selected in the Shaded Rendering Options dialog, as summarized in the Table of Figure 3.6.4.2. However, the user need not be concerned about what rendering procedure is executed, since the program does the selection automatically on the basis of the options selected.
### Rendering Effects and Methods

<table>
<thead>
<tr>
<th>Rendering effects applied:</th>
<th>Rendering method used:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth Shade</td>
<td>Flat</td>
</tr>
<tr>
<td>Antialiasing</td>
<td>Flat oversampled</td>
</tr>
<tr>
<td>Transparency</td>
<td>Gouraud/Phong</td>
</tr>
<tr>
<td>Shadows</td>
<td>Gouraud/Phong oversampled</td>
</tr>
<tr>
<td>No</td>
<td>Z-buffer</td>
</tr>
<tr>
<td>Yes</td>
<td>Z-buffer oversampled</td>
</tr>
<tr>
<td>No</td>
<td>Z-buffer</td>
</tr>
<tr>
<td>Yes</td>
<td>Z-buffer</td>
</tr>
</tbody>
</table>

*One or both of these two effects used.

**Figure 3.6.4.2:** The shaded rendering effects and the rendering methods that produce them.

### Options outside the tabs

**Set Image Size:** When this option is selected, before starting the execution of the rendering, form-Z allows you to delineate the area of the model you wish to render. When Set Image Size is on and the Shaded Render* command is executed, the cursor appears as a brush icon (🖌️) and the tool palette is grayed out to indicate that it is temporarily inactive. A rectangle is defined by clicking the mouse twice to enter two opposite corners of the rectangle. After the first click, a rectangle is rubber banded allowing you to visually define the portion of the scene that you wish to render. An example is shown in Figure 3.6.4.3.

**Figure 3.6.4.3:** Rendering part of a scene:
(a) delineating the area to be rendered, and (b) partially rendered image.

**Project Rendering Options...** Clicking on this button invokes the Project Rendering Options dialog, which is also accessed from several of the display mode options dialogs. Its content is discussed in the next section.

**Save Image:** As for Hidden Line*.

**All Windows:** As for Wire Frame* and the other display modes.
Shading tab

Antialiasing: When this is selected, the rendered image will be antialiased, that is, the jaggies that appear along edges that are not horizontal or vertical will be smoothed.

Transparencies: When this option is selected, objects to which transparent surface styles have been assigned will be rendered transparent. The degree of transparency is as set in the respective surface style. Recall that 0% transparency results in an opaque object, and 100% transparency results in a completely transparent and therefore invisible object. Note that this option can only be selected when Antialiasing is on. This is due to the fact the rendering method which supports transparency also automatically applies antialiasing.

Shadows: When this option is selected, objects whose shadow attribute is on will cast shadows from the light sources whose shadow option is on, onto objects whose shadow receiving attribute is on. Recall that the shadow casting and receiving attributes of objects are turned on by default when objects are created. The shadow option of lights also is on by default when lights are created. The shadow type, Opaque, Transparent, or Per Light is selected from the popup menu. Transparent Shadows can be selected only if the Transparencies option is selected above. If All Transparent is selected, a partially transparent surface will filter the light which passes through it by the surface color, resulting in a colored shadow. If All Opaque is selected, as if the surface is not transparent, shadows are always opaque. While these options are applied globally to all the lights, when Per Light is selected you are allowed to set the type of shadow per individual light.

Show Grid, Axes And Underlay: If this option is on, the grid and axes are generated as a 2D background image. That is, a 3D object will always cover the grid and axes, even if they pass through the object.

Render Grid And Axes As 3D Lines: When this option is selected, the grid and axes are generated as 3D entities. As a result, if the grid is at a location where it passes through the middle of an object, the rendering will show the grid lines intersecting properly with the object. The axis labels (x, -x, y, -y, z, -z) are not rendered with this option.

Unsupported Lights: These two options determine what to do with lights other than distant, which may have been set for RenderZone* and are not handled by Shaded Render*.

Ignore: When this option is selected, all the lights other than distant that may be in a scene are ignored and not included in the rendering.

Approximate: When this option is on, all the non-distant lights are approximated and processed as distant lights. Note that, when this option is on, there is a high risk of overexposing the rendered image. This is because the other types of light typically affect a small area of a scene. When they are processed as distant lights, they illuminate the whole scene. This additive effect of all the lights may result in a dramatically overexposed image.
Geometry tab

Decompose Non Planar Faces: The rendering produced by the Shaded Render* item does not require that faces be planar, and does not automatically triangulate the faces of objects, even when they are non planar. This represents a significant savings in rendering time. However, if objects in a scene contain non planar faces, the rendering may not be accurate. This typically depends on the degree to which non planar faces deviate from planarity. Selection of this option instructs the program to triangulate the non planar faces, and should be used when a scene contains non planar faces. This option is off by default.

Wire Frame Width: This pair of options determines how wire frame lines that may be included in a shaded rendering will be plotted. Recall, that whether an object is rendered as a shaded surface or as a wireframe or both is set through the Render Attributes tool.

  Scale By: When this option is selected, the line widths of all wire frames are scaled by the factor entered in the text field. This is useful for example when in a final rendering the image resolution is increased from 72 to 300 dpi, in the Image Options dialog. In such a case scaling the line width by 300/72 (=4.16) preserves the widths relative to the image size.

  Set To n Pixel: When this option is on, all wire frames are displayed with the line width entered in its field. That is, this option overrides the widths assigned to individual objects with the Rendering Attributes tool.

Silhouette Quality: This slider bar controls the quality of shading of smooth objects along their silhouette edges. At rendering time, form•Z will generate a facetted representation of smooth objects. If this slider is set to its highest value, the facetting is done so densely, that no facetting artifacts can be observed along the silhouette edges of curved surfaces. If the slider is moved to the left, less facets will be created, which results in memory savings and an increased rendering speed, but will also produce visible facetting artifacts.

Surface Quality: This slider bar controls the quality of shading of smooth objects on their surfaces. This setting controls the facetting of curved surfaces on their insides independently from the Silhouette Quality option. In general, the Surface Quality slider can be set to a lower value than the Silhouette Quality and still result in high quality renderings.
3.6.5 Doodle rendering

Doodle is a plugin renderer that produces line drawings with a hand-drawn appearance. Instead of drawing the original edges of models as wire frame or hidden line do, doodle acts as a filter. Each original line is modified by one or more effects to create an image that appears to be hand-drawn. Each effect performs a simple operation on a line, which changes its appearance and may split it into multiple lines. These effects applied individually can achieve some interesting results. However, the power of the doodle renderer is realized when it combines, in some order, a few effects. The permutation of effects that produce a final look is called a style. Several styles come with the Doodle plugin. Doodle renderings can be based on a wire frame or hidden line rendering. Figure 3.6.5.1 shows a few examples of doodle renderings.

![Sample doodle renderings](image_url)

*Figure 3.6.5.1: Sample doodle renderings: (a) wire frame rendering, (b) doodle rendering with the Rough, (c) Simple Dashed, and (d) Freehand default styles.*
**Doodle***

Selecting this item generates a “doodled” drawing in the active window. Pressing the option key (Macintosh) or ctrl+shift (Windows) while clicking on **Doodle*** invokes the **Doodle Options** dialog shown in Figure 3.6.5.2. The dialog consists of three parts: **Base Rendering** at the top, **Doodle Styles** in the middle, and **Style Parameters** at the bottom.

**Base Rendering**: This section determines the rendering type that is used as the starting point. A doodle style is applied to each line from the base rendering to create the doodle rendering.

**Wire Frame**: When selected, the base rendering is a wire frame rendering (see section 3.6.2 in the form•Z User’s Manual). Note that only a small subset of the original wire frame options are useful to doodle renderings. Examples of wire frame based doodle renderings were shown in Figure 3.6.5.1.

**Show Back Faces, Show Color**: As for the **Wire Frame Options** dialog.

**Show Facets Of Smooth Objects**: When on, the facets of smooth objects are shown. When off, which is the default, the iso lines of the smooth objects are shown. For doodle renderings the iso lines are shown at half of the intensity of the facets.

**Hidden Line**: When selected, the base rendering is a hidden line. This rendering is the same as the rendering produced by the **Hidden Line*** menu command (see section 3.6.3 of the form•Z User’s Manual). The **Options...** button invokes the **Hidden Line Options** dialog (see Figure 3.6.3.5 in the User’s Manual). Examples of hidden line based doodle renderings are shown in Figure 3.6.5.3.
Doodle Styles: This section contains the available styles that can be used for doodle renderings. Each style is shown as a small preview of a cube rendered with the respective style. One of the styles in the list is the current style. The current style is shown with a heavy black border. Clicking on a style makes it the current style. The current style is the one that is used for rendering in the project window. The parameters of the current style can be edited by setting options in the Style Parameters section.

- **New**: Clicking on this button creates a new empty style and makes it the current style.

- **Copy**: Clicking on this button creates a new style by copying the current style and makes it the current style.

- **Delete**: Clicking on this button deletes the current style, which is a permanent deletion. Deleted styles will not show up even after restarting form-Z. The style adjacent to the deleted style becomes the current style.

- **Defaults...**: Clicking on this button restores all default doodle styles to their original default parameters. If any additional doodle styles have been created (with the New or Copy buttons) they can be optionally kept or deleted.

*Figure 3.6.5.3:* (a) Hidden line rendering, (b) Hidden line based doodle renderings using the **Rough**, (c) **Simple Dashed**, and (d) **Freehand** default styles.
**Style Parameters**: This section is used to edit the parameters of the current style. A style is composed of a set of effects. Each effect has its own parameters. The order of the effects is significant as the effects are applied sequentially and the result of one effect is the base for the next effect. When the order of the effects is changed, the resulting style is almost always different, as shown in Figure 3.6.5.4. In this example, the *Extend* and *Jagged* effects are used in both images. In (b) the order is *Extend* and then *Jagged*, while in (c) the order is *Jagged* and then *Extend*.

Some effects create multiple lines (*Break, Curl, Multi, Jagged, Bleed*), while others only modify the source line (*Extend, Skew*). If an effect that creates multiple lines is applied, the next effect operates on each of the lines that resulted from the effect. A satisfying style can usually be created with just a few effects. A large number of effects for a single style tends to affect performance and to produce a confusing result.

![Figure 3.6.5.4: Effect ordering: (a) Wire frame cube (b) Extend and then Jagged, (c) Jagged and then Extend.](image)

**Name**: This is the name of the style. It must be unique.

**Effects**: This list shows the effects of the current style and the order in which they are applied (top to bottom). One of the effects in the list is highlighted to indicate that it is the *current effect*. Clicking on the name of an effect in the list makes it the current effect. Effects in the list can be reordered by clicking and dragging the current effect to the desired location in the list.

**Effects Library**: This list shows the library of effects that can be used in a style. One of the effects in the list is highlighted to indicate that it is the *current library effect*. Clicking on the name of an effect in the list makes it the current library effect.

**Add**: Clicking on this button adds the current library effect to the end of the *Effects* list and makes it the current effect.

**Replace**: Clicking on this button replaces the current effect with the current library effect.

**Delete**: Clicking on this button deletes the current effect from the *Effects* list.

**Delete All**: Clicking on this button deletes all the effects from the *Effects* list.

**Preview**: This image is a preview of the style. It is updated when the *Effects* list or a parameter of an effect is changed.
Effect Parameters

Under the lists and buttons is a section that displays the parameters of the current effect selected in the **Effects** list. Both the title and the content of this section change as a new current effect is selected. The non regular results of the doodle rendering are achieved by randomization of the parameter values. This is controlled by setting the minimum and maximum values for each parameter in the style. More variance will appear in the image when the minimum and maximum values are farther apart. Less variance will appear in the image when the minimum and maximum values are closer. The parameter values are specified as a percentage of the width of the image. This allows the doodle image to look the same regardless of the image resolution.

All effects use the following common parameters:

**Minimum**: This slider controls the minimum value for the parameter. The text field to the right specifies the percent of the image for the minimum value. The second text field indicates the number of pixels that this percentage yields for the active window.

**Maximum**: This slider controls the maximum value for the parameter. The text field to the right specifies the percent of the image for the maximum value. The second text field indicates the number of pixels that this percentage yields for the active window.

**Lock**: This check box locks the minimum and maximum values to each other which yields no variance of the parameter. When this is enabled, the minimum and maximum values change together.

Following are the parameters for each effect:

**Extend Parameters**

This effect lengthens or shortens each line. It has a single parameter, as shown in Figure 3.6.5.5.

**Length**: The amount to lengthen (positive) or shorten (negative) each line.

*Figure 3.6.5.5: Extend effect: Length of (a) -3 %, (b) 3% and (c)10 %.*
Break Parameters
This effect breaks up each line into several colinear line segments for a dashed appearance, as shown in Figure 3.6.5.6.

Segment Length: The length of the solid part of each line segment.

Gap Distance: The distance of the gap between solid parts.

Curl Parameters
This effect curls each line, giving it a few or several curves. It actually creates several small straight segments. Thus, when combined with another effect, the other effect will operate on all of the small segments, which may yield unexpected results. Examples of this effect are shown in Figure 3.6.5.7.

Distance: This controls the distance to a bend or curve of the line. Higher numbers yield smoother curves. Smaller numbers yield noisier and bumpier lines.

Amplitude: This parameter controls how high the waves of the curves reach.

Maintain End Points: If this option is on, each line’s end points are not moved from their original position. This allows an object to maintain its cohesive form by being connected at the corners.

Figure 3.6.5.6: Break effect with Segment Length and Gap Distance of (a) 0.8 %, 1.3 %; (b) 1.8 %, 6.0 %; and (c) 6.5 %, 1.5 %.

Figure 3.6.5.7: Curl effect with Distance and Amplitude of (a) 14.5 %, 1.3 %, (b) 2.2 %, 0.2 %; and (c) 2.2 %, 2.8 %.
**Bleed Parameters**

This effect creates thicker lines at the end points of each line for a “bleeding pen” effect.

**Length:** The length of the bleed lines at each end of a line.

---

**Multi Parameters**

This effect creates multiple lines for each original line and offsets them some distance from the original, as shown in Figure 3.6.5.9.

**Lines:** This is the number of new lines.

**Distance:** This is the distance by which the lines are offset from each other.

---

*Figure 3.6.5.8:* Bleed effect with **Length** of (a) 1.1 %, (b) 5%, and (c) 9%.

*Figure 3.6.5.9:* Multi effect with **Lines** and **Distance** of (a) 2, 0.6 %; (b) 3, 1.4 %; and (c) 5, 2 %. 
Jagged Parameters

This effect creates a jagged or zigzag line from each original line, as shown in Figure 3.6.5.10.

**Distance**: This is the distance between “jags” (changes in direction). It is calculated along the original input line. Higher values yield less and smaller yield more jagged lines.

**Amplitude**: This parameter controls the height of the jagginess.

**Maintain End Points**: If this option is on, the end points of the lines are not moved from their original position. This allows an object to maintain its cohesive form by being connected at the corners.

Skew

This effect twists or offsets each line by moving the end points or rotating the line. Examples are shown in Figure 3.6.5.11.

**By Distance**: When this option is on, the values in its fields specify the distance to which the end points of each line are moved.

**By Angle**: When this option is on, the values in its fields determine the angle by which each line is rotated.

**Figure 3.6.5.10**: Jagged effect with **Distance** and **Amplitude** of (a) 0.8 %, 1.4 %; (b) 2.2 %, 0.2 %; and (c) 8.7 %, 1.4 %.

**Figure 3.6.5.11**: Skew effect with **Distance** of (a) 1.4 %, (b) 3 %; and (c) **Angle** of 12 %.
**Perspective Depth:** This option is only enabled when the current window contains a perspective view. When enabled, the parameters of the current style are adjusted relative to the depth of the lines in the scene. That is, the style parameters for each line are scaled down proportionally to the depth of the lines in the scene. The slider to the right determines the scale factor that is applied. The default is 50%. A value of 100% indicates no scaling and a value of 0% indicates the most scaling. Figure 3.6.5.12 shows an example of perspective depth. In this example the fuzzy style was used with the parameters locked which removes the randomness from the style and makes it easier to see the perspective depth effect.

**All Windows:** This option functions as with all other renderers with the same option.

*Figure 3.6.5.12: Perspective Depth:* (a) off, (b) 50%; and (c) 0%.
3.6.6 Project level rendering options

Several of the display mode options dialogs and the Radiosity Options dialog include a button labeled Project Rendering Options.... This is also an item available in the Display menu (see section 3.6.9). Clicking on it invokes the Project Rendering Options dialog, which offers condensed rendering parameters, stored on a per project basis.

The complete Project Rendering Options dialog consists of two tabs: Smooth Shading and Default Texture Map Control. Depending on which rendering mode the dialog is invoked from, it contains one or both of the tabs. For example, both tabs are included in the dialog invoked from RenderZone*, as shown in Figure 3.6.6.1. The Project Rendering Options dialog invoked from Shaded Render* contains the first tab only. It will be discussed here. The Default Texture Map Control tab is discussed in the RenderZone Plus User’s Manual (see section 6.3).

Smooth Shading tab

Facetted Objects: The options in this group determine whether or not facetted objects will be smooth shaded when rendered.

None: When this option is selected, facetted objects do not get smooth shaded.

Edges With Angle Greater Than: When this option is selected, facetted objects are smooth shaded, where the angle between two faces (across an edge) is larger than the angle entered in the text field.

Smooth Objects: The options in this group determine how smooth objects are treated when rendered.

Use Facetted Objects Setting: When this option is selected, smooth objects are rendered with the option selected for facetted objects (None or Edges With Angle Greater Than).

Figure 3.6.6.1: The Project Rendering Options dialog with the Smooth Shading tab open.
**Per Renderer:** This option allows each rendering type to render smooth objects as best as it can. Some renderers, such as Surface Render or Interactive Shaded, just use the faces as shown in wireframe. Other renderers, such as RenderZone, will render smooth objects so that no faceting artifacts can be noticed.

**Allow Object And Layer Override:** When this option is on, objects that have their own Smooth Shading attribute, will override the settings of the Project Rendering Options. The same is true, if a layer uses a Smooth Shading override attribute. If this option is off, only the Project Rendering Options apply. For example, if a scene needs to be rendered completely flat shaded, the Facetted Objects option should be set to None, the Smooth Objects option set to Use Facetted Objects Setting and the Allow Object And Layer Override should be turned off.
3.6.7 Lights and shadows

When using Quick Paint* or Surface Render*, objects cast shadows only from the light designated as the sun. When using Shaded Render* or RenderZone*, all the active lights may cast either hard or soft shadows, or no shadows at all. The light attributes are discussed in section 2.10. For shadows to be cast, the following parameters must be set properly:

- The shadow casting attribute of an object must be on. It is on by default when a new object is created. You can render a scene with some objects casting and some not casting shadows.
- The shadow receiving attribute of an object must be on.
- The light’s shining attribute must be on in the Light Parameters dialog or Lights palette.
- The Shadows option must be on for this light in the Light Parameters dialog or Lights palette. For Shaded Render* or RenderZone* renderings, the type of shadow (Soft (Mapped) or Hard (Raytraced)) is selected from the Shadows pop up menu. For Quick Paint* and Surface Render*, the sun’s shadow option is assumed always to be on.
- Shadows must be on in the Options dialog of the type of rendering executed. Recall that these dialogs can be invoked directly from the menu item before executing the operation.

The following is a discussion of the shadow parameters of a light. They can be edited in the Light Parameters dialog (Figure 2.10.3.1).

When the Soft (Mapped) (default) type of shadows is selected from the Shadows menu, additional options that affect the quality of the shadow can be selected from the Shadows tab. These options only effect Shaded Render* and RenderZone*. Shadow casting in general adds a significant amount of computation time to the rendering of a scene. When using soft shadows in a Shaded Render* or RenderZone* rendering, the higher the quality of the shadow, the longer it will take to complete the rendering. Consequently, it is generally recommended that lower qualities of shadows be used initially and the higher quality shadows be used for a final rendering. Each additional light adds to the time required for a rendering and consumes additional memory. In general, soft shadows require significant quantities of memory but are faster than hard shadows.

Quality: A Low, Medium, or High quality of shadow can be selected from this pop up menu. Low quality requires less time and less memory, but may show noticeable jaggies at the perimeter of shadows. High quality shows less jaggies, but requires more time and memory.

Softness: This sliding bar controls the smoothness of the edges of the shadows. Zero (0) softness produces the hardest shadow edge, while 100 produces the fuzziest edge.

Tolerance: This parameter is used to eliminate self-shadowing artifacts. Note that when the Softness parameter is increased, it is possible that a surface casts shadows on itself. This artifact manifests itself as random patterns of darker pixels on that surface. Increasing the Tolerance parameter reduces these artifacts. Note, however, that high Tolerance values may cause the shrinking of the area of a shadow. In some cases this will create the effect of an object floating above a plane, although the object is placed directly on the plane.
Resolution: Soft shadows for Shaded Render* and RenderZone* are generated by producing a shadow map for each of the shadow casting light sources before any rendering occurs. These maps are then used during rendering to determine whether a pixel is in a shadow region or not. Shadow maps can be produced at different sizes (levels of resolution).

If the n Times Image Size option is selection, the resolution of the shadow map depends on the size of the image at the time the rendering is executed. If, for example, the image size is 500 x 500 pixels and a value of 2 is entered, the shadow map is generated at a resolution of 1000 x 1000 pixels. Choosing this option guarantees that the quality of the shadows is the same even if the window is resized between individual renderings. If the m by n Pixels option is selected, the shadow map is created at a fixed resolution, defined by the value entered in the text field. This option guarantees that the memory consumption for the shadow map is the same for each rendering, regardless of the image size. However, if the image size is increased, the shadow quality may decrease.

Limit Map To: One of three available items can be selected from this pop up menu, which determines whether objects outside the view will cast shadows. This selection affects the size of the shadow maps and the quality of the shadows cast in an indirect way.

All Objects: When this option is selected (default), the shadow map is generated for all the objects in the project that cast shadows, even those that are completely out of the view.

All Completely Visible Faces: When this option is selected, only the faces which are entirely visible in the screen are used to determine the shadow map.

All Completely Visible Objects: When this option is selected, only the shadow casting objects that are completely visible in the displayed view will be used to calculate the shadow map. The last two options can increase the crispness and quality of the shadows significantly, but will produce inaccurate shadows whenever objects that are not visible in the window cast shadows on objects that are visible.

Warnings: As already noted, soft shadows are generated by producing a shadow map before any rendering is executed. This shadow map includes all the objects that are set to cast shadows, regardless of whether or not they actually do. For example, a more or less flat ground plane with the sun above it does not produce any shadows, even though it accepts shadows from other objects that may be sitting on it. If such a ground object is significantly larger than the objects that actually cast shadows, a shadow map which is calculated at the size of the screen will only contain a small portion of shadows. If a close up rendering containing only the smaller objects that cast shadows is produced, the shadows will be fuzzy. This is because the small portion of the shadow map that contains the shadows is magnified to (possibly) fill the screen. This causes the edges of the shadows to lose their crispness. To avoid this, when a scene contains objects such as ground planes that do not cast shadows, the shadow attribute of these objects should be turned off. In general, crisp soft shadows are achieved when the size of the shadow map is close to the size of the rendering. For crisper shadows, you should avoid objects which either do not cast shadows or which are outside the view and therefore cast shadows which will not be visible. This will not only produce crisper shadows, but it will also render faster. These distinctions are illustrated in Figure 3.6.7.1.
Menus • The Display menu

These considerations should be made in conjunction with the Limit Map To options. If there are objects outside the view that do not cast shadows on visible objects, then the All Completely Visible Faces or All Completely Visible Objects option should be selected. This will result in smaller shadow maps and thus crisper shadows.

Shadow maps can be very memory intensive and should be used conservatively if your machine has limited memory. Each pixel in a shadow map requires 4 bytes of storage. For example, a map of 500 x 500 pixels requires 1 Megabyte of memory. Light Type, Map Resolution, Quality, and Softness are parameters relative to each other. In general, shadows with a high softness require a larger shadow map and a higher Quality parameter. Low resolution shadow maps may result in jaggy shadow edges and self-shadowing artifacts, which show as dark patterns. The memory consumption of the shadow map is shown in the Memory Usage field.

Hard (Raytraced) shadows do not use shadow maps but raytracing, and always produce crisp edges. They are only available to Shaded Render* and RenderZone*. Their memory requirements are minimal but generally require more processing time. Hard (Accelerated) shadows are raytraced but also use shadow maps to speed up the raytracing procedures. Their memory requirements are the same as soft shadows. Which shadow type should be used in a rendering scene depends on the desired effects. Since the shadow type is an independent attribute of each light, you can mix the two types of shadows within the same rendering.
3.6.8 Network rendering

Network rendering uses a group of machines (referred to as a rendering farm) to render an image or animation by splitting the rendering task among the available machines. This allows form•Z on the original machine to continue to be used for modeling or other local operations. There are three components of network rendering: the form•Z application, the form•Z Render Server, and the form•Z Render Client. These are separate applications that require specific installation. There are separate installers on the form•Z CD for installing the form•Z Render Server and form•Z Render Client. These do not require a form•Z hardware key.

TCP/IP must be enabled on any machine that will run any of the components. Consult your operating system’s user manual for specific details on how to enable TCP/IP. It is on by default for most modern operating systems. In order to access the server, the form•Z application and the render client need to know the TCP/IP address and port number of the server. A server can be added by network browsing or by machine name on networks that support it (see the Server Settings dialog in Figure 3.6.10.4) for details. The port number is an identifier on the server that identifies the render server network traffic from other network traffic. The default port number is 28838.

The form•Z application is used to create the model and setup the rendering scene and parameters. The scene is then sent to the form•Z Render Server to be generated. This scene is called a render job. The server manages all the jobs that have been sent to it and distributes the work to the form•Z Render Client software running on each machine in the farm. When the rendering or animation is complete it can be retrieved from the server by form•Z.

The rendering content files (form•Z project files, textures, etc.) are transferred and stored on the server machine so that they are available for rendering. The files are also transferred to the client when needed for rendering. The files are stored in directories on the server and client machines based on how the form•Z application connects to the server. form•Z supports two types of connections: registered user and guest. When connected as guest, the files are only kept on the server until rendering is completed and then they are removed. When connected as a registered user the files can be kept on the server and client machines so that subsequent network renderings are faster, since they do not need to redistribute the content files. See below for details on connection types.

All of the networks rendering components communicate using TCP/IP. This allows the components to be installed in mixed environments of Macintosh and Windows operating systems. The other advantage of using TCP/IP is that it allows rendering over the Internet. The form•Z application, server, and clients can be at completely different addresses, anywhere in the world. Note that the speed of the network between the components can be a big factor in how fast the farm can generate images. This is especially true of the connection between the server and the clients, as they must communicate frequently during generation of the images. Multiple components can be run on the same machine, however, this will degrade performance.
Network rendering is accessed in form-Z through three items in the Display menu: **Network Render...** is used to initiate static renderings on the network and **Network Render Status...** is used to check the status of the network rendering and to retrieve completed renderings. The **Animation Generation Options** dialog, invoked from the **Generate Animation...** item in the Display menu, has options for network renderings of animations.

**Network Render...**

This menu item is used to initiate a static rendering job over the network. When invoked for the first time after installing or whenever there are no known render servers, the **No Known Servers** dialog is invoked as shown in Figure 3.6.8.1. This dialog provides a convenient way to get connected to a render server. The **Add...** and **Browse...** buttons are the same as in the **Network Settings** dialog discussed in the next section.

Selecting this menu item invokes the **Network Render** dialog shown in Figure 3.6.8.3. This dialog is used to name the rendering job, select the rendering type, and select the render server to be used. Once the desired options have been set, clicking **OK** initiates the network rendering and places it in the job queue. If the project has not been saved, the standard File Save As dialog appears to save the project before network rendering can occur.

When sending a job to the server that has changed since it was last saved, form-Z presents an alert dialog (3.6.8.2) asking the user if they want to save the change to the project file (**Save Project File**) or write the changes to a temporary file (**Write Temporary File**). The **Write Temporary File** option is for changes that the user wants rendered on the network but doesn’t want saved with the project.
The options of the **Network Render** dialog (Figure 3.6.8.3) are as follows:

**Job Name**: The name of the rendering job is entered in this field. This identifies the job in the job queue.

**Renderer**: This pop-up menu identifies the rendering mode to be used for the network rendering. The **Options...** button to the right invokes the corresponding options dialog for the selected rendering mode.

**Server**: This pop-up menu identifies the rendering server that the job will be sent to. The **Settings...** button invokes the **Network Settings** dialog, discussed in the next subsection, which is used to add servers that will be displayed in the menu.

**Priority**: This slider determines the priority assigned to the job. If the priority selected exceeds the user’s priority, the job priority will be capped to the user priority.

**Keep Project On Server**: When this option is selected, rendering content files (form•Z project files, textures, etc.) are kept on the server for future renderings. This option is only available when the **Registered User** option is selected for the current server, as set in the **Server Settings** dialog.

**Project**: This pop-up menu identifies the name of the project to use for the rendering job. The project name is used to identify all of the rendering content files for the project. This name will be used in the directory name on the server machine where the files are stored.

**Add...**: This button can be used to add a new project name to the list. The **Add Project** dialog is invoked for entering the name of the project.

**Remove...**: Clicking on this button removes the current project from the project list. This will also delete any associated jobs which use this project. This will remove the project and its jobs from the server as well.
File Retrieval: This group of options is used to determine how the completed renderings and animations are retrieved from the server.

Manual: When this option is selected, completed renderings and animations are retrieved manually by the user. Manual retrieval is done by selecting the Retrieve... button in the Network Status dialog.

Automatic: When this option is selected, completed renderings and animations are automatically retrieved when they are completed. Note that automatic retrieval only happens while the form-Z application is running. The remainder of the options in this group are the same as the corresponding options in the Image Autosave Options dialog as documented in Section 6.1.1 of the form-Z RenderZone Plus User’s Manual.

Display Image After Retrieval: When this option is selected, the completed rendering will automatically be displayed, once retrieved from the Server.

Leave File On The Server: When this option is selected the completed rendering or animation is left on the server after retrieval. This option allows for future manual re-retrieval.

The Network Settings dialog

This dialog, shown in Figure 3.6.8.4, is invoked by clicking on the Settings... button in the Network Render Status dialog. It maintains the list of servers form-Z can use for network rendering. It also contains settings used when connecting to the server.

Servers: This list contains a list of all of the servers that the form-Z application is aware of. The current server is highlighted in the list.

Browse...: This button invokes the Browse For Server dialog (see Figure 3.6.8.5).

Edit...: This button is used to edit the settings for the current server. The Server Settings dialog is invoked for editing the server address and login information (see Figure 3.6.8.6).

Figure 3.6.8.4: The Network Settings dialog.
Add...: This button is used to add a server to the list. The Server Settings dialog is invoked for entering the server address and login information of the server to be added (see Figure 3.6.8.6).

Remove: This button is used to remove the current server from the list. This will also delete any associated jobs which use this server.

Settings: This group of options controls parameters that affect the link between the form•Z application and the Render Servers.

Communicate With Server Every n Seconds: This setting controls how frequently form•Z checks with the server. A smaller time will improve responsiveness in retrieving completed images, updating the Network Status dialog and receiving error messages. A smaller time will also slow down other operations in form•Z while a job is pending or rendering on the server.

Port: This group of options allows you to designate which port will be used.

Static: When this is selected, form•Z will use the port number entered in the text field to the right of this option as the port number. Care must be taken that this port number is not being used by any other application on the same computer. The default static port number is 28837.

Dynamic: When this is selected, form•Z will automatically use the free port number that the Operating System selects. This is the default option.

The Browse For Server dialog

This dialog, shown in Figure 3.6.8.5, shows a list of all servers found on the network by browsing. When the dialog is opened, the browser starts searching for servers. As each server is found, it is added to the server list. At any time you can select a server or close the dialog without waiting for the search to finish.

Servers: This list contains the names of all the servers found on the network by browsing. The current server is highlighted.

Refresh: This button is used to rebrowse the network for servers.

Add...: This button is used to invoke the Server Settings dialog to enter login information and add a server to the list of known servers.

Figure 3.6.8.5: The Browse For Server dialog.
The Server Settings dialog

This dialog, shown in Figure 3.6.8.6, is used to add or edit a server in the server list. It can be invoked in different ways: (1) by clicking on the Add... button of the Network Status dialog; (2) by clicking on the Edit... button after selecting a server in the list of the Network Status dialog; and (3) by clicking on the Add... button after selecting a server in the Browse For Server dialog, which is invoked by clicking Browse... in the Network Status dialog.

**Server**: This group specifies the server location.

- **Server**: This is the TCP/IP address or Machine name of the machine on which the server is running as shown at the top of the render server window.

Note that Machine names will only work under the following conditions:

- On any platform if the server machine name is registered with a DNS server.
- On windows if a WINS server is running in the server's domain.

- **Port**: This is the TCP/IP port that the server is running on. It must be the same as the port set in the Server Configuration dialog of the render server (Figure 3.6.8.13). The default is 28838.

- **Connect As**: This group of options determines how the form•Z application connects to the server. When a registered user name is used, rendering content files can be kept on the server after renderings are completed so that subsequent renderings do not require data to be re-uploaded to the server unless it has changed.

- **Registered User**: When selected, the form•Z application connects using a user name and password that have been setup in the form•Z render server's Users section.

  - **Name**: This field contains the user name to use when connecting to the server.

  - **Remember Password**: When this option is selected, form•Z remembers the specified password and it does not need to be entered when connecting to the server. When this option is off, which is the default, the password will be prompted for when the form•Z application connects to the server.

- **Guest**: When selected, the connection is established using a guest connection.

- **Settings**: This group of options controls parameters that affect the link between the form•Z application and this Render Server.

  - **Synchronize Time To Server**: When this option is on, the date and time of the form•Z machine is changed to match the server’s time, when form•Z connects to the server. Only one server can be chosen to synchronize with.
Network Render Status...  

Selecting this menu item invokes the **Network Render Status** dialog as shown in Figure 3.6.8.7. It is used to manage the rendering jobs that have been sent to the server.

**Job Queue**: This list shows the list of all of the jobs that have been sent to a render server. Clicking in the list makes the selected job the active job of the list.

**Pause/Continue**: This button will pause the current job. If the job is being rendered, the server will stop rendering the job. When the job is paused, the title of the button changes to **Continue**. When **Continue** is selected, the job will resume.

**Info...**: This button invokes the **Job Information** dialog for the current job, as shown in Figure 3.6.8.8.

**Remove...**: This button removes the active job from the queue. If the server is rendering the job, the rendering is aborted and all intermediate data is lost.

**Retrieve...**: This button retrieves a completed rendering or animation of the active job of the list from the render server. The standard file save dialog is invoked for saving the file on the local machine. The format of the file is determined by the selection from the **Format** menu in the dialog. If the job is static rendering, the **Format** menu contains all of the applicable 2D image formats. If the job is an animation, the menu only contains the **form•Z** Animation (.fan) file format.

**View...**: This button works similar to the View... button in the Imager window and displays the most recently generated image. This item is only available when a completed rendering has been retrieved from the Render Server.

**Stop/Start**: This button stops the job queue. When the job queue is stopped, rendering jobs that are created accumulate in the queue, but they are not sent to the server. When the queue is stopped, the title of the button changes to **Start**. When **Start** is selected, the queue is started and any pending renderings are sent to the server.

**Settings...**: This button invokes the **Network Rendering Settings** dialog described previously.

**Log...**: Selecting this button invokes the **Network Render Log** dialog. This dialog contains the contents of the **form•Z** application log file that details the actions of the **form•Z** application. This information can be used to diagnose problems with the Network Render. The log file is stored on disk in the text file “formZ Network Render Log.txt”. The **Clear** button clears the content of the log file.
The Job Information dialog

This dialog shows information about the job, as follows.

**Name**: The name of the job.

**Server**: The server the job is on.

**User**: The name and IP address of the user who submitted the job.

**Status**: The current status for the job. This can be sending/receiving files, queued, rendering, completed or retrieved.

**Progress**: The rendering progress, which is 0% while the job is pending in the queue, and 100% when it is completed.

**Number Of Clients**: If the job is currently being rendered, this shows the number of clients rendering the job.

**Renderer**: This indicates which renderer (Surface Rendering, RenderZone, etc.) is being used for the job. This field also indicates if the job is processing a static image or an animation.

**Added To Queue**: The date and time the job was added to the queue.

**Start Render Time**: The date and time the job started rendering.

**End Render Time**: The date and time the job completed rendering.

**Total Render Time**: The total time the job took to render. This is measured from the time that the job's status changes from pending to rendering and from rendering to complete.

**Last Retrieved By**: The name and IP address of the user who last retrieved the job.

**Last Retrieved Time**: The date and time when the job was last retrieved.

**File Retrieval**: Displays whether the job is set for manual or automatic file retrieval. If automatic, the format of the output file is displayed.

**Assigned Work**: This field shows the work assigned to the job. If the job is an animation, this field displays the range of frames in the animation. If the job is a static image, this field displays the number of bands in the image.

**Disk Space**: This field displays the amount of disk space the job is using on the Render Server. For jobs that are part of a project, this number may include files that are shared with other jobs in the project.

*Figure 3.6.8.8*: The Job Information dialog.
**Priority:** This is the priority for the job and can range from the highest available priority for the user who has created the job to the low priority of 99. The priority for the job can be changed anytime until the job has finished rendering. If changing the priority for a job causes another job to have a higher priority, the current job will stop rendering and the new higher priority job will start rendering.

**Results...:** This button invokes the Results dialog shown in Figure 3.6.8.9. This dialog shows a list of which clients worked on the job and what part and for how long they worked.

**The form•Z Render Server and its window**

The **form•Z Render Server** is the network rendering component that performs the management of the rendering. The Server receives rendering jobs and content from the **form•Z** application, distributes the rendering to the clients that are connected to it, composites the image or animation, and returns the completed image or animation to the **form•Z** application machine.

The **form•Z Render Server** interface consists of a few standard menu items and an interface window. The enabled menu items are **Quit** from the **File** menu, and **form•Z Web Site...**, **form•Z Web Support...**, and **email Tech Support...** in the **Help** menu. The main part of the render server interface is the **form•Z Render Server** window, shown in Figure 3.6.8.10. This window is used to control the render server and displays the activity and progress of the server.

**Server IP:** This is the IP address and port of the machine the server is running on. The IP address and network name is determined by the operating system’s network settings. The port is set in the **Server Settings** dialog as described later in this section.

**Users:** This list shows all of the registered users for the server. The highlighted user is the current user.

- **Total:** The total number of users in the user list is entered in this field.

- **Add...:** This button is used to add a user to the list. The **User Settings** dialog is invoked for entering the user’s information, as described in the next subsection.

- **Edit...:** This item is used to edit the current user in the user list. The **User Settings** dialog is invoked for entering the user’s information as described in the next subsection.

- **Remove:** This button removes the current user from the list. If the user has jobs in the queue that are being rendered, they are also removed. This button will not remove the guest account.
**Figure 3.6.8.10:** The *form•Z Render Server* window.

**Job Queue**: This list shows all of the *form•Z* jobs that the server is processing. The highlighted job is the current job. The following buttons operate on the current job.

- **Total**: The total number of jobs in the **Job Queue** list is entered in this field.

- **Pause/Continue**: This pauses the current job. All of the clients that are processing the job stop processing and the server collects any intermediate data that can be preserved. When the job is paused, the title of the button changes to **Continue**. When the latter is selected, the job is changed to pending and its processing continues.

- **Info...**: This button invokes the **Job Information** dialog for the current job as shown in Figure 3.6.8.8. This dialog was discussed earlier in this section.

- **Remove**: This button removes the current job from the list. If the job is rendering, it is stopped before being removed.
Clients: This list shows all of the form-Z render clients that are connected to the server. The highlighted client is the current client. The following buttons operate on the current client.

Total: The total number of clients in the list is entered in this field.

Stop/Start: This button stops the client. When the client is stopped it does not perform any rendering and the server collects any intermediate data that can be preserved. When the client is stopped, the title of the button changes to Start. When it is selected in this state, the client resumes processing.

Info...: This button invokes the Client Information dialog for the current client, as shown in Figure 3.6.8.11.

Remove: This button removes the client from the list. If the client is rendering, it is stopped before being removed.

Configure...: This button invokes the Server Configuration dialog, which is shown and discussed in the next section.

Stop/Start: This button stops the render server. When the server is stopped, new rendering jobs can not be sent to the server and the rendering clients can not communicate with the server. When the server is stopped, the title of the button changes to Start. When it is selected in this state, the server resumes processing. It is not recommended to stop the server while it is rendering a job, as the clients will continue to try to contact the server and this could lead to network slow down.

Log...: Selecting this button invokes the Render Server Log dialog. This dialog contains the contents of the server’s log file that details the actions of the server. This information can be used to diagnose problems with the server. The log file is stored on disk in the text file “formZ Render Server Log.txt”. The Clear button clears the content of the log file.
The Client Information dialog

This dialog, shown in Figure 3.6.8.11, shows information about the client, as follows:

**Name**: The name of the client.

**Status**: The status of the client.

**Job Name**: The name of the job the client is working on.

**Progress**: The progress of the job on the client.

**Renderer**: This indicates which renderer (Surface Rendering, RenderZone, etc.) is being used for the job. This field also indicates if the job is processing a static image or an animation.

**Assigned Work**: This field shows the work assigned to the client. If the job is an animation, this field displays the range of frames in the animation. If the job is a static image, this field displays the number if bands in the image.

**Strength**: This slider can be adjusted from 0 to 100 with 100 being the strongest. Clients default to a strength of 50.

The User Settings dialog

This dialog, shown in Figure 3.6.8.12, is used for adding new users and editing existing users. It can be invoked in two ways: (1) by clicking on the **Add**... button in the Z Render Server window or (2) by clicking **Edit**... after selecting a non-guest user in the user list of the same dialog.

**Name**: This text field specifies the user name, which can be up to 64 characters long and must be unique.

**Password**: This text field specifies the user password. Passwords can be up to 64 characters long.

**Priority**: This slider determines the priority assigned to jobs that are sent to the server by the user. 1 is the highest priority and 99 is the lowest.
The Server Configuration dialog

This dialog, shown in Figure 3.6.8.13, contains the server configuration information. It is invoked by clicking on the Configure... button in the form•Z Render Server window.

**Port:** This text field specifies the TCP/IP port that the server is operating on. The default is 28838. This number should not be changed unless you encounter a collision with this port on the network.

**Clients:** This group of options specifies the client connection settings.

**Check Client Every \( n \) Seconds:** This text field determines how often the server will check on the client’s progress (in seconds.) Increasing this number increases the performance at the expense of accurate information in the Clients section of the Render Server window.

**Disconnect Client After \( n \) Missed Checks:** This number controls how tolerant the server is to a client which is not responding to server queries. This can be caused by network problems or problems with the client machine. A client that is not responding properly can slow down the overall rendering process.

**Ban Client After \( n \) Crashes:** When this option is on, this number controls how tolerant the server is to a crashed client. A client that is not functioning properly can slow down the overall rendering process. Note that the server thinks that the client is crashed when it loses contact. Thus a network problem with a client is treated as a crash.

**Limit Jobs To \( n \) Clients:** When this option is on, the server will only split a job across the specified number of clients when there are other jobs in the queue. If there are more clients available than the current job can use, the additional jobs in the queue are rendered simultaneously. This setting can dramatically affect the performance of network rendering of static images. If a job is split between too many clients, the overhead of transferring data to the clients can be more than the actual rendering time.

**Allow Guests:** When this option is on, the server allows guests to connect to the server in addition to registered users. A guest login allows for anyone on the network with a copy of form•Z or the Imager to send jobs to the Render Server without a user specific login account. Any user signed in with a guest account can remove or retrieve any job submitted by any other guest user.

To more securely handle jobs, a registered user can be added to the Render Server that will require the users to enter a user name and password to access the jobs on the Render Server. Turning off Allow Guests will remove the guest account and any jobs on the server sent from a guest login.

**Guest Priority:** This slider determines the priority assigned to jobs that are sent to the server by guest users. 1 is the highest and 99 the lowest priority.
The form•Z Render Client and its window

The form•Z Render Client is the network rendering component that performs the actual rendering of the image or animation. The client is controlled by a form•Z Render Server. The server sends rendering content (form•Z project files, textures, etc.) and rendering commands to the client. The client renders the data and returns the rendered image data or animation back to the server.

The form•Z Render Client interface consists of a few standard menu items and an interface window. The enabled menu items are Quit from the File menu, and form•Z Web Site..., form•Z Web Support..., and email Tech Support... in the Help menu. The main part of the render client interface is the form•Z Render Client window, shown in Figure 3.6.8.14. This is used to control the render client and shows the activity and progress of the client.

Client Name: This is the name of the client computer. This name is used to identify the client in the server window.

Client IP: This is the IP address of the machine that the client is running on. This address is determined by the operating system’s TCP/IP settings. Consult your operating system’s user manual for specific details of how to control your TCP/IP address.

Server IP: This is the IP address of the current render server. This address is determined by settings in the Client Settings dialog as described below.

Connection Status: This is the status of the client’s connection with the server. This will read “Off Line” if the client is connected to the server and “Connected” when it is connected. When the client is connected to the server, the server is able to communicate with the client. When the client is not connected, the server does not know that the client exists.

Job: This is the name of the job being rendered.

Figure 3.6.8.14: The form•Z Render Client window.
**Job Status:** This displays the task that the client is currently performing. This will be blank when the client is not connected to a server. When it is connected, it will commonly show one of the following:

- **Idle:** Waiting for the server to send instructions.
- **Receiving:** Receiving data from the server.
- **Rendering:** Rendering the data.
- **Sending:** Sending completed data to the server.
- **Loading:** Loading the project.

**Progress:** This is the progress of the current rendering. Since the server divides the rendering among the available clients, this progress is just for the small portion that the server has asked the client to render. This also displays the progress of sending and receiving files from the server.

**Settings...** This button invokes the **Client Settings** dialog (see Figure 3.6.8.14).

**Connect/Disconnect:** This button connects or disconnects the client to the current render server. This button reads **Connect** when the server is not connected and **Disconnect** when it is connected. If you disconnect from the server while the client is rendering, the current rendering is stopped and any completed image data is sent to the server. Incomplete render data is discarded.

**Stop/Start:** This button stops the client’s actions. When the client is stopped, the server will retrieve any intermediate rendered files. After selecting this button the title changes to **Start.** Selecting the button in this state resumes the client process.

**Log...**: Selecting this button invokes the **Render Client Log** dialog. This dialog contains the contents of the client’s log file that details the actions of the client. This information can be used to diagnose problems with the client. The log file is stored on disk in the text file “formZ Render Client Log.txt”. The **Clear** button clears the content of the log file.
The Client Settings dialog

This dialog, shown in Figure 3.6.8.15, maintains the list of servers that the client can use for network rendering. It also contains settings the client uses when connecting to the server.

**Servers**: This list contains a list of all of the servers that the client is aware of. The current server is highlighted in the list.

**Browse**: This button invokes the Browse for Server dialog (see Figure 3.6.8.16).

**Add...**: This item is used to add a server to the list. The Server Settings dialog is invoked for entering the TCP/IP address or machine name and port of the server to be added (see Figure 3.6.8.16).

**Edit...**: This button is used to edit the setting for the current server. The Server Settings dialog is invoked for editing the TCP/IP address or machine name and port of the server (see Figure 3.6.8.15.)

**Remove...**: This item is used to remove the current server from the list.

**Synchronize Time To Server**: When this option is selected, the date and time of the client machine is changed to match the server’s time when the client connects to the server.

**Port**: This group of options allows you to designate which port will be used.

**Static**: When this is selected, form•Z Render Client will use the port number entered in the text field to the right of this option as the port number. Care must be taken that this port number is not being used by any other application on the same computer. The default static port number is 28836.

**Dynamic**: When this is selected, form•Z Render Client will automatically use the free port number that the Operating System selects. This is the default option.

**Client Strength**: This slider can be adjusted from 0 to 100 with 100 being the strongest. Clients default to a strength of 50.

**Remove Job Files...**: This button will remove any files sent to the client by the server.
The Server Settings dialog

This dialog, shown in Figure 3.6.8.16, is used to add or edit a server in the client’s server list.

**Server**: This is the TCP/IP address or Machine name of the machine on which the server is running as shown at the top of the render server window. Note that Machine names will only work under the following conditions:

- On any platform if the server machine name is registered with a DNS server.
- On windows if a WINS server is running in the servers domain.

**Server Port**: This is the TCP/IP port that the server is running on. This must be the same as the port set in the **Server Configuration** dialog of the render server (see earlier in this section.) The default is 28838.

The Browse For Server dialog

This dialog, shown in Figure 3.6.8.17, shows a list of all servers found on the network by browsing. When the dialog is opened, the browser starts searching for servers. As each server is found, it is added to the server list. At any time you can select a server or close the dialog without waiting for the search to finish.

**Servers**: This list contains the names of all of the servers found on the network by browsing. The current server is highlighted in the list.

**Refresh**: This button is used to rebrowse the network for servers.

**Add...**: This button is used to add the current server to the list of known servers.

**Add All...**: This button is used to add all the servers found by browsing to the list of known servers.
3.6.9 Display and rendering options

Display Options...

This item invokes the **Modeling Display Options** dialog (Figure 3.6.9.1) if the active window is a modeling window. If it is a drafting window, it invokes the **Drafting Display Options** dialog, discussed in section 3.6.13 (Figure 3.6.13.1). This dialog mostly contains button commands that invoke additional dialogs, one for each of the plotting/rendering modes in form-Z. These are the same dialogs invoked by the respective items in the **Display** menu and have been discussed earlier in this section. The buttons are organized into separate sections for interactive and static renderers.

*Figure 3.6.9.1: The *Modeling Display Options* dialog.*
3.6.10 QuickTime VR movies

A form•Z scene can be saved as a QuickTime VR™ (QTVR) movie by selecting the Generate QuickTime VR... item in the Display menu (Figure 3.6.0.1). These movies can be played by Apple’s MoviePlayer, or any other application that supports QTVR. You can instruct form•Z to invoke the QTVR player as soon as it completes the generation of a VR movie. Or you can invoke the QTVR player from View File... (File menu) and by selecting a QTVR movie file.

To create and save a QTVR movie you need QuickTime version 2.0 or later on Macintosh and 3.0 or later on Windows installed on your machine. QuickTime is distributed with the Macintosh system software. Installers for Quicktime for both Macintosh and Windows can be found at www.QuickTime.com.

QuickTime VR offers three types of movies: cylindrical panoramic, spherical panoramic and object movies. A cylindrical panoramic movie consists of a single rendering produced by placing the viewer in the middle of a scene, looking outwards. The image used for the panoramic movie roughly corresponds to the inside of the curved surface of an imaginary cylinder, after it has been unfolded. In form•Z, a cylindrical panoramic image can also be rendered and saved as an independent image, by selecting Panoramic* in the Views menu (see section 3.5.3). When saving a cylindrical panoramic movie, you can either use a panoramic image already rendered, or you can tell the program to generate an image at the time Generate QuickTime VR... is executed. When a panoramic movie is played back by a QTVR player, a portion of the panorama is displayed and mapped onto a regular perspective view. Dragging the mouse left or right, interactively changes the view, as if you were spinning around an axis. A spherical panoramic movie is similar to a cylindrical movie, except that six images are used. Unlike with a cylindrical movie, spherical panoramas do not exist as a view type in form•Z and must always be generated at the time the movie is exported.

A QTVR object movie stores images of a scene rendered from a number of viewing positions, located on an imaginary sphere that surrounds the scene, and looking towards the center of the sphere. Note that there is no way to meaningfully unfold the object movie images into a single image. Consequently, there is no command to generate such an image independently and the object movie images are always rendered at the time the file is generated. When an object movie is played back, QTVR selects the appropriate image based on which direction the mouse is dragged. This gives the impression of rotating the scene around an axis.

Selecting Generate QuickTime VR... invokes the QuickTime VR Movie Options dialog, shown in Figure 3.6.10.1.
 Saving a cylindrical panoramic movie

You can save a cylindrical panoramic movie by selecting the **Generate Cylindrical Panoramic Movie** option in the **QuickTime VR Movie Options** dialog. A cylindrical panoramic movie can be saved and exported by either using a panoramic view already rendered or by generating a panoramic rendering at the time the **Generate QuickTime VR...** command is executed.

**Use Current Panoramic View:** When this option is selected, the panoramic image currently displayed in the active window is used to create the panoramic movie. If the view in the active window is not a panoramic view, this option is not available.

**Generate New Panoramic View:** When this option is selected, a new panoramic view of the scene in the active project will be rendered. The rendering type, image size, and panoramic view parameters are determined by options set in the **QuickTime VR Movie Options** dialog.

**Display Type:** This pop up menu allows you to select one of the available display types.

**Image Resolution:** The width and height of the panoramic image are entered in these text edit fields.

**Panoramic View Parameters...:** Clicking on this button invokes the **View Parameters** dialog with the **View Type** option set to **Panoramic**. This is the same dialog used by the **Panoramic** item in the **Views** menu. Its is discussed in section 3.5.3.
Saving a spherical panoramic movie

A spherical panoramic movie is similar to the cylindrical panoramic movie. In addition to rotating the camera horizontally, as is done for the cylindrical movie, in a spherical movie the camera can also be tilted vertically, which allows it to reach each spot in the environment. One can think of a cylindrical panorama as a camera mounted on a tripod that only has one axis of rotation, the vertical axis. With a spherical panorama the camera would be mounted with a fully swiveling connection. Note, that, unlike cylindrical panoramic movies, spherical movies cannot be generated from an existing panoramic view, but are created like object movies, and the required images can only be generated at export time.

To be able to play back spherical panoramic movies, Quicktime version 5.0 needs to be installed. While earlier versions of Quicktime are able to play those movies, they can only show them as standard (cylindrical) panoramic movies.

You save a spherical panoramic movie by selecting the **Generate Spherical Panoramic Movie** option in the **QuickTime VR Movie Options** dialog shown in Figure 3.6.10.1. The **Panoramic View Parameters**… button invokes the **Spherical Panoramic Movie View Parameters** dialog, shown in Figure 3.6.10.2.

**Eye Point**: This category contains the coordinate values of the camera’s position. Initially, these values are the world origin. By pressing the **Center Of Scene** button below, the center of the scene’s bounding box can be copied into the fields. By pressing the **Current** button, the eye point of the current view is copied in the fields.

**Use Complete Field Of View**: When this option is selected, the movie is generated with a 360 degree horizontal (pan) angle and a 180 degree (tilt) angle, covering the entire scene.

**Custom Field Of View**: When this option is selected, custom pan and tilt angles can be selected. The pan angles are measured from the positive y axis (12 o’clock). The tilt angles are measured from the xy plane.
Saving an object movie

You can save an object movie by selecting the Generate Object Movie option in the Quick-Time VR Options dialog. In contrast to the panoramic movie, an object movie can only be generated at the time it is saved and exported.

**Display Type:** One of the available display types is selected from this pop up menu.

**View Type:** This menu allows you to select a view type: Axonometric or Perspective.

**Image Resolution:** The width and height of the object movie images are entered in these text fields. Note, that these values also indicate the dimensions at which the object movie is first presented when played back by a QuickTime VR application.

**Move Lights With View:** When this option is on, all lights are rotated with the eye point around the center of the scene. When the object movie is played back, this gives the impression of the objects moving, while the lights stay fixed. In other words, the hotspots on curved and shiny objects move across the objects. When this option is off, the hotspots stay at the same positions.

**Object View Parameters:**

Clicking on this button invokes the Object Movie View Parameters dialog shown in Figure 3.6.10.3. It contains the parameters that affect the generation of an object movie.

An object movie consists of individual images rendered by looking at the scene from a sphere around the scene. To generate the images the view is rotated a number of times horizontally around the sphere’s vertical axis. This is called **panning.** After the horizontal rotations, the view is rotated vertically about the sphere’s horizontal axis, which is referred to as **tilting.** A new set of panned images is then taken from the new tilted position. This is repeated until all rotations are completed.

In order to render a complete movie of a scene, the view starts at the north pole of the sphere, looking straight down the vertical axis. After each set of horizontal rotations covering 360 degrees, the vertical tilt rotates the view a number of steps for a total of 180 degrees, until the last set of images is taken from the south pole looking straight up the vertical axis. This process is illustrated in Figure 3.6.10.4.
**Figure 3.6.10.4:** Object movie with

# Of Pan Steps = # Of Tilt Steps = 8:

(a) 8 x 8 sphere; (b) the images, and (c) exploded view.

**Figure 3.6.10.5:** The pan and tilt angles.

- **# Of Pan Steps:** This sets how many times the view is rotated around the vertical axis.
- **# Of Tilt Steps:** This sets how many times the view is rotated around the horizontal axis.
- **Center Of Interest:** The coordinates in these X, Y, and Z fields indicate the center of interest for each image of the object movie.
- **Center of Scene:** Clicking on this button causes the coordinate values of the center of the scene's bounding box to be entered in the Center Of Interest fields.
- **Current:** Clicking on this button causes the values of the center of interest of the current view in the active window to be entered in the Center Of Interest fields.
- **Rotate Relative To:** The vertical and horizontal axes of the sphere are set relative to the selected reference plane, as follows:
  - **XY Plane:** the vertical axis is the world Z axis and the horizontal axis is the world Y axis.
  - **YZ Plane:** the vertical axis is the world X axis and the horizontal axis is the world Z axis.
  - **ZX Plane:** the vertical axis is the world Y axis and the horizontal axis is the world X axis.
  - **Active Plane:** vertical axis is the Z axis and horizontal axis is the Y axis of the active reference plane, which may be an arbitrary plane or one of the Cartesian (XY, YZ or ZX) planes.
- **Use Complete Field Of View:** When this option is selected, the object movie is generated with a 360 degree pan angle and a 180 degree tilt angle, covering the entire scene.
- **Custom Field Of View:** When this option is selected, start and end angles can be entered for pan and tilt. Note that the pan angle is measured relative to the axis which is perpendicular to the horizontal axis of the sphere, whereas the tilt angle is measured relative to the equator of the sphere, as illustrated in Figure 3.6.10.5.
- **Zoom To Fit Scene:** When this option is selected, the view for all renderings is adjusted so that all objects in the scene will be visible in all images.
- **Use Current Zoom:** When this option is selected, the zoom factor of the current view in the active window is used for all renderings.
Sizing QuickTime VR movies

The following parameters are used to set the size of the movie window.

Movie Size: The values in pixels entered in these fields represent the dimensions of the window, in which the panoramic movie is first shown, when played back by a QuickTime VR application. For best results, the larger dimension should not exceed the smaller dimension of the panoramic image resolution.

Viewing QuickTime VR movies

You can instruct form-Z to play back the movie just saved by selecting the View Movie When Complete option in the QuickTime VR Movie Options dialog. With this option on, as soon as the generation of the movie file is completed, the standard QuickTime VR player window is invoked, showing the default view of the movie. You can also play a QTVR movie using the View File... item in the File menu and selecting a movie file from the File Open dialog that is invoked.

For both panoramic and object movies, the QTVR player initially displays one frame or portion of the movie. For example, the image displayed in Figure 3.6.10.6 is only a portion of the complete panoramic image shown in Figure 3.6.10.7. You can navigate through the movie as follows:

Panoramic movie: Pressing and dragging the mouse
- right, spins the view clockwise;
- left, spins the view counterclockwise;
- up, tilts the view up;
- down, tilts the view down.

Figure 3.6.10.6: The QTVR window.

Figure 3.6.10.7: A QuickTime VR panoramic image.
Menus

• The Display menu

Note, that a view can only be tilted in a cylindrical panorama if it has been zoomed in.

**Object** movie: Pressing and dragging the mouse
- sideways, spins the scene around the vertical axis,
- up and down, spins around the horizontal axis.

For both panoramic and object movies, you can zoom in and out using the Zoom tools located at the lower margin of the view player window, or using modifier keys: the **shift** key zooms in, while the **control** key zooms out.

**Compressing QuickTime VR movies**

QuickTime offers a number of compression algorithms, which may be selected when saving a movie.

**Compression Options...**: Clicking on this button in the **QuickTime VR Movie Options** dialog, invokes the **Compression Settings** dialog shown in Figure 3.6.10.8. Note that this is a standard dialog provided by Apple and may look different depending on which version of QuickTime you are using.

**Compressor**: The two pop up menus in this box allow you to select a compression method and its parameters. The compression method is selected from the top menu, whose content is as shown in Figure 3.6.10.8(b). The content of the second pop up menu changes, depending on the selection in the first menu. The second pop up menu contains the parametric variations available for each compression method. Note that different versions of QuickTime may have different compression methods available.

**Quality**: A quality level, expressed as a percentage, can be selected from this slide bar.

The combination of compression, color depth and quality determines how much disk space is consumed when saving the movie file. In general, the better the image quality and the greater the color depth, the more disk space will be required. A preview image in the upper right corner shows the effect of the selected options on a sample image.

*Figure 3.6.10.8:*  
The **Compression Settings** dialog.
3.6.11 Animations, rendering memory, and showing surfaces

The eighth group of the Display menu contains three animation related commands, namely, Generate QuickTime VR..., Generate Animation..., and Play Animation.... The first item is discussed in section 3.6.10, the latter two are discussed in Chapter 7.

The ninth group of the Display menu contains two items that determine when the rendering memory is cleared and one that controls how surface objects are displayed.

**Clear Rendering Memory**...

Selection of this item clears the memory occupied by rendering information that may have been saved from previously executed rendering calculations.

**Always Clear Rendering Memory**...

This item is a switch. When selected, a check mark is displayed in front of it, causing the program to always clear the calculations that resulted from a rendering. As a result, each new rendering will be calculated from the beginning. You want to turn this item on when you do not expect to render different views of the same model and your memory is tight. This option is off by default.

**Show Surfaces As Double Sided**

This item is a switch when selected, both sides of one sided surface objects are visible in any rendering mode. This option is on by default.
3.6.12 Setting image parameters

**Image Options...**

Selecting this item invokes the **Image Options** dialog (Figure 3.6.12.1). This dialog contains options controlling the size and color depth of rendered images.

**Image Size**: One of two options to define the size of the rendered image may be selected from this section. The image size can be specified by using the size of the graphics window where the image is rendered (default), or by using a custom size.

**Use Window Size**: Selection of this option sets the image size to the dimensions of the project window. Upon selection of this option, the **Width** and **Height** items are updated to display the window dimensions. This option is the default.

**Use Custom Size**: This option allows customized specification of the image size. One of two methods may be selected.

**By Number Of Pixels**: When this option is selected, one of thirteen preset image sizes may be selected from the pop up menu shown in Figure 3.6.12.2, or the desired size may be entered directly in the **Width** and **Height** text fields. These dimensions are expressed in pixels. The largest numbers that can be used for width and height of windows are 32,767 pixels.

**By Size and Resolution**: When this option is selected, the desired size, in inches, and the desired resolution, in pixels per inch, is entered in the appropriate **Width**, **Height**, and **Resolution** fields.

**Maintain Proportions**: When this option is selected, the proportions of the project window are maintained in the rendered image. When selected, if the **Width** or **Height** dimension is changed, the other is automatically adjusted to maintain the existing proportions. This option is dimmed and inactive when the **Use Custom Size** option is not selected.
The Display menu

**Use Screen Size:** Clicking on this button sets the values in the *Width* and *Height* fields to the monitor’s screen size.

When the **Use Custom Size** option is selected, in most cases, the window will be automatically resized when you exit the **Image Options** dialog, as follows. In addition:

- The proportions of the window are always adjusted to the proportions of the image.
- If the image size is less than the screen size, the window is resized to the image’s size.
- If the image size is greater than the screen, only the proportions of the window will be adjusted. If the **Maximize Window In Screen** option is also selected (see below), then the largest possible window size that fits in the screen will be used.
- Resizing the window using its resize box only allows you to decrease the window size from its indicated size.

**Maximize Window In Screen:** This option is active when **Use Custom Size** is on. When selected and the size of the image is greater than the size of the screen, the window takes the largest possible size that fits in the screen.

**Image Color Depth:** One of three options may be selected from this pop up menu (Figure 3.6.12.3) to define the color depth of the rendered image. These three options reflect the possible color depths of the monitor. The current color depth of the monitor is selected by default, and is set through the standard Monitor control panel or system preferences (see your operating system’s Manual for details).

![Maintain Prop] 8 Bit (256)
16 Bit (Thousands)
✓ 32 Bit (Millions)

*Figure 3.6.12.3:* The **Image Color Depth** pop up menu.
3.6.13 Plotting in drafting

A single display method suffices for plotting the drafting elements. It is provided by the Replot* command that appears in the place of Wire Frame* when a drafting window is active. In the drafting, the other display commands are dimmed and inactive.

Replot*

This is the only method available for plotting drafting elements. When clicking on it while pressing the option key (Macintosh) or ctrl+shift (Windows), the Drafting Display Options dialog, shown in Figure 3.6.13.1 is invoked. This dialog consists of six sections that let you select how the colors, line styles, line weights, text, arcs, image elements, and symbols will be plotted.

Unless otherwise noted, the options in the Drafting Display Options dialog affect both how lines are displayed on the screen, and how they are printed on printers or plotted on plotters. In the following discussion the term “plot” is used to represent all these cases. Note, however, that printed drawings are affected by the options selected in the Print dialog, and also depend on specific plotter/printer devices and their drivers.

Color Display: The item selected from this pop up menu determines what colors will be used when drafting elements are plotted. Recall that, in form-Z, colors are assigned to individual elements, and that the line weight attributes assigned to them also include a color, which may generally be different from that assigned to the element. You can choose to plot a drawing using one or the other or none.

Black/White: When this option is selected, all elements are drawn in a uniform black or white color, depending on whether the background color is light or dark, respectively.

Color Of Element: When this option is selected, each element is plotted using the color assigned to it.

Color Of Line Weight: When this option is selected, the elements are plotted using the colors of the line weights assigned to them.

Figure 3.6.13.1: (a) The Drafting Display Options dialog with the Lines tab, (b) the Text tab, and (c) the Elements tab.
Hide Ghosted: When this option is on, ghosted entities are not shown. They are shown in the project’s ghost color (grey by default), when this option is off. This option can be applied to either Elements or Layers or both, depending on the selection from the two options under it.

Lines tab: This group of options determines how lines are plotted.

Line Quality: One of five items can be selected from this pop up menu to determine the level of plotting quality. The items are listed in order from worst to best. The higher the quality, the longer it takes to plot a drawing. The available line plotting qualities are in three categories:

- The first category contains a single level of quality, **Solid**, which ignores the line styles assigned to the elements and produces only continuous lines.
- The second category contains three levels labeled **Sketch**, **Good**, and **Better**. They all use the line styles assigned to the elements and apply different levels of pattern justification. They all draw lines as lines by relying on the lines supported by the operating system.
- The third category consists of a single level of line quality, called **Best**. This also uses the assigned line styles, but plots the lines as filled polygons. That is, the boundaries of the dash segments are calculated by form.Z and then plotted as filled polygons. The main advantage of this method is that the ends of lines and dashes are perpendicular to the lines, as opposed to being parallel to the axes of the screen. This difference in edge quality becomes particularly apparent with relatively heavy diagonal lines, as illustrated in Figure 3.6.13.2(f), (g), (h), and (i).

**Solid**: When this option is selected, all lines are drawn and printed as continuous lines, regardless of the line styles that are assigned to the elements. This is the fastest line plotting method (Figure 3.6.13.2(a)).

**Sketch**: When this option is selected, all elements are plotted using the line styles assigned to them. Dashed lines are produced by starting at the first point of a shape, applying the style’s pattern module sequentially, and wrapping it around the shape without making adjustments. The pattern module placed last will usually be incomplete (Figure 3.6.13.2(b)).

**Good**: This method plots dashed lines by applying different adjustments for open and closed shapes:

- For open shapes, it starts the generation of the dashed line at the midpoint of the shape, and repeats the dash module by moving in both directions. This results in a symmetric dashed line, since the pattern module at the two ends of the open shape are of the same size.
- For closed shapes, the size of the pattern module is adjusted to the size of the shape, resulting in a round number of modules, which fit exactly on the shape. This produces a smooth continuity at the closing point of a shape (Figure 3.6.13.2(c)).

**Better**: This method plots dashed lines by making two adjustments: the line styles are centered on each segment, and a dash is always placed at the corner points of the shape. When this method is used, the **Wrap Corners** option also becomes available (Figure 3.6.13.2(d)).
Figure 3.6.13.2: The Line Quality options.
**Best:** This method plots lines by precisely calculating their boundaries, and then plotting them as filled polygons rather than as lines. It does not use the line drawing procedures supported by the operating system, but calculates lines and dashes as closed polygons which are then filled, resulting in properly oriented ends (Figure 3.6.13.2(f)). When this method is used, the **Wrap Corners**, **Bevel Corners**, and the **Show Axis Lines** options are also available.

**Wrap Corners Greater Than $n$:** This option is available when **Line Quality** is set to **Better** or **Best**. When it is off, dash lines are always placed at the corner points of shape, and, when shapes contain small segments, these segments may appear as continuous lines. When it is on, corner points of an angle greater than $n$ are exempted from having dashes placed on them. Instead, they are treated as continuous segments, and the dash pattern is wrapped around them. This option is particularly useful for drawing circles, arcs, ellipses and smoothly curved lines. Examples are shown in Figure 3.6.13.2(g).

**Bevel Corners Less Than $n$:** This option is available when **Line Quality** is set to **Best**, which calculates the polygons of dashes by generating parallel lines, and then finding their points of intersection. At acute angles, this method produces spikes which are geometrically correct but may appear improper. Selection of this option bevels the corners of angles of less than $n$ degrees. This option is particularly useful when significantly heavy lines are used. Examples are shown in Figure 3.6.13.2(h).

**Show Axis Lines:** This option is available when **Line Quality** is set to **Best**. When selected and the line weight exceeds two pixels, a thin axial line is also drawn. Especially when working with thick lines, the display of the axial lines facilitates reading the corner points and the segments of a shape. The examples shown in Figure 3.6.13.2(i) were plotted with all three options (**Wrap Corners**, **Bevel Corners**, and **Show Axis**) on.

**Line Weight Display:** This group of options affects only how the lines are displayed on the screen. It has no effect on how the lines will be printed on a printer or a plotter, which is determined by the line styles and the line weights assigned to them.

**Show Line Weight:** When this option is on, the weights of the lines are drawn as specified by the line weights assigned to them. When off, all lines are drawn with a one pixel width.

**Line Size:** When **Show Line Weight** is on, the item selected from this pop up menu determines whether lines will be drawn according to their **Screen Size** or **Print Size**. When **Screen Size** is used, the **Scale To Zoom Scale** option also becomes available. Selection of **Print Size** offers a preview of how a drawing will be printed.

**Scale To Zoom Scale:** This option is available only when **Screen Size** is selected from the **Line Size** menu. When on, the widths of lines are scaled according to the zoom scale of the graphics window. When off, the widths are displayed as specified in the light weights assigned to the elements at all scales.
Text tab: The options in this tab determine how text is plotted in drafting. There are three ways to display text: as outline, stick font, or bounding box. These and other options affect screen refresh times.

Outline: The three items selected from this popup menu control the color in which the outline of the text will be displayed, as follows:

Off: When this item is selected, no outline is drawn.

Black/White: With this item on, the text is drawn black on a light screen background, or white on a dark background.

Color: With this item selected, which is the default, the color assigned to text is used to draw it.

Fill: The three items in this popup menu, which are the same with Outline, control the color with which outline fonts will be filled. Default is Off.

Other than the color of its outlines and fills, there are four mutually exclusive ways in which text in drafting can be displayed, as follows:

Use Text Element’s Smoothness: When this option is on, which is the default, text is displayed with its actual font and in its actual resolution.

Use Stick Font: Selecting this option causes form•Z to convert the font(s) currently selected for display into form•Z’s stick font. This is a “midway” alternative to the slower refresh times of the outline text generated from TrueType, Postscript, and Bitmap fonts, and the fast refresh but illegible bounding boxes (see below). The stick font preserves text element legibility and speeds up screen refreshes, while sacrificing smoothness, double outlines, and fill. The stick font will retain the proportions of the selected outline font to ensure space continuity.

Bounding Box: Selecting this option causes form•Z to convert the font(s) currently selected for display to bounding boxes. The bounding boxes provide the fastest screen refresh, but the text elements they represent are illegible. As with the stick fonts, the bounding boxes retain the proportion of the text. They are also drawn with their diagonals to easily distinguish them from any rectangular elements that may exist in the project space.

Override Text Smoothness: When this option is active, all the text elements are displayed with a smoothness that can be controlled through the slider bar below the option. The smoothness slider bar functions as with its counterpart in the Text Editor dialog.

Bounding Box Below n Pixels: This option forces all text elements with a display size smaller than n pixels to be displayed as bounding boxes. Using this option avoids the longer screen refresh time of text elements that are illegible because of their small size. However, as opposed to the Bounding Box option, this option displays text elements with a size greater than that specified in the numeric field.
**Elements** tab: The options in this tab determine how elements are drawn and displayed in drafting.

**Circle, Arc And Ellipse Display:** These mutually exclusive three options control the resolution with which arc elements are drawn.

**Use Element’s Resolution:** When this option is on, drafting elements of the draft type (circles, ellipses, angular dimensions, arc dimensions, and open arcs) will be drawn at a resolution corresponding to the number of points that was defined at the time they were created.

**By Resolution:** When this option is on, the resolution stored with the arc element is ignored and is overridden by the value entered in the field of this option.

**By Size:** Similar to the above, when this option is on, the resolution of the arc is ignored and is overridden. The resolution it is drawn with is calculated from the value entered in the field of this option. This value is a size and the total length of the arc is subdivided by this value to determine the total number of points the arc will be drawn with. Thus the resolution varies with the size of the arc.

**Image Element Options:** The popup menu in this group controls how image elements are shown in a drafting drawing.

**Detail:** Three items can be selected from this popup menu, as follows:

**Bounding Box:** When this item in selected, no image is displayed, but a rectangle with an X in it is shown instead.

**Preview:** When this item is selected, the low resolution preview image stored with an image element is displayed. This is the default.

**High Resolution:** When this item is on, the image with its full resolution is displayed.

**Dimension Display:** The one option in this group controls whether the points of non-associative dimensions are displayed in a drawing. Because non-associative dimensions do not reference actual points in a drawing, it may be difficult to identify the placement points. This option makes it possible to show them, so that they may be selected for operations such as the geometric transformations.

**Show Non Associative Points:** When this option is on, the placement points of non-associative dimensions are displayed and can be picked. This option is off by default.

**Show Symbol Instances as Bounding Box:** When this option is selected, symbols are shown as rectangles on the screen rather than the detailed symbol. This option can be used to improve plotting performance on big files.
3.7 The Options menu

This menu, shown in Figure 3.7.0.1, contains items that invoke dialogs affecting project level settings. Many of these dialogs can also be invoked from other tools or palettes.

- The first group, for both modeling and drafting, contains items relating to layers, project colors, symbol libraries, and working units.
- The second group is different for modeling and drafting. For modeling it contains items that invoke dialogs with options affecting animations, lights, macros, objects, profiles, reference planes, status of objects, surface styles, attributes, and information management. For drafting, it contains items that invoke options for colors, line styles, and line weights.

Figure 3.7.0.1: The Options menu in (a) modeling and (b) drafting.
3.7.1 Project level options

Layers...

This item invokes the Layers dialog that complements the Layers palette and can also be invoked by option (Macintosh) or shift+ctrl (Windows) clicking in that palette. Its structure and functions are discussed in subsection 2.7.2.

Project Colors...

This item invokes the Project Colors dialog, shown in Figure 3.7.1.1. This dialog contains a list of different system features that use color to enhance their readability. The colors currently assigned to these features are shown in boxes next to them and can be changed. To change a system color click in its box, which invokes the Color Picker dialog, where another color can be defined in the usual manner.

When Pasting Between Projects:
This pop up menu allows you to instruct the program what to do with the color of an object when it is transferred (pasted) from another project.

Use Active Color: When this item is selected, the active color in the importing project will be used, and no color will be transferred with the object.

Use Best Matching Color: When this option is selected, the current color of the object or element will be taken into consideration but will not be transferred with the object. The color in the Colors palette of the importing project that is the closest to the original color of the object/element will be used instead and will be assigned to the incoming object or element.

Transfer Original Color: When this option is selected, the color of the incoming object or element will be added to the Colors palette of the importing project and the transferred entity will keep exactly the color it had in the original project. Note that, in this context, color is used to identify the surface style attribute in the modeling environment.

Symbol Libraries...

This item invokes the Symbol Libraries dialog that supports operations applying to symbol libraries and to symbol definitions. It is also invoked from the Symbols palette. The operations it supports are discussed in section 4.20.2.
Working Units...

This item invokes the **Project Working Units** dialog shown in Figure 3.7.1.2. It contains two tabs, **Numeric Options** and **Angle Options**, as well as options outside tabs. They allow you to select the size level of the default values, the type of measurement and its accuracy.

**Options outside of the tabs**

**Data Scale**: This pop-up menu contains four items: **Miniature (Watch)**, **Small (Lamp)**, **Medium (Building)**, and **Large (City)**. They represent four scales for the sets of default values used in the application dialogs.

Next in the dialog are two groups of options that determine the Unit Type and its subdivision formats.

**English, Metric**: These are the two available measurement units. Selecting one or the other sets all the values of the program to English or metric respectively. Each type is followed by its respective subdivisions, one of which can be selected as the base unit. The **English** units can be set to **Inches** or **Feet**. The **Metric** can be set to **Millimeters**, **Centimeters**, **Decimeters**, or **Meters**.

**Coordinates Palette Unit Type**, **Prompts Palette Unit Type**: These are two groups of options at the bottom of the **Project Working Units** dialog that allow you to set the units used in these palettes. These are the same options that can be controlled from the little boxes found on the two palettes and are labeled **A**, **W**, and **C**. They are discussed in section 2.4.1.
**Numeric Options** tab

**Numeric Accuracy**: This field represents the accuracy module used in the project for the numeric values entered through the keyboard or graphically. It determines the accuracy of the numeric and graphic input. The default value is 1/16" (or 0.001 m metric).

**Numeric Display Options**: This group of contains the formats which can be used for the display of numbers in the Coordinates and Prompts palettes.

- **Architectural, Engineering, Fractional, Decimal**: These four mutually exclusive buttons determine the display format of the numeric values. The default format is Architectural.
- **Number Of Decimal Places**: This value determines how many digits will be shown after the decimal point. The default is 3.
- **Include Trailing Zeros**: This option determines if trailing zeroes after the decimal will be shown. By default, this option is on.

**Angle Options** tab

**Angle Accuracy**: This field controls the accuracy of the angles. The default is 1°.

**Angle Display Options**: This group contains four mutually exclusive formats that can be used for the display of angles. The default is decimal numbers expressing degrees.

- **Positive Angle Direction**: This pair of options allows you to determine which direction (Clockwise or Counterclockwise) should be interpreted as positive. By default, counterclockwise is positive.
- **Location Of 0° Angle**: This group of options allows you to determine which orientation angles are measured from. The default is east or 3 o’clock.
- **Measurement Method**: An angle can be measured in two ways: one in the positive and one in the negative direction. The (absolute values of the) two angles add up to 360°. The options in this group determine how to read and interpret the angles. The Small and Large Angle options echo the small (less than 180°) and large (greater than 180°) angle, respectively. The Positive and Negative Angle options echo the positive and negative angles, regardless of size. Note that which direction is taken to be positive depends on the selection of the Positive Angle Direction option.
- **# Decimal Places**: This determines how many digits will be shown after the decimal point. The default is 2.
- **Include Trailing Zeros**: This option determines if trailing zeroes after the decimal will be shown. By default, this option is off.
3.7.2 Modeling options

Animation...
Clicking on this item invokes the Animation Options dialog, which is discussed in section 7.1.1.

Animation Manager...
Clicking on this item invokes the Animation Manager dialog, which is discussed in section 7.1.4.

Lights...
This item invokes the Lights dialog, shown in Figure 2.10.2.1, which can also be invoked directly from the Lights palette, which it supplements. Its functions and commands are discussed in subsection 2.10.2.

Macro Transformations...
This item invokes the Macro Transformations dialog, shown in Figure 4.23.4.1, which can also be invoked from any of the Macro Transformation tools (1 2 3). It is used to assign a macro to one of these tools, to create a new macro through keyboard input, or to revise an already defined macro (see 4.23.4).

Objects...
This item invokes the Objects dialog, shown in Figure 2.8.2.1, that supplements the Objects palette from which it can also be accessed. Its functions and commands are discussed in subsection 2.8.2.

Profiles...
This item invokes the Profiles dialog, shown in Figure 4.11.1.6, that supplements the Profiles palette from which it can also be invoked. Profiles are used in conjunction with the Deform tool. The functions of the Profiles dialog are discussed in subsection 4.11.1.

Reference Planes...
This item invokes the Reference Planes dialog, shown in Figure 2.13.1.2, that supplements the Planes palette from which it can also be accessed. Its functions and commands are discussed in subsection 2.13.1.

Status Of Objects...
This item invokes the Status Of Objects tab, shown in Figure 4.5.0.2. Its options are discussed in section 4.5.

Surface styles...
This item invokes the Surface Styles dialog (Figure 2.6.2.2) that supplements the Surface Styles palette, from which it can also be accessed. They are discussed in subsection 2.6.2.
**Attributes...**

Selecting this item invokes the **Attributes** dialog shown in Figure 3.7.4.1. Through this dialog new custom attributes can be defined and assigned to objects, as discussed in section 3.7.4.

**Information Management...**

Selecting this item invokes the **Information Management** dialog shown in Figure 3.7.5.1. Through this dialog a user can create new and edit existing information management (IM) records, as discussed in section 3.7.5.
3.7.3 Drafting options

Colors...

This item invokes the Colors dialog, shown in Figure 3.7.3.1, that can also be accessed from the Colors palette in the drafting environment. Its options are discussed in subsection 2.6.1.

Line Styles...

This item invokes the Line Styles dialog, shown in Figure 3.7.3.2, that can also be accessed from the Line Styles palette, discussed in subsection 5.15.1.

Line Weights...

This item invokes the Line Weights dialog, shown in Figure 3.7.3.3, that can also be accessed from the Line Weights palette, discussed in subsection 5.15.2.
3.7.4 User defined attributes

Relative to who the creator is, there are three types of attributes in form•Z: (1) those preset by the program itself, (2) those defined by plugins, and (3) the custom attributes that may be created by a user. The latter are defined through the Attributes dialog and are discussed in this section.

An attribute is composed of one or more fields. A field may be formatted to represent a number, fraction, distance, area, volume, angle, percentage, currency value, date, time, duration, Boolean value, string, or an item from a value list. Combining fields with different formats, a user can create an attribute definition that describes a complete set of data, which may be attached to objects. Once a user defined attribute is created, it behaves in the exact same fashion as form•Z or plugin created attributes. For example, when an object is copied, any user defined attributes assigned to the original object are transferred to the copy object. User defined attributes may also be created with the intent of producing information records, as discussed in more detail in the next section.

Clicking on the Attributes... item of the Options menu invokes the Attributes dialog shown in Figure 3.7.4.1. In this dialog, a user may define new custom attributes that can be assigned to objects.

The dialog is divided into two sections. At the top there is a list where all the currently defined attributes are displayed. It will be referred to as the attributes list. This list also contains attributes that are preset by form•Z and are always present, such as Surface Style or Layer, and attributes that may have been defined by plugins. These attributes cannot be edited, but are listed so that a user may examine their content for the purpose of creating an information record listing (see next section).

![Figure 3.7.4.1: The Attributes dialog.](image)

At the lower portion of the Attributes dialog there is a list with the fields that make up the attribute selected in the attributes list. It will be referred to as the fields list. Next to this list, the properties of the currently selected field are shown.
The attributes list

The attributes list is divided into two columns. The left column contains the name of the attribute. The right column shows the creator of the attribute. There are three possible creators: form•Z, Plugin, and User. One of the entries in the list is always highlighted and is called the current attribute. The name of the current attribute can be edited directly in the list by clicking on it for a second time. If a user types an attribute name that already exists, it is not accepted.

To the right of the attributes list there are three buttons:

New: When this button is pressed, a new user defined attribute is created. Initially, the new attribute has a single field. More fields can be added, as discussed later in this section. The new attribute becomes the current attribute in the list.

Copy: When this button is pressed, the current attribute is copied to a new attribute. All fields of the current attribute are copied. The new attribute becomes the current attribute in the list.

Delete: This button is disabled if the current attribute’s creator is form•Z or a plugin. If the current attribute’s creator is a user and the button is pressed, the current attribute is deleted.

Under the attributes list there are two more options:

Assign Through Query Attributes Dialog: When this option is on, the currently selected attribute will not be automatically assigned to new objects. The attribute can be assigned to an object, after its initial generation, using the Query Attributes tool. When an object is picked with that tool, the user defined attributes are shown in the Additional Attributes list in the Query Object Attributes dialog. Double clicking on an attribute in the list assigns it to the object and brings up a dialog where the values for each attribute field can be set. This process is described in more detail in section 4.22.3. This option is disabled for attributes created by form•Z or a plugin.

Assign When New Object Is Created: When this option is on, the attribute will automatically be assigned when a new object is created. This option should be chosen with care, as some attributes may require a significant amount of data storage. This option is disabled for attributes created by form•Z or a plugin.
The fields list

Under the attributes list is a list that displays all the fields of the current attribute. One of the fields is always selected and is called the current field. Next to the list are the properties of the current field and New and Delete buttons. Clicking on another item in the list makes it the current field. Fields may be moved around in the list by clicking on them and dragging them to new positions. The properties of the current field may be edited, if the current attribute is user defined. Otherwise, the field properties are shown in read only mode. The name of the current field can be edited by clicking on it for a second time. The names of all the fields within each attribute must be unique. If a user types a name that already exists, the name is not accepted.

Note that the fields for form•Z attributes do not completely describe the entire content of the attribute. Only the fields that are relevant to creating information record listings (see section 3.7.5) are shown. The fields of plugin attributes may also not be complete. It is up to the plugin to decide which of the fields of the attribute it wishes to list in this dialog.

New: When this button is pressed, a new field is created at the end of the list and made the current field. Its initial properties are taken from the field that is current at the time the New button is clicked.

Delete: When this button is pressed, the current field is deleted.

Format: This pop up menu contains 14 field types, as shown in Figure 3.7.4.2. The item selected from this menu determines how the value of an attribute field will be shown, when the attribute is edited in the Query Object Attributes dialog. The field types are as follows:

Whole Number: This field type allows a user to enter a whole (integer) number.

Fraction: This field type allows a user to enter a fractional (floating point) number. The precision and number of decimal places is determined by the settings in the Project Working Units dialog.

Distance: This field type allows a user to enter a distance value. The precision and display format of the distance value is determined by the settings in the Project Working Units dialog.

Area: This field type allows a user to enter an area value. The display format is determined by the item selected in the Units menu, which appears when this type is selected.

Volume: This field type allows a user to enter a volume value. The display format is determined by the item selected in the Units menu, which appears when this type is selected.
Angle: This field type allows a user to enter an angular value. The precision and display format of the angle is determined by the settings in the Project Working Units dialog.

Percent: This field type allows a user to enter a percentage value. The precision and number of decimal places is determined by the settings in the Project Working Units dialog.

Currency: This field type allows a user to enter a monetary value. The currency symbol is determined by the setting in the operating system.

Date: This field type allows a user to enter a date value (year, month and day). The date format is determined by the setting in the operating system.

Time: This field type allows a user to enter a time value. The time format is determined by the setting in the operating system.

Duration: This field type allows a user to enter a time duration value (hours, minutes and seconds).

Boolean: This field type allows a user to choose from a pop up menu, which contains the items False and True.

String: This field type allows a user to enter a set of characters (string) which define a name.

Value List: This field type allows a user to choose an item from a menu. The content of the menu is determined by the value list chosen from the List menu, which is displayed when Value List is selected from the Type menu. Value lists are described in more detail below.

Units: This menu appears when Area or Volume is selected from the Format menu. The units in which the area or volume value of the attribute field is displayed can be selected from it.

Default: The values displayed in this field depend on which field type is selected in the Format menu. It allows the user to set the default value in the respective format. For example, when the field type is Boolean, the Default item becomes a menu consisting of False and True. The selected default value is assigned to the field, when an attribute is first assigned to an object.

If the selected field type is one of the first 11 items (Whole Number, Fraction, Distance, Area, Volume, Angle, Percent, Currency, Date, Time, Duration), the user may also set an upper and lower limit for the field, through the Min and Max options that appear under Default. If Min is checked, the value entered to its field defines the smallest possible value that can be entered by a user for the current field. If the Inclusive option next to it is also on, the minimum value is allowed as the smallest value. If it is off, a value entered by the user for the current field must be greater than the minimum value. The same applies to the Max option, except that it defines the upper limit.
**Value Lists...:** When this button is pressed, the **Value Lists** dialog is invoked, as shown in Figure 3.7.4.3.

A **value list** defines a set of named items, that are presented to the user in the form of a menu. One of the field types can be a value list, as described above. An example of the use of a value list would be a menu with the names of all the States of the US. In the **Value Lists** dialog, the user may define new value lists and edit the content of existing ones.

The dialog consists of two areas. On the left is an area that shows all the currently defined value lists in a scrollable list window. One of the value lists is always highlighted and is called the **current list**. On the right is a window, which shows all the items of the current list. One item, the **current item**, is always highlighted. The name of the current list and current item can be edited by clicking on it for a second time. The names in each window must be unique. Under each window there are **New** and **Delete** buttons.

**New:** Clicking on the button below the left window creates a new value list with one item. The new list becomes the current list. Pressing the **New** button below the right window creates a new item in the current list and makes it the current item.

**Delete:** Pressing the left button deletes the current list. Pressing the right button deletes the current item in the right window.
An example

In the following example, a new user defined attribute is created, assigned to an object and then edited in the Query Object Attributes dialog.

The Attributes... menu item is selected from the Options menu. In the Attributes dialog, the New button, next to the attributes list, is pressed. The new attribute is named “Paint”, as shown in Figure 3.7.4.4. The Assign Through Query Attributes Dialog option below the attributes list is on by default.

Next, the fields for the new attribute need to be defined. By default, a new attribute is always created with one field. For this first field, a value list is created. Pressing the Value Lists... button at the bottom of the dialog opens the Value Lists dialog. Pressing New under the left window creates a new value list. It is renamed to “Paint Manufacturer”. On the right, new items are created and given the names of some paint suppliers, as shown in Figure 3.7.4.5. The same process is repeated to create another value list called “Paint Gloss”, which gets three items on the right: “Flat”, “Satin” and “Gloss”.

Figure 3.7.4.4: The Attributes dialog

Figure 3.7.4.5: The Value Lists dialog.
After pressing **OK** in the **Value Lists** dialog, the current field is named “Manufacturer”. From the **Format** menu, **Value List** is chosen. From the **List** menu, “Paint Manufacturer” is selected. From the **Default** menu, the preferred paint supplier is chosen. The **Fields** section of the **Attributes** dialog after these steps is shown in Figure 3.7.4.6.

A second field for the paint gloss is created by pressing **New** next to the **Fields** window. The same steps as for the manufacturer field are repeated, except that the “Paint Gloss” value list is selected. See Figure 3.7.4.7.

The next field created is of type **String** and will be called “Color Name”, where a user can enter the representative name of a paint color. From the **Format** menu, **String** is chosen and “White” is entered in the **Default** text item, as shown in Figure 3.7.4.8.

The fourth field created is of type **Whole Number** and will be called "Color ID". It is intended to store a paint i.d., which is usually supplemental to the paint name. The **Min** option is checked and the minimum value is set to 1000, inclusive. See Figure 3.7.4.9.

The last field created is of type **Currency** and is called “Cost Per Gallon”. The default is set to $20 and the **Min** option is set to 0, exclusive, as shown in Figure 3.7.4.10.

Now the definition of the attribute is complete and the **Attributes** dialog is closed by pressing **OK**.
Next, a new object is created, for example a 3D enclosure, representing a wall. The Query Attributes tool is selected and the new wall object is picked. This invokes the **Query Object Attributes** dialog, shown in Figure 3.7.4.11. The “Paint” attribute shows up in the **Additional Attributes** list, but its status is **Unused**, since the attribute is not automatically added when the object is created. Double clicking on the “Paint” item in the list assigns the attribute with default values and invokes a dialog, in which the fields of that attribute can be edited, as shown in Figure 3.7.4.12. Initially, the fields show the default values set up for this attribute, as described above. New values can be entered. Pressing **OK** saves the new attribute values.

![Figure 3.7.4.11: The Query Object Attributes dialog.](attachment:image1)

![Figure 3.7.4.12: The Paint dialog.](attachment:image2)
3.7.5 Information management

The Query and Query Attributes tools are designed to retrieve object quantities, attributes, and other specific object information and present it to the user. As described in more detail in section 4.22, these tools allow the user to look at one object at a time. In order to generate a listing of more than one object and extract specific object information, the Information Management... (IM) command, found in the Options menu, can be used. With a simple three step process a spreadsheet-like listing can be generated for viewing, exporting, and printing:

1. An IM record is defined. A record consists of a number of fields. Each field defines a particular piece of information of an object, such as the layer it is on, its volume or an attribute value.

2. An object selection method is chosen. It defines which objects are shown in the listing. Each selected object generates one row in the output listing. Each field of the record creates one column. It is possible to show all objects in a project or a subset, which may be picked manually or chosen based on certain criteria.

3. The listing is generated. This step is automatic and involves just the click of a button. Depending on the fields defined in the record, the listing can become, for example, a door or window schedule, a bill of materials, or an inventory.

IM records are saved with a form•Z project file. This allows the user to continuously produce updated listings of a project as it evolves and as changes are made to the objects.

Selecting the Information Management... item invokes the Information Management dialog shown in Figure 3.7.5.1. In this dialog, a user can create new and edit existing IM records, create and edit fields of a record, and produce output listings.

The dialog is divided into three sections:

- The Records section consists of a list of all currently defined IM records. One of the records in the list is always highlighted and is called the current record. Buttons next to the list allow the user to create new, copy, and delete records, and to generate the output listings.

- The Selection section has three options, which define how objects for the current record are selected.

- The Fields section also has a list. It shows all the fields defined for the current record. New fields can be created and existing fields can be deleted with buttons next to the list. One of the fields is always highlighted and is called the current field. The parameters of the current field are shown under the list. A field can display one of three types of information:
• A known object quantity, such as the object’s value.

• The value of a field of an attribute that may be assigned to an object.

• A value that is calculated by a formula. This third method gives the user the most flexibility in calculating very specific object information by using a script function. This script function is implemented in the FSL language, which is described in more detail in Chapter 3 of the form•Z SDK Manual. The script function can use any of the form•Z api functions to extract and calculate object quantities, attributes, and other information.
The Records section

Under and next to the records list are a number of buttons that manipulate the records in the list. The name of a record can be edited directly in the list by clicking a second time on the current record.

**New**: When this button is pressed, a new record is created, which becomes the current record. By default it has one field.

**Copy**: When this button is pressed, the current record is copied as a new record. The new record becomes the current record and is appended to the list. All the fields and the selection method of the current record are copied as well.

**Delete**: When this button is pressed, the current record is deleted. The record next to the deleted record becomes the current record.

**Generate Data...**: When this button is pressed, the output listing for the current record is generated and presented in a dialog, as shown in Figure 3.7.5.2. This dialog consists of a large list. Each row in the list represents one object. Which objects are listed depends on the selection method chosen for the current record, as discussed below. Each column of the list represents one field of the current record. As a result, one particular cell in the list contains the information defined by a specific field extracted from a particular object. Under the list there are two buttons:

![Figure 3.7.5.2: An output listing from a record.](image)
**Export As CSV...**: When this button is pressed, the standard File Save dialog is invoked. It allows the user to save the content of the listing, which is saved in CSV format, a standard spreadsheet format. Thus the exported file can be imported into Excel and be further manipulated.

**Platform**: A platform format for the exported CSV file is chosen from this menu.

**Append Units**: When this option is selected, the units for volume, area, and weight are included in the CSV file. If this option is not selected, only the bare number is written. When the file is opened in a spreadsheet, this option allows the spreadsheet to treat the data as pure numbers, and allows it to perform additional numeric operations on it.

**Print...**: When this button is pressed, the output list is printed. The standard print dialogs are invoked to determine the page layout and number of pages to be printed.

Two additional options are available when generating the output listing:

**Debug Formulas**: When this option is selected and the value of a field is generated by a formula script (see later in this section), the script statements are shown in the Debug dialog when the value is calculated. This allows the user to skip through the script code one statement at a time and determines whether the statements have the desired effect. Debugging scripts is explained in further detail in section 3.8.2 of the form•Z SDK documentation.

**Show Row Content**: When this option is selected, an additional column is shown in the output listing. It contains the content for each row, as well as labels for total and average values.

**The Selection section**

There are three options in this section that control how objects are selected for inclusion in a record.

- **All Objects**: When this option is selected, all objects are included in the output listing.

- **Picked Objects**: When this option is selected, the objects that are picked at the time the Information Management dialog is invoked are included in the output listing. This allows the user to select very specific objects, either manually or through other selection methods. If no objects are picked when the dialog is opened, no objects will be shown in the output listing.

- **Select By**: When this option is on, the objects to be included in the output listing are chosen based on the criteria defined in the Selection Criteria: Modeling dialog. This dialog can be invoked by pressing the Options... button next to the Select By option. This is the same dialog as the one invoked from the Select By... item, located in the Edit menu of the main menu bar. It is described in more detail in section 3.2.4. Choosing the Select By method allows the user to identify a particular group of objects, based on certain criteria. For example, only objects on certain layers that are symbol instances from a specific library may be selected through this method. This method is often preferred, when a project contains a large amount of data, such as a building, but only a subset of that data needs to be listed, such as the doors and windows of the building. By carefully organizing the objects on layers and choosing attributes appropriately, the Select By method can be used to always pick only those objects that represent the doors and windows of the building. For example, the selection criteria may be set up in such a way, that only objects on layers “1st Floor”, “2nd Floor”, and “3rd Floor” and only those objects that have an attribute called “Door Details” are selected.
The Fields section

The list in the Fields section contains all the fields defined for the current record. The order of the fields in the list can be changed by dragging the current field to a new position. A new field can be created by pressing the New button next to the list. The current field can be deleted by pressing the Delete button. At least one field must remain in a record. The name of the field can be edited directly in the list by clicking a second time on the current field.

The following three options define what kind of object information is specified by the current field.

**Quantity:** When this option is on, the object quantity selected from the menu is computed and listed. When a particular quantity is not applicable, such as the volume for a surface object, a value of zero is substituted. The quantities available are shown in Figure 3.7.5.3.

- **Volume, Weight, and Surface Area** are as calculated in the Query dialog.
- **Length** quantity only applies to curve like objects, which consist of a single, open or closed wire face.
- **Width, Depth, and Height** quantities are the extent of the object along the object axes.

**Attribute:** When this option is on, the value of the field of the attribute assigned to an object is listed. The attribute is chosen from the menu next to the option and the particular attribute field is chosen from the Field menu under it. Depending on the attribute selected from the Attribute menu, the Field menu changes to show which fields are available for the attribute. There are three kinds of attributes that will show up in the menu.

- Preset form•Z attributes, which are **Object Name, Layer, and Surface Style**. Their only field is **Name**, which is a text string.
- Attributes defined by plugins. These are shown in the menu, if any exist and if a plugin attribute publishes its fields for IM listing.
- User defined attributes. All of these attributes are listed and all the fields of each user defined attribute are available in the Fields menu. These are discussed in more detail in section 3.7.4. If a selected object does not have a particular attribute, a “-” is displayed in the respective cell of the output listing.
Formula Script: When this option is on, the object information is computed by a formula, which is defined by a script function, written in the FSL script language. Depending on which format option is chosen from the Format menu under it, the script function will compute a floating point, an integer, a Boolean (TRUE/FALSE), or a string value. The script function can be thought of as a small program that executes once per selected object, when the output listing is produced. It can calculate any desirable information of the object that is not readily available in the Quantity or Attribute option. Typically, the script function will call one or more form•Z api functions, which extract basic object information. Api functions are provided by form•Z and can be called from a script or plugin extension. They can perform various actions, such as object creation, object editing, and object analysis. More details about the FSL script language and all available form•Z api functions can be found in the form•Z SDK Manual.

Edit...: Pressing this button next to the Formula Script option invokes a simple text editor dialog, where the body of the script function can be typed, as shown in Figure 3.7.5.4.

Figure 3.7.5.4: The Edit Formula dialog.
Only those statements in the function that constitute the body need to be defined. The function header and the function return statement are already included and cannot be changed. This is evident in the layout of the dialog, where the header and the return syntax are drawn outside of the editable window. In order to produce the proper result, the function variable \texttt{rv} needs to be assigned the desired calculated value. Depending on which format is chosen for the \texttt{Formula Script} option, the \texttt{rv} variable is defined as a double, long, string, or \texttt{fzrt\_boolean} type.

\textbf{Format}: From this menu, the format in which the value computed by the formula can be chosen. For example, the floating point value for a wall area represents an area. In such case the \texttt{Area} format should be selected. This will cause the number to be interpreted as an area value and it will be formatted accordingly.

\textbf{Units}: When the current field represents a numeric value of a certain kind, such as an area or volume, the units in which the computed value is displayed is chosen from this menu.

\textbf{Footer}: The two options in this group determine what summation will be displayed at the foot of the numeric columns. These options are only available if the field defines a numeric quantity.

\textbf{Total}: If this option is on, the sum of all the values calculated for the current field is added at the end of the column in the output listing.

\textbf{Average}: If this option is on, the average value of all computed values will be displayed.
An example of a script function is shown below. It computes the surface area of an object, but only considers those faces that are oriented perpendicular to the xy plane. This may be useful, for example, when creating a paint schedule, and only the vertical (i.e. wall) surfaces of wall objects need to be considered.

double formula_double(long windex, fz_objt_ptr objt)
{
    double rv = 0.0;

    long i,nface;
    fz_plane_equ_td plane;
    double area;

    /* COUNT THE NUMBER OF FACETTED FACES IN THE OBJECT */
    fz_objt_get_face_count(windex,objt,FZ_OBJT_MODEL_TYPE_FACT,nface);

    /* LOOP FOR EACH FACE */
    for(i = 0; i < nface; i++)
    {
        /* GET THE PLANE EQUATION FOR THE ith FACE */
        fz_objt_alys_get_face_plne_parm(windex,objt,i,
            FZ_OBJT_MODEL_TYPE_FACT,plane);

        /* IF THE c COMPONENT OF THE EQUATION IS ZERO, THE FACE IS VERTICAL */
        if ( fabs(plane.c) < FZ_MATH_TOL_LIN )
            {
                /* GET ITS AREA AND ADD IT TO THE RETURN VALUE */
                fz_objt_alys_get_face_area(windex,objt,i,
                    FZ_OBJT_MODEL_TYPE_FACT,area);
                rv = rv + area;
            }
    }

    return(rv);
}
Another example of a Formula Script, which defines a boolean value, is shown next. It determines whether to paint the ceiling of a room object, based on the presence of faces that point straight down. This is in essence a variation of the script above, except that \textit{rv} is now of type \texttt{fzrt\_boolean}, which can only have the value \texttt{TRUE} or \texttt{FALSE}.

\begin{verbatim}
fzrt boolean formula_boolean(long windex, fz_objt_ptr objt) 
{ 
    fzrt boolean rv = FALSE;

    fz_plane_equ_td plane;
    fz_xyz_td cog,f_cog;

    /* GET THE CENTER OF GRAVITY FOR THE OBJECT */
    fz_objt_alys_get_objt_cog(windex, objt, cog);

    /* COUNT THE NUMBER OF FACETTED FACES IN THE OBJECT */
    fz_objt_get_face_count(windex, objt, FZ_OBJT\_MODEL\_TYPE\_FACT, nface);

    /* LOOP FOR EACH FACE */
    for(i = 0; i < nface; i++)
    {
        /* GET THE PLANE EQUATION FOR THE ith FACE */
        fz_objt_alys_get_face_plne_parm(windex, objt, i,
            FZ_OBJT\_MODEL\_TYPE\_FACT, plane);

        /* GET THE CENTER OF GRAVITY FOR THE ith FACE */
        fz_objt_alys_get_face_cog(windex, objt, i,
            FZ_OBJT\_MODEL\_TYPE\_FACT, f_cog);

        /* IF THE c COMPONENT OF THE EQUATION IS -1.0, AND THE CENTER OF */
        /* GRAVITY OF THE FACE IS ABOVE THE CENTER OF GRAVITY OF THE OBJECT, */
        /* THE FACE CAN BE CONSIDERED PART OF THE CEILING */
        if ( fabs(plane.c + 1.0) < FZ_MATH_TOL\_LIN \&\& f_cog.z > cog.z )
        {
            /* ASSUME THE ROOM OBJECT HAS A CEILING AND STOP THE LOOP */
            rv = TRUE;
            break;
        }
    }

    return(rv);
}
\end{verbatim}
A listing of selected objects based on a record that uses both formulas is shown in Figure 3.7.5.5.

![Paint Schedule](image)

**Figure 3.7.5.5:** The **Edit Formula** dialog.
3.8 The Palettes menu

The Palettes menus can be used to open or close palettes. As shown in Figure 3.8.0.1, in both modeling and drafting, they consist of four groups.

- The first group contains floating palettes that support different program features many of which remain open during a session. When the default preferences are used, at start up time, the system opens a modeling window and the Animation, Coordinates, Layers, Lights, Modeling tools, Objects, Prompts, Surface Styles, Tool Options, and Views palettes. When you open a drafting window the Colors, Coordinates, Drafting tools, Layers, Line Styles, Line Weights, Prompts, Tool Options, and Views palettes are displayed by default. Note that some of the palettes are available in both modeling and drafting, while others are only available in modeling or only in drafting.

- The second group, in both modeling and drafting, consists of palettes that also exist as dialogs. These are used frequently enough to deserve the ability to stay open, which they can do as palettes but not as dialogs. They are the same in modeling and drafting, except that modeling has one more, Status Of Objects. At start up time the system by default opens the Tool Options palette, but the user may open more.

Open palettes are indicated with a check mark in the Palettes menu. They can be closed by clicking on their close box or by deselecting them in the Palettes menu. The palettes, their features, and their functions are discussed in detail in sections 2.3 through 2.13.

- The remaining two groups contain one item each, Hide Palettes and Customize Tools..., which are the same in modeling and drafting. These are discussed in this section.
3.8.1 Animation palettes

The top items in the Palettes menu of the modeling environment refer to animation operations. While they are shown here for the sake of completion, they are discussed in detail in Chapter 7: Animation of the form•Z User's Manual.

**Animation Editor**

This item invokes the Animation Editor palette, shown in Figure 3.8.1.1. It is discussed in section 7.1.6.

**Animation Score**

This item invokes the Animation Score palette, shown in Figure 3.8.1.2. It is discussed in section 7.1.3.

**Animation Time Line**

When this item is selected, the Animation Time Line palette, shown in Figure 3.8.1.3, is displayed, which is the default. This palette is discussed in section 7.1.2.
3.8.2 Modeling palettes

Of the twenty two palettes available in modeling (which includes the animation palettes covered in the previous section), eleven are unique to modeling and five are shared with drafting. Following all the modeling palettes are outlined in alphabetical order, as they appear in the menu.

Coordinates

This item opens or closes the Coordinates palette (Figure 3.8.2.1) which displays the position of the mouse as it moves about the graphics screen. Different types of measurement and a variety of formats can be used to display numeric values. This palette can also be used to type in numeric values. The Coordinates palette is discussed in section 2.4.

Layers

This item opens or closes the Layers palette shown in Figure 3.8.2.2. Modeling objects and drafting elements are placed onto layers when they are created, and can then be moved from one layer to another. Layers are essentially a method by which the entities of a project can be grouped and organized. The Layers palette allows you to define new layers, delete existing layers, change the attributes of a layer, and select the active layer. The layers and their palette are discussed in detail in section 2.7.
Lights
This item opens or closes the Lights palette shown in Figure 3.8.2.3, which is only available in modeling. In 3D, lights complement the colors assigned to objects and affect their shades. Multiple lights can be defined in the Lights palette all of which are of a single type, distant, which includes the sun, in the regular version of form•Z. Different types of lights are available in form•Z RenderZone Plus. The lights and their palette are discussed in detail in section 2.10 and in the RenderZone Plus User's Manual.

Modeling Tools
This item opens or closes the Modeling Tools palette shown in Figure 3.8.2.4. This palette is open by default at start up time.

Objects
This item opens or closes the Objects palette shown in Figure 3.8.2.5. This palette is only available in modeling. It contains all the objects and groups of objects in a project. These can be selected directly from the palette as well as by clicking on them with the Pick tool. The Objects palette is discussed in section 2.8.

Planes
This item opens or closes the Planes palette shown in Figure 3.8.2.6, which is only available in modeling. A reference plane can be any one of the Cartesian planes (XY, YZ, or ZX) or any arbitrarily positioned plane. Any number of arbitrary planes can be generated by the user through tools available in the program. These planes can then be named and saved in the Planes palette from where they can be reapplied with a click of the mouse. The Planes palette is discussed in subsection 2.13.1.
Profiles
This item opens or closes the Profiles palette shown in Figure 3.8.2.7. Profiles apply to modeling only, and are shapes that are used to determine the form of mesh movements. The active profile is selected from the Profiles palette, which is also used to create and store new profiles. The profiles and the functionality of the Profiles palette are discussed in section 4.11.1.

Prompts
This item opens or closes the Prompts palette shown in Figure 3.8.2.8. The Prompts palette performs three functions: it issues instructions about the parameters that are needed for the execution of an operation; it displays the values of these parameters when entered graphically; and it supports input of these parameters through the keyboard. The functions of the Prompts palette are discussed in detail in section 2.4.

Scenes
This item opens or closes the Scenes palette shown in Figure 3.8.2.9. The function of this palette is discussed in section 2.11.1.

Surface Styles
This item opens or closes the Surface Styles palette shown in Figure 3.8.2.10 for both modeling and drafting. The function of the Surface Styles palette is discussed in section 2.6.
Symbols

This item opens or closes the Symbols palette shown in Figure 3.8.2.11 for both modeling and drafting. The function of the Symbols palette is discussed in section 2.12. Whether the Symbols palette displays modeling or drafting symbols depends on the active window.

Views

This item opens or closes the Views palette shown in Figure 3.8.2.12. Any current view (for both modeling and drafting) can be assigned a name and saved through the Views palette. Saved views may be recalled at a later time. The Views palette is discussed in section 2.9.

Window Tools

Recall that the Window Tools palette is located at the left portion of the lower margin of the form•Z window and is shown in Figure 3.8.2.13. While it is not possible to tear off the entire palette from the lower margin of the window, it is possible to separate it through this item. The items contained in this palette are discussed in section 2.2 of this manual.
3.8.3 Drafting palettes

From the eleven drafting palettes, four are unique to drafting and seven are shared with modeling.

**Colors**

This opens or closes the Colors palette (Figure 3.8.3.1) which contains the colors available to drafting elements. Individual colors may be redefined through the Color Picker dialog, accessed by double clicking on a color box. A single click selects that color as the active color.

**Coordinates**

This item opens or closes the Coordinates palette as for modeling (Figure 3.8.2.1), except that for drafting it only displays two coordinate values (X and Y). This palette is discussed in 2.4.

**Drafting Tools**

This item opens or closes the Drafting Tools palette shown in Figure 3.8.3.2. This palette is open by default at start up time.

**Hatch Patterns**

This opens or closes the Hatch Patterns palette (Figure 3.8.3.3). Hatch patterns apply to drafting only and are used to fill bounded areas of drawings. They are selected from the Hatch Patterns palette discussed in section 5.16.

**Layers**

This item opens or closes the Layers palette (Figure 3.8.2.2) as in modeling.
**Line Styles**

This item opens or closes the Line Styles palette (Figure 3.8.3.4), which is only available in drafting. It contains 11 preset line styles and more can be added. The active line style is used when new elements are drawn, as discussed in section 5.15.

![Figure 3.8.3.4: The Line Styles palette.](image)

**Line Weights**

This item opens or closes the Line Weights palette (Figure 3.8.3.5), which is only available in drafting. It contains 11 preset line weights and more can be added by a user. The active line weight is used when new elements are drawn. The line weights and their palette are discussed in detail in section 5.15.

![Figure 3.8.3.5: The Line Weights palette.](image)

**Prompts**

This item opens or closes the Prompts palette (Figure 3.8.2.9), as for modeling.

**Symbols**

This item opens or closes the Symbols palette, as for modeling. Note however that a different part of the symbols library, which is the drafting symbols, is displayed when this palette is opened in the drafting environment, as shown in Figure 3.8.3.6.

![Figure 3.8.3.6: The drafting Symbols palette.](image)

**Views**

This item opens or closes the Views palette (Figure 3.8.2.13), as for modeling.

**Window Tools**

This item opens or closes the Window Tools palette (Figure 3.8.2.14), as for modeling.
3.8.4 Dialog based palettes

Some frequently used dialogs are also available as palettes and can be invoked by items in the second group of the Palettes menus. These palettes can stay open on the screen, which makes their options more accessible than they are in their dialog counterparts. They are as follows:

**Display Options**

This item opens the Display Options palette for the current rendering mode. The content of this palette is the same with the rendering mode specific Options dialog. This is a context sensitive palette. That is, its content changes as the display mode changes and is different for modeling and drafting windows. Examples are shown in Figure 3.8.4.1.

![Screenshot a](image1.png) ![Screenshot b](image2.png) ![Screenshot c](image3.png)

**Figure 3.8.4.1:** Display Options palettes for (a) modeling *Wire Frame*, (b) modeling *Hidden Line*, and (c) drafting.
**Pick Options**

This item invokes the Pick Options palette (Figure 3.8.4.2) whose content is that of the **Pick Options** dialog (see section 4.3.2).

**Snap Options**

The palette invoked by this item has the content of the **Snap Options** dialog (see section 2.2.4).

*Figure 3.8.4.2:* The Pick Options palette.

*Figure 3.8.4.3:* The Snap Options palette with the (a) **Grid**, (b) **Angle**, (c) **Radial**, and (d) **Object** tabs open.
**Status Of Objects**

This item opens or closes the Status Of Objects palette shown in Figure 3.8.4.4. The function of this palette is discussed in section 2.13.3.

**Tool Options**

This item opens or closes the Tool Options palette (Figure 3.8.4.5). The Tool Options palette shows the options of the currently active tool and modifiers. When an icon is selected, the Tool Options palette is refreshed to show the current options which are used by the active tool and the modifiers it uses. These options are the same as the options that are found in the dialog which is invoked by double clicking on the icon. The Tool Options palette offers a more direct way to access the features. It is discussed in section 2.5.

**Window Options**

This item opens the Window Options palette Figure 3.8.4.6) whose content is that of the Window Setup dialog (see section 3.3.5).
3.8.5 Hiding the palettes and customizing the tool bars

The two items at the lower end of the Palettes menu perform the same functions in modeling and drafting.

Hide/Show Palettes

This item is used to hide all of the palettes. You do so by clicking on it when it reads Hide Palettes, which causes it to change to Show Palettes, while all the open palettes disappear. Clicking on Show Palettes brings the open palettes back. Note that this item does not close the palettes. It simply makes them invisible.

Customize Tools...

This last item of the Palettes menu, invokes the Icons dialog (Figure 3.8.5.1), which is used to customize form-Z’s default tool palettes (modeling, drafting, windows). How this is done is discussed in detail in section 1.9.

Figure 3.8.5.1: The Icons dialog.
3.9 Extensions menu

The items in the Extensions menu are used for controlling which plugins and scripts are enabled and where they are located on your computer’s hard disk.

An initial set of plugins and scripts is created during form•Z installation depending on your product features and options chosen during installation. Additional plugins and scripts may be installed during separate installations, by newer form•Z versions and third party form•Z extensions.

A plugin is an extension to form•Z that is contained in a form•Z plugin file (.fzp). form•Z plugin files are binary executable libraries that perform specific tasks. Like the form•Z application, form•Z plugin files are compiled for specific operating systems and processors, and can not be moved between operating systems. Because they are optimized in this fashion, plugins execute significantly faster than scripts. Plugins also have more access to the internal workings of form•Z, hence they often offer features that can not be created with a script. A single form•Z plugin file may actually contain a number of plugins. Plugins may also use a number of support files such as form•Z resource files (.fzr). These support files are installed in the same directory as the form•Z plugin file and must be kept in this location.

The following types of plugins are some of those currently supported:

- File translator: Imports/exports data to/from file formats.
- Image file translator: Imports/exports image data to/from image formats.
- Tool: Adds a new operation to form•Z, including a new tool icon to run it from.
- Utility: Code that supports a task, which is not a complete tool.
- Digitizer: Interfaces to hardware digitizers.

A script is an extension to form•Z that is contained in a form•Z script language file (.fsl) or the form•Z script binary file (.fsb). The form•Z script language file is a text file which contains the script instructions in the form•Z script language (.fsl). The form•Z script binary file is a binary version of the script that is generated from the text file and allows for more efficient processing of the script. The script files are not platform specific and can be interchanged between different platforms. Scripts are interpreted rather than compiled for a specific system so they do not execute as fast as a plugin. As for the plugins, different types of scripts will be supported.
The **Extensions** menu, shown in Figure 3.9.0.1, contains 4 items in the top group. The remainder of the menu may contain additional items or hierarchical menus created by extensions. Selecting one of these items performs the corresponding extension defined action.

**Extensions Manager...**

This item invokes the **Extensions** dialog, shown in Figure 3.9.0.2. It is composed of two main sections: the plugins and scripts list which occupies the upper portion of the window, and a number of menus and buttons in the lower portion.

![Extensions dialog](image)

**Figure 3.9.0.2:** The **Extensions** dialog.
**Run Utility...**

This item is used to run utility extensions. Utility extensions are designed to execute a task which is either less frequently used or it is not desired to have a menu item for the task appear in the form•Z interface. Utility plugins are best used on tasks that are linear in nature (like batch processing). Utility plugins are not loaded by form•Z at startup and are not listed in the Extensions dialog. When the **Run Utility...** item is selected, a standard file open dialog is invoked to select the extension file to run. A utility can be a plugin file (.fzp) or a script file (.fsl/.fsb). Once the file is selected, the utility is executed.

**Run Recent Utility**

This pop-out menu lists the most recent utilities that were executed using the **Run Utility...** command. Selecting the utility file name from the menu immediately executes the utility.

**Use Script Debugger**

This item enables and disables the script debugger. For details on debugging scripts, see section 3.7.5 of the SDK Documentation.

The Extensions dialog displays information about each plugin and script that form•Z has knowledge of. Each row of the list represents a single plugin or script. The list contains 6 columns, as follows:

- **On**: This determines if the plugin or script is enabled or disabled. Enabled plugins and scripts will be loaded the next time form•Z starts. Disabled plugins and scripts will not be loaded. A check (√) in this column indicates that it is enabled (no mark is shown if it is disabled). The title for this column has a small menu next to it. This menu can be used to turn on or off all the plugins and scripts in the list. Changes to the enabled plugins do not take effect until you restart form•Z. If you have changed the “enable” state of any of the plugins or scripts when you exit the dialog with the **OK** button, you will be given the option to quit form•Z.

- **Name**: This is the name of the plugin or script. Note that this is not the name of the plugin file, but rather the name that the developer of the plugin or script has chosen.

- **Type**: This is the type of the plugin or script as described above.

- **Kind**: This indicates whether the item is a plugin or a script.

- **Version**: This is the version of the plugin.

- **Vendor**: This is the name of the vendor who developed or distributed the plugin or script. Contact that vendor for updates to the plugin or script and for technical support.

Clicking in any column except the first column in the list invokes the **Plugin Details** or **Script Details** dialog. These dialogs contain detailed information about the plugin or script as follows:
The Plugin Details dialog, shown in Figure 3.9.0.3, displays the following information:

**Name**: The name of the plugin (not the plugin file).

**ID**: A numeric ID assigned by the plugin developer.

**Version**: The version of the plugin (assigned by the plugin developer).

**File Name**: The name of the plugin file (.fzp) containing the plugin.

**Path**: The full path to the folder containing the plugin file.

**Type**: The type of the plugin as described above.

**Type ID**: A numeric ID which identifies the plugin type.

**Type Version**: The version of the plugin type.

**API Version**: The version of the plugin API used to create the plugin.

**Vendor**: The name of the vendor who developed or distributed the plugin.

**Vendor URL**: The URL to the plugin vendor’s web site.

**Status**: Whether or not the plugin is loaded.

**Description**: If the plugin developer wished to provide additional information, it will be contained in this field. If no description is provided by the plugin developer, this field will not be displayed.
The **Script Details** dialog, shown in Figure 3.9.0.4, displays the following information:

**Name**: The name of the script (not the script file).

**ID**: A numeric ID assigned by the script developer.

**Version**: The version of the script (assigned by the script developer).

**Executable File Name**: The name of the executable script file (.fsb) containing the script.

**Executable Path**: The full path to the folder containing the executable script file.

**Type**: The type of the script as described above.

**Status**: Whether or not the script is loaded.

![Figure 3.9.0.4: The Script Details dialog.](image)
Clicking on the column title in the plugins and scripts list (at the top of a column) sorts the list alphabetically using this column. Pressing the **option** key on the Macintosh or **ctrl+shift** on Windows when clicking in the column title will sort the list in descending order rather than the default ascending order.

At the lower part of the **Extensions** dialog, there are a few menu items and buttons, as follows:

**Selected Sets**: This group of options allows you to define new sets of plugins and/or scripts or to delete existing sets. A selected set is a number of items picked from the plugins and scripts list, to which a name is assigned, to be able to identify them as a set of selections. Any number of such sets can be defined and show up in the **Selected Set** menu.

**Selected Set**: This menu allows for the selection of previously defined sets of enabled plugins. When an item from this menu is selected, the enabled state of all the plugins and scripts is restored to the state at the time the selection set was saved. The default selection set is named **Default Set** and can not be removed.

**New...**: This button saves the enabled states of the plugins and scripts shown in the **Plugins And Scripts** list to a named selection set. When selected, you are prompted to enter a name for the set. This name will show up in the **Selection Set** menu. Selecting that name in the **Selection Set** menu will set the enabled states of the plugins and scripts to the states they were in at the time the selection set was saved.

**Delete...**: Clicking on this button will remove the named selection set currently shown in the **Selection Set** menu.

**New Plugins, New Scripts**: These menus set how plugins and scripts, which are new to form•Z, are handled. The menus contain the three items shown in Figure 3.9.0.5.

- **Load**: Simply load all new plugins or scripts.
- **Don’t Load**: Don’t load any plugins or scripts.
- **Ask Before Loading**: Prompt before loading each new plugin or script.

**Search Paths...**: This button brings up the **File Search Paths** dialog, as shown in Figure 3.9.0.6. This dialog can be used to add and remove folders where form•Z should look for plugins and scripts. This dialog functions as the same dialog invoked from the **Preferences** (see section 3.2.7 in the form•Z User’s Manual). When form•Z is installed, two folders are added, by default: “Plugins” and “Scripts.” They are relative to the form•Z installation folder.
3.10 The Help menu

The Help menu (Figure 3.10.0.1) consists of four groups of items.

- The first group has three items that offer direct access to the internet, the form•Z website, and Technical Support through email.
- The second group has seven items that access information about form•Z and its operations.
- The third group contains a single item that offers access to the form•Z User's Manuals.
- The fourth group contains two items that invoke error messages issued by the system and information about the active project.

While a complete section has been dedicated to the Help menu at this point of the manual for the sake of completion, the items in the second group and the dialogs they invoke have already been presented earlier, in section 1.8.

![Figure 3.10.0.1: The Help menu.](image)
3.10.1 Accessing the Internet

The three items in the first group are intended to offer quick access to technical support about form•Z, either through the Web site or through email.

form•Z Web Site...

Clicking on this item launches your operating system’s default web browser and opens the Home Page of the auto•des•sys, Inc. web site (http://www.formz.com), where a variety of information about our products as well as support material can be found.

form•Z Web Support...

Clicking on this item works as for the previous item, except that it takes you to the Technical Support page of our web site (http://www.formz.com/support/index.html).

email Tech Support...

Clicking on this item opens your operating system’s default email application and sets a blank email addressed to the form•Z Technical Support department (support@formZ.com).
3.10.2 Summary information about the operations of form•Z

General...

This item invokes the General Help dialog shown in Figure 3.10.2.1. This dialog gives you information about the active tool (if one is selected) and offers you buttons for branching to the different parts of the Help environment. You can also go to the different parts of Help directly by selecting one of the other buttons. Selecting one of these buttons invokes the respective dialog. These dialogs and their contents are discussed in section 1.8.

![General Help dialog](image)

Figure 3.10.2.1: The General Help dialog.
3.10.3 Accessing the form•Z Manuals

Manual...

Clicking on this item invokes the form•Z Manual Viewer with which the manual can be browsed. The manual opens to its title page from where other pages can be accessed.

Recall that there are a number of additional ways for accessing the form•Z User's Manuals. These are discussed in section 1.8.

The form•Z Manual Viewer looks and behaves differently on Windows, and Mac OS X. This is due to the availability of different PDF document viewing technologies on these platforms. The highlights of each of these applications are described in the remainder of this section.
Mac OS form•Z Manual Viewer

The Mac OS Help Viewer screen is shown in Figure 3.10.3.1. Its features are as follows:

1. A horizontal **menu bar**, located at the top of the screen, contains five menus, plus the Apple () menu (Macintosh only). Their commands are discussed later in this section.

2. A horizontal **tool bar** or iconic menu, positioned right under the menu bar. Its tools provide commonly used operations and are discussed later in this section.

3. A collapsible **search results pane**, located just under the tool bar, is closed by default. It can be opened either by performing a search or dragging the separator bar (4). This pane shows the result of a text search and is discussed below.

5. The **content pane** shows the content of the form•Z manual, which can be scrolled horizontally and vertically using scroll bars located at the bottom and right side of the pane.

6. An outline of the form•Z manual is shown in the **outline drawer**. The outline provides an overview of the manual and can be used to navigate through it.

---

**Figure 3.10.3.1**: The OS form•Z Manual Viewer screen.
The tool bar

**Previous**: This button sets the view to the previous page of the form•Z Manual.

**Next**: This button sets the view to the next page of the form•Z Manual.

**Back/Forward**: These buttons set the view to the previously viewed page of the form•Z Manual. This is similar to the Forward and Back buttons of a web browser.

**Current Page**: This text field shows the page number of the current page. Typing a new page number followed by a return will take you to that page.

**View Mode**: This button sets the view to one of three modes:

- **Continuous**: The entire form•Z Manual can be scrolled in the view.
- **Single Page**: The form•Z Manual is displayed one page at a time.
- **Facing Pages**: The form•Z Manual is displayed 2 pages at a time. The pages are arranged side by side as though looking at a book.

**Zoom In**: This button increases the magnification of the view.

**Scale**: This text field shows the scale factor of the view. Typing a scale factor followed by a return will set the scale of the view. Values greater than 100% magnify the view. Values less than 100% reduce the view.

**Zoom Out**: This button decreases the magnification of the view.

**Search**: Typing text in this field will start a search of the form•Z Manual for the typed string. The results of the search are shown in the search results pane.

**Outline**: This button opens or closes the outline drawer.
The menu bar

File menu

Open...: This item opens a PDF file into the form·Z Manual Viewer. The document to be opened is selected through the standard Macintosh File Open dialog.

Open Recent: This item is a pop out menu that contains a list of recently viewed PDF files. Selecting one of the items in the pop out menu will open that file.

Page Setup...: This item invokes the Page Setup dialog for the current printer.

Print...: This item invokes the Print dialog for the current printer. Clicking the OK button sends the document or a portion of the document to the printer.

Edit menu

Copy: This item copies selected text to the Macintosh clipboard.

Find: This item is a pop out menu that contains additional search capabilities.

Find...: This item invokes the Find dialog, shown in Figure 3.10.3.2. A search string is entered into the Find text field. When the OK button is pressed, the dialog will close and the next occurrence of the string will be located in the form·Z manual. The page containing the string will be displayed in the content pane and the string will be highlighted. Unlike the Search tool on the tool bar, this command does not fill the search results pane. If the Ignore Case option is on, the search will ignore the case of the string. If it is off, only text with matching case will be found.

Find Next: This item continues a Find... operation by finding the next occurrence (searching toward the end of the form·Z manual) of the string without the need to invoke the Find dialog and reenter the string. This item is disabled until a Find... or Use Selection For Find operation is performed.

Find Previous: This item continues a Find... operation by finding the previous occurrence (searching toward the beginning of the form·Z manual) of the string without the need to invoke the Find dialog and reenter the string. This item is disabled until a Find... or Use Selection For Find operation is performed.

Use Selection For Find: This item sets the currently selected text to the search string for the Find Next and Find Previous operations. Text is selected by dragging the cursor over a portion of the text.
Jump To Selection: This item sets the view to the page containing the currently selected text.

Special Characters...: This item invokes the standard Macintosh character palette.

Go menu

Next: This item sets the view to the previous page of the form•Z Manual.

Previous: This item sets the view to the next page of the form•Z Manual.

GoTo Page...: This item invokes the Go To Page dialog, shown in Figure 3.10.3.3. When the OK button is pressed, the content pane will display the page entered in the Page text field.

Back: This item sets the view to the previously viewed page of the form•Z manual. This is similar to the Back button of a web browser.

Forward: This item sets the view to the previously viewed page of the form•Z manual. This is similar to the Forward button of a web browser.

Window menu

Minimize: This item minimizes the active window to the dock.

Open Windows: The remainder of this menu is a list of all open PDF files. Selecting one of these items brings the corresponding window to the front and makes it the active window.

The search results pane

This pane displays the results of a search performed with the Search tool on the tool bar. After a search is performed, this pane is filled with the search results on a tabular form. Each line of this table represents one occurrence of the search string. This table contains three columns. The first column displays the page number of each occurrence, if the search string is on. The second column displays the section title of each occurrence, if the search string is in. The third column contains some text surrounding the search string.

Clicking on a row of the results table will cause the content pane to display the page of the form•Z manual containing that instance of the search string. The search string will be highlighted. Down/up arrow keys on the keyboard will select the next/previous line in the search results. This will also cause the content pane to display the page of the manual containing that instance of the search string.

A vertical scroll bar on the right side of this pane can be used to scroll through the search results.
The outline drawer

Volumes, chapters, and sections of the form•Z manual are shown in the form•Z Manual Viewer outline drawer, which can be opened and closed using the Outline button on the tool bar. The width of the outline drawer can also be adjusted by clicking on the right edge of the drawer and dragging it to the desired width.

The outline is displayed in a hierarchical fashion; volumes are at the highest level, followed by chapters contained within them, followed by sections contained within the chapters. Each level of the hierarchy can be collapsed or expanded by clicking on the arrow icon to the left of the title string. Clicking on the title string will cause the content pane to display the first page of the selected volume, chapter, or section. The keyboard arrow keys can also be used to navigate through the outline. The down/up arrow keys will select the next/previous displayed item in the outline. This will also update the contents pane. The right arrow key will open the next lower level of the selected item. The left arrow key will close the next lower level of the selected item.

A vertical scroll bar on the right side of this drawer can be used to scroll through the outline.
The Windows form•Z Manual Viewer

The Windows form•Z Manual Viewer is an application that wraps Adobe Reader and provides the control interface between form•Z and Adobe Reader.

To use the form•Z Manual Viewer on Windows, Adobe Reader 7.0 or greater must be installed.

Figure 3.10.3.4 shows the Windows form•Z Manual Viewer with Adobe Reader version 7.0.5. Depending on the version of Adobe Reader installed the interface may look different. Refer to the documentation installed with Adobe Reader for details on using this viewer.

Figure 3.10.3.4: The Windows form•Z Manual Viewer screen.
3.10.4 Messages and information

Error Messages...

This item invokes the Error Messages dialog (Figure 3.10.4.1), which lists the error messages issued by the system. This list is particularly useful when the Show Error Message Immediately option (in the Project: Warnings group of the Preferences dialog) is off and the error messages are not displayed right after an irregularity is encountered (see section 1.7).

Project Info...

Clicking on this item invokes the Project Info dialog, shown in Figure 3.10.4.2, which displays information about the active project and the entities currently contained in both its modeling and drafting environments (see section 1.7).

Figure 3.10.4.1: The Error Messages dialog.

Figure 3.10.4.2: The Project Info dialog.
3.11 The contextual menu

The contextual menu, shown in Figure 3.11.0.1, is a shortcut for accessing frequently used items. It is designed to improve workflow by minimizing mouse movement and extra clicks. This menu is derived from the standard method for contextual menus established by the operating system.

Different methods for accessing the contextual menu are available for single-button and multi-button mice, as follows:

• When using a multi-button mouse, click on the right (secondary) mouse button while the cursor is over a modeling or drafting window. The menu can also be accessed by using the control + left (primary) mouse button (Macintosh) or ctrl + left (primary) mouse button (Windows).

• When using a single-button mouse, click while pressing control (Macintosh) or ctrl (Windows).

The contextual menu key combination can be changed to any desired combination with a mouse button, using the key shortcuts manager.

When the mouse is clicked as above, the contextual menu appears at the click point on the screen. The content of the menu appears different, depending on the current state of the modeling or drafting windows, such as whether any entities are picked and where the mouse is clicked. For example, the Cut, Copy, etc. items will be dimmed if no item is selected in the window. Similarly, Paste will be dimmed if the Clipboard is empty. After the menu is invoked, any of its active items can be selected by clicking on it. This causes the menu to disappear and the selected action to be executed.

The mouse click that invokes the contextual menu can also act as a pick (selection), if it is on a pickable entity. That is, it works the same as the Pick tool. However, unlike the Pick tool, if the cursor is clicked away from a pickable entity, the selections are not cleared.

Figure 3.11.0.1: The modeling contextual menu of form•Z.
The first group of items in the menu contains 5 items. Cut, Copy, Paste, and Duplicate are the same items found in the Edit menu. The last item, Delete, is the same as the Delete tool. That is, selecting this item deletes the selected objects.

The next group contains 4 items commonly used for querying and editing objects: Query, Query Attributes, Edit Parameters, and Edit Controls. When these are selected, the corresponding operation is executed.

The next sections of the menu contain shortcuts for the form-Z modifiers. There is a section for each of the modifier types, however, only the modifiers that apply to the current tool are shown in the menu. For example, if the active tool is a drawing tool that uses the direct generation and insertion modifiers, then a group containing these items appears in the menu. Selecting one of these from the menu is the same as selecting the respective icon from the tool palette. If the active tool uses the topological level modifiers, then a group containing these items appears in the menu. Likewise if the active tool uses the self/copy modifiers, then a group containing these items appears in the menu. A check mark is placed in front of the active modifier in each group.

The last three groups in the contextual menu offer access to the three types of snapping: Grid, Directional, and Object. Selecting an item from these groups enables the respective snapping type. A check mark is placed in front of an item to indicate that it is active. Note that during a drawing or other interactive action, only the snaps are available; the items in the other groups do not apply and are dimmed.
3.12 The form•Z Imager

The Imager is a utility that allows you to produce sequences of static renderings, animations and QuickTime VR files in batch mode. It is incorporated in form•Z and is also available as an independent utility, which is shipped with the product. The Imager is useful for rendering images that may require considerable time to complete during time that the computer would not be used, such as overnight or during weekends. It is also useful for rendering different views from the same model. It supports all the plotting/rendering types available in form•Z or in form•Z RenderZone Plus. Images can also be printed after they have been generated. Renderings can be performed locally or rendered over the network using the form•Z Render Server.

From within form•Z, the Imager is launched by clicking on the New Imager Set... or the Open item, both in the File menu. The first item creates a new Imager set. The second allows you to open a previously saved Imager set, edit it, and regenerate its images. Both invoke the Imager set window shown in Figure 3.12.1.1, from which the Imager is run.

To produce renderings and print outs with the Imager, the following steps are involved.

• First, set up the views in a form•Z project file, and save them using the appropriate tools.

• Then open an Imager set window and use the Add... button to create entries in the Imager set. These are essentially links to the modeling project to be rendered, and to one of the views saved in the project.

• Then set up the desired rendering parameters, including the type of rendering to be executed. This is done independently for each entry, but operations that apply the rendering options to all of the entries are also available.

• Start the rendering process by clicking on the Generate or Network Render button.

The renderings will be produced one after the other continuously. Upon completion of each image, it is saved as one of the available image file formats or a form•Z Drafting project. Wireframe and hidden line images can also be saved as form•Z drafting project files. As images are produced, they are not displayed on the screen, but they may be printed. Once the images are completed, they can be viewed using the View... button.
3.12.1 The Imager set window and list

The Imager set window consists of two sections: the **Imager set list**, which occupies the upper portion of the window, and a number of **button commands** in the lower portion (Figure 3.12.1.1).

The Imager set list displays the names of the images in the current Imager set, in the order in which they will be generated. Each row of the list represents a single image. Clicking the mouse in the list highlights the image/row identified by the location of the mouse, and defines the **active image**. The active image is the image which will be affected by the selection of a command button and some commands from the pull down menus. There can only be one active image at a time, therefore, the active image is toggled when another image is selected. The active image can also be changed using the up and down arrows on the keyboard. If the list is empty, or the mouse is clicked below the last image in the list, no active image is defined.

Clicking in the scroll bar scrolls the Imager set list to view any additional entries that do not fit in the list area.

The list consists of five columns which identify the image and display its status. The width of the imager set columns can be changed by clicking and dragging the column dividers in the list or in the title area at the top. When the cursor is positioned over a column divider, the cursor changes to (.), indicating that the column divider can be repositioned.

The Imager set list can be resorted by clicking on any one of the column titles. Clicking on a title column sorts the list alphabetically by the items contained in that column. Pressing the **option** key (Macintosh) or **ctrl+shift** (Windows), when clicking in the column title will sort the list in descending, instead of ascending alphabetical order. The sorting is applied only when the command is executed, and is not repeated automatically when new items are inserted or existing items are deleted or moved.

![Figure 3.12.1.1: The Imager Set window.](image)
The order of the list can also be changed manually by moving image entries up or down. This is done by clicking on an image name and dragging it to the desired position.

- The first column (Project) identifies the name of the project to which the image is linked.
- The second column (View/Scene) identifies the name of the view in the project to which the image is linked. It also shows the view type (Pers, Axon, Panm, etc.) in brackets next to the name. The first two columns display the first 19 characters of a name. Longer names are displayed using the first 19 characters, followed by the truncation symbol (…). Double clicking in the View column has the same effect as selecting the Preview... button (see section 3.12.2).
- The third column (Rendering Type) displays the type of rendering which will be used to generate the image. Double clicking in this column has the same effect as selecting the Image Specs... button (see section 3.12.2).
- The fourth column (Generation Type) controls what type of generation is used for the image. There are four types of generation that can be performed: Animation, Image, Image And Print and QuickTime VR. By default if a file contains animation information, the generation mode is set to Animation otherwise it is set to Image.
- The last column displays the current status of the image, which changes during the generation process. The following status indications are displayed:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pending</td>
<td>This image is waiting to be generated.</td>
</tr>
<tr>
<td>Generating</td>
<td>This image is currently being generated.</td>
</tr>
<tr>
<td>Skip</td>
<td>This image is being temporarily skipped and is not generated.</td>
</tr>
<tr>
<td>Complete</td>
<td>This image has been successfully generated.</td>
</tr>
<tr>
<td>Printing</td>
<td>This image has been successfully generated and is currently being printed.</td>
</tr>
<tr>
<td>Error[xxx]</td>
<td>An error occurred while generating or printing the image, where “xxx” is the numeric identifier of the error.</td>
</tr>
</tbody>
</table>

Double clicking in the status column makes the selected image the active image, and invokes the Image Status dialog, shown in Figure 3.12.1.2. It displays messages regarding image generation and provides the ability to change the status of the image.

At the top of this dialog is the image Status pop up menu, which displays the current status of the active image. This status can be changed by selecting another item from the menu in the usual manner.
• The “Generating” and “Printing” status are controlled by the generation process, and therefore always appear dimmed.

• The “Complete” status is only available after the image is successfully completed.

• The “Error[xxx]” status is set whenever an error occurs during image generation or printing. The error is identified by its numeric identifier, and a description of the error is displayed in the message area of the dialog. Since the “Complete” and “Error[xxx]” status are controlled by the generation process, they appear dimmed when they are unavailable.

• The “Pending” status indicates that an image is ready to be generated. This is the default status and is always available.

• The “Skip” status, which is also always available, indicates that the image should be skipped by the generation process.

The area outlined with a rectangle is the message box. This box displays a variety of messages related to the image generation, which includes error messages, generation timing information, and printing information. This message is cleared when the image begins generation, therefore the message always refers to the most recent generation of the image.
3.12.2 The Imager operations

Next to the titles of the three last columns there are pull down menus (▼) and at the lower portion of the Imager set window there are ten buttons. These are used to define the parameters for the active image, or for all the images in the Imager set.

**Display Type**: The desired image type for all images is set by selecting one of the rendering options from this pull down menu: **Wire Frame, Interactive Shaded, Quick Paint, Hidden Line, Surface Render, Shaded Render, RenderZone, Shaded Radiosity**, or **RenderZone Radiosity** (see Figure 3.12.2.1(a)).

**Generation Type**: The generation method of all the images in the set may be determined by selecting **All Animation, All Image, All Image And Print** or **All QuickTime VR** from this pull down menu. (Figure 3.12.2.1(b))

**Status**: Selecting an option from this pull down menu (Figure 3.12.2.1(c)) changes the status of all the images to **Make All Pending** or **Skip All**.

While the settings selected from the menus in the titles of the last three columns affect the complete image set, it is also possible to select settings for individual images only. This can be done using pop up menus that are invoked from within the Imager set list. Clicking and holding the mouse down in one of these columns on the active image invokes the respective settings menu for the column. Selecting an item from these menus changes the setting for the active image only. These menus are shown in Figure 3.12.2.2.

*Figure 3.12.2.1:* The (a) Rendering Type, (b) Generation Type, and (c) Status column menus.

*Figure 3.12.2.2:* The (a) Rendering Type, (b) Generation Type, and (c) Status item menus.
The button commands at the lower portion of the Imager set window are used to perform the following operations:

**Imager Setup...**: Clicking on this button invokes the **Imager Set Up** dialog (Figure 3.12.2.3). This dialog is used to control the default rendering type, the type of image file generated for the rendered images; to select auto save and naming options; and to set the destination folder for the generated images. The options in this dialog affect all the images in the Imager set.

**Default Image Type**: This pop up menu is used to specify the default rendering type that is assigned to new images when they are added to the Imager set using the **Add...** command (see below, this section). The items in this menu represent all the rendering levels available.

**File Format**: This group determines the file format that the exported images are saved in.

**Vector Renderings**: This group is used to determine how vector renderings like wire frame and hidden line images are saved. The items in the pop up menu identify the type of file that will be saved. The menu on both platforms contains items for all the available image formats plus Illustrator and form•Z drafting project. The **Options...** button invokes the options dialog for the type of file selected from the format pop up menu. The **form•Z Export Options** dialog is shown in Figure 3.12.2.4.

**Polygon Renderings**: This group is used for Quick Paint and Surface Renderings and is identical to the **Vector Renderings** item, except that it does not have the option for saving to **form•Z Drafting Project** files since this format only supports line drawings.

**Pixel Renderings**: This group is used for Shaded Render and RenderZone renderings and for all panoramic views except Wireframe. It is identical to the **Polygon Renderings** item, except that it does not include the item for saving to the **Illustrator** file format, which is an exclusively vector based format.
Generate Unique Image File Names: When this option is selected, the system automatically generates unique names for each image generated from the Imager set. When this option is not selected, successive images with the same name replace those generated earlier.

Completed images are saved as files named using the project name and the view name separated by a hyphen (-). On the Macintosh, file names are limited to 31 characters; therefore, the name will be truncated to 31 characters. If the name is identical to an existing file name, it will have a unique suffix added to the file name, before the file extension. This suffix is in the form of a hyphen followed by a unique numeric identifier. The numeric identifier starts at 1 and is increased until a unique file name is created. For example, for an image from project “myProject” with view “myView,” saved as TIFF, the file name would be “myProject-myView.TIFF.” If this name already exists, the file is named “myProject-myView-1.TIFF.” If this file also exists, the file is named “myProject-myView-2.TIFF,” etc. until a unique name is found.

Destination: This group of options controls where the generated image files, form•Z files, and QTVR files are saved. They are the same as with the Image Autosave Options dialog explained in section 3.6.3.

With Project: When this option is selected (default), the generated images and QTVR files are saved in the same folder as the project from which they are generated.

Custom: When this option is selected, all of the generated images and QTVR files are saved in a designated folder, which is selected using the Set Destination Folder button. The current custom location is shown under the Set Destination Folder button.

Set Destination Folder...: Clicking on this button invokes a dialog from which the destination folder for the images can be set. Upon selection of this button, the Set Destination dialog is presented, as shown in Figure 3.12.2.5. An existing folder can be selected by highlighting any file in a folder and selecting the Open button or by highlighting a folder name and clicking on the Use Current button.

A new folder can be created by clicking on the New Folder button. This opens the New Folder dialog (Figure 3.12.2.6) in which the name of the new folder may be entered. The new folder is created at the hierarchical level selected when the New Folder button was pressed.
Auto-Save Imager Set During Generation: When this option is selected, the current Imager set is automatically saved to disk after the completion of each image during the generation progress. This is useful if you are running a large number of images that will take a long time, and there is a chance that the process may be interrupted (by, for example, a power outage). This option is only available to new Imager sets that have been saved at least once or to existing Imager sets that were opened using the Open... command from the File menu. This option does not significantly slow down the generation process and is on by default.

Clear Rendering Memory During Generation: When selected, this option can be used to control the use of memory allotted to rendering information while the Imager set is running. One of two options may be selected to determine when the rendering memory will be cleared.

When Switching Between Projects: When this option is selected, the rendering memory will be cleared when switching between projects in the Imager set.

After Each Image is Complete: When this option is selected, the rendering memory will be cleared after each image is completed.

File Search Paths...: Clicking on this button invokes the File Search Paths dialog (see Figure 3.2.7.20). This is identical to the dialog invoked by File Search Paths... in the Preferences dialog, and its workings are discussed in section 3.2.7 of this User’s Manual.

Add...: Clicking on this button invokes the standard Open File dialog (Figure 3.12.2.8), from which a new image may be added to the Imager set. Images are added by selecting saved views from a form•Z project file. The desired project file is selected by double-clicking on the file’s name or by highlighting the project’s name and clicking on the Open button. Selecting the Cancel button terminates the Add... command.

Figure 3.12.2.7: The Open File dialog.
Selecting **Open** from the Open File dialog invokes the **Imager Add Views And Scenes** dialog (Figure 3.12.2.8), from which the desired views and scenes may be selected. This dialog presents lists of the names of all of the views and scenes saved in the project file. The views and scenes to be loaded are selected by highlighting their names in the list. Multiple views and scenes are selected by holding down the mouse button and dragging it. Individual views or scenes can be selected or deselected by holding down the **shift** key and clicking on the view or scene name or dragging the mouse across a range of names. There are two **Select All** button, one for Views and one for Scenes. These buttons can be used to select all the views or all the scenes in the project. The **Cancel** button terminates the **Add...** command. If there are no views or scenes saved in the project file, the system displays a warning (Figure 3.12.2.9) and cancels the **Add...** command.

Once the desired views and scenes have been selected, clicking **OK** completes the view or scene selection process. Each selected view or scene is loaded as a new image and is added to the end of the list in the order it appears in the **Views From** or **Scenes From** dialog. The names of the project and the view or scene are placed in the first two columns. The image rendering type is set to the default, which is set in the **Imager Set Up** dialog (Figure 3.12.2.3). The image print option is initially set to “Print None,” and the status is set to “Pending.” The active image is not affected by **Add...**.

While the **Image Options** (image size and resolution) and **Display Options** (rendering settings) can be set independently for each image in the image set, they also can be linked to the window settings of the project being rendered. The following three items enable the Imager Set and the project to be linked so that changes made to respective settings in the project may be automatically picked by the Imager.

*Figure 3.12.2.8:* The **Imager Add Views And Scenes** dialog.
Use Settings From: This menu contains a list of all of the windows of the project being added to the Imager. Each window is identified by the window and frame number as they appear at the top of the form-Z windows. These settings are initially defaulted to the Image Options and Display Options in effect for the selected project window.

Link Image Options: With this on, Image Options are linked to the default window.

Link Display Options: With this on, Display Options are linked to the default window.

Remove: Clicking on this button removes the active image from the Imager set. The same result can be achieved by pressing the delete (or del) key on the keyboard. The images below the removed image are shifted up in the list, and the image directly below the removed image becomes the active image. If the removed image is the only image in the list, there will be no active image after it is removed. When there is no active image, this command is inactive and dimmed.

Preview...: Clicking on this button invokes the Preview window (Figure 3.12.2.10(a)), which is a graphics window placed on top of the Imager set window, with a wire frame preview of the image drawn in the window. It shows the size and proportions of the image. If it is too large to fit on the screen, the preview window is scaled down proportionately. The project name and the scale of the image are shown in the title bar. This window can be moved and closed in the usual ways.

View...: This button works as Preview... and displays the most recently generated image (Figure 3.12.2.10(b)), rather than a wire frame. The image name is shown in its title bar and not the project file. This item is only available when there is an already generated active image.

Print/Plot Setup...: Clicking on this button invokes the Plot/Print Setup dialog, from which the printing parameters may be set (see section 3.1.6).

QuickTime VR Options...: When QuickTime VR (QTVR) export is enabled for the active image, this button is enabled and it invokes the QuickTime VR Movie Options dialog. This dialog is the same dialog that is used by the Save QuickTime VR menu item in the File menu (see section 3.1.3 and Figure 3.1.3.1.) When this dialog is accessed from the Imager set window, it has the following modifications:
• The **Use Current Panoramic View** option is only enabled if the active image is a panoramic view.

• The **Display Type** menus in both the **Generate Panoramic Movie** and **Generate Object Movie** are disabled and show the rendering type selected in the Rendering Type column of the active image.

• The **View Movie When Complete** option is disabled.

**Animation Options...**: This button is enabled when the generation type for the currently active image is set to Animation. This button invokes the **Imager Animation Options** dialog that is used to control the animation settings. The contents of this dialog are the same as the **Animation Generation** dialog described in section 7.4.2. Note that the **Generate On Network** and **Generate Animation** items are not present in the **Imager Animation Options** dialog as the generation execution the Imager is controlled by the **Generate** and **Network Render...** buttons in the imager window.
Display Options...: Clicking on this button invokes the Imager Display Options dialog (Figure 3.12.2.11), from which the type of rendering and its options may be set. This is very similar to the dialog invoked from the Display Options... item in the Display menu (section 3.6), but it has the following three additional options:

Use Display Options From: With this option on, the display options are derived from the project window identified by the menu to the right. The project window settings are shown in the remainder of the dialog and the rendering type specific options dialogs, but they cannot be edited.

Use Custom Display Options: With this on, the display options are set specifically for the current image using the parameters and dialogs in the remainder of the dialog.

All Images: When this option is selected, the settings from the display options dialog are applied to all images in the Imager set that are based on the same project file.

Image Options...: This button invokes the Image Options dialog (Figure 3.12.2.12), where the image type, size, and color depth are set. This dialog is similar to that invoked from the Image Options... item (Display menu, see section 3.6.9) but has the following three additional Imager specific options:

Display Type: This menu determines the display type for the active image. It is the same as the Rendering Type menu shown in Figure 3.12.2.2 (a).

Use Image Size From: When this is on, the image options are derived from the project window shown in the menu to the right. The project window settings are shown in the lower portion of the dialog but cannot be edited.

Use Custom Image Size: When this option is on, the image options are set specifically for the current image using the parameters in the lower portion of the dialog.
**All Images**: When this option is on, the settings from the image options dialog are applied to all images in the Imager set that are based on the same project file.

**Generate**: Clicking on this button initiates the image generation process. If there are no pending images, this item is dimmed and inactive.

The Imager begins generating images starting with the first image in the Imager set with a status of “Pending.” It continues to generate pending images in the order of the image list. The status of the image changes to read “Generating” as it is generated, “Printing” as it is printed, and “Complete” when it is done. If an error occurs, the status changes to indicate the error, and generation continues with the next image.

During generation, a progress window is displayed as shown in Figure 3.12.2.13. The bottom bar fills from left to right, indicating the approximate progress of the rendering.

![Figure 3.12.2.13: The progress bar.](image1)

Prior to starting the image generation, the Imager checks if any of the projects to be rendered reference symbol libraries or image map files that cannot be found. When some are not found, a warning is issued, and you have the option to stop the generation, or you can choose to render the Imager set anyway. In the latter case, where the system does not find the file it needs, it will substitute bounding volumes for missing symbols and the default image map for missing images.

Image generation can be interrupted by pressing `command+period` (Macintosh) or `ctrl+period` (Windows), or `esc` (both platforms). This invokes the **Image Generation Interrupt** alert, shown in Figure 3.12.2.14, which presents three choices. **Continue** (default) resumes the generation process. **Next Image** aborts the current image and generates the next pending image. **Cancel All** aborts the complete Imager set. The interrupted image is returned to pending status.

If the interrupted image is partially complete, the **Save Partially Generated Image** alert (Figure 3.12.2.15) is presented. Selecting **Save** saves the partial image following the same naming conventions as a completed image. The directory/folder where the images are saved can be set through the **Set Destination Folder...** menu command.

![Figure 3.12.2.14: Image Generation Interrupted alert.](image2)

![Figure 3.12.2.15: Save Partially Generated Image alert.](image3)
3.12.3 The Imager menus

When form-Z Imager is active, the form-Z menu bar changes to accommodate it. Menus that are not needed by the Imager are not shown. The File (Figure 3.12.3.1), Edit (Figure 3.12.3.3), and Window menus remain active with only options that are used for an Imager set.

The File menu

When the Imager is active, the items that are available in the File menu are shown in see Figure 3.12.3.1.

New [Model], New [Draft]

These items work as in the main form-Z module.

New Imager Set

This command opens a new, blank Imager set window and creates a new Imager set which is named “Untitled.” Only one Imager set can be opened at a time. Therefore, this item is inactive and appears dimmed when an Imager set is already open.

Open...

This item opens a previously saved Imager set from a disk file. Selecting it invokes the standard Open File dialog, from which the Imager set to be opened may be selected. When an Imager set is already open, if the Open... command is executed, a warning message is posted offering two options: either to cancel the operation, or to close the currently open Imager set and open another one.

After selection of an Imager set, the system checks the links to the project files and views found in each image of the Imager set. If a project file cannot be found, the system will request that the location of the project file be identified through the standard Open File dialog, shown in Figure 3.12.3.2. The link is updated by highlighting the project file's name, and double-clicking the mouse or selecting Open. Selecting Cancel removes all the images in the Imager set which are linked to the missing project file. The links will need to be updated when the project file is moved to a different folder, or the project file is renamed, or the disk is renamed.

Figure 3.12.3.1: The File menu.

Figure 3.12.3.2: The standard Open file dialog.
If a linked view does not exist in the linked project file, the images which use the view will be removed from the Imager set. A view may be found missing when the view is deleted from the saved views in the project file, or if the Imager project file is relinked to a different file.

**Open Recent**
This popup menu lists files that have recently been saved. See section 3.1.2.

**Close**
This command closes the current Imager set. If changes have been made to the Imager set since the last time it was saved, a prompt is presented giving the opportunity to save the changes before closing the Imager set.

**Save**
This command saves the current Imager set as a disk file. If there have been no changes to the Imager set, this item is dimmed. If the Imager set has not been previously saved, the **Save As...** command is automatically invoked.

**Save As...**
This command is used to save an Imager set for the first time or under a different name. Upon selection of this command the standard **Save File** dialog is presented in which the file name may be specified. Once the new name is entered and the **Save** button is pressed, the file is saved to disk and the title at the top of the window is changed to reflect the new name.

**Save A Copy As...**
This command is used to save a copy of an already saved Imager set under a different name. Upon selection of this command, the standard Save File dialog is presented in which the file name may be specified. Once the new name is entered and the **Save** button is pressed, the file is saved to disk and the title at the top of the window is changed to reflect the new name.

**View File...**
This item invokes the Open File dialog, and allows you to select an image file to view. It is similar to the **View...** button at the bottom of the Imager set window (see section 3.12.2), except that the latter does not require you to select an image file, but displays the image that is active in the Imager set window.

**View Recent**
This popup menu lists files that have recently been exported or viewed. See sections 3.1.4 and 3.1.5.

**Quit / Exit**
This command terminates the current **form•Z** session (see section 3.1.7). If changes have been made to the Imager set since the last time it was saved, the system asks if the changes should be saved before quitting.
The Edit menu

When the Imager is active, the items are available in the Edit menu, as shown in Figure 3.12.3.3.

**Undo**

This command cancels the most recent change to the Imager set and restores it to the state it was before the last operation. This includes any changes made to the image list or the image options, such as specification and display option changes. This command only affects the last operation, and is dimmed and inactive when there is nothing that can be undone.

**Cut**

This command makes a copy of the active image in the clipboard and removes it from the image list. This command is applied to the active image, and is inactive and dimmed when no active image is selected.

**Copy**

This command also makes a copy of the active image in the clipboard, but leaves the active image intact. It is inactive and dimmed when no active image is selected.

**Paste**

This command adds the image from the clipboard to the image list. The new image is added after the active image. It is inactive and appears dimmed when the clipboard is empty, or when no active image is selected.

**Duplicate**

This command is equivalent to the sequential execution of the Copy and the Paste commands. That is, it makes a copy of the active image and places it in the list after the active image. It is inactive and dimmed when no active image is selected.

**Key Shortcuts**

When this item is selected, it invokes the Key Shortcuts Manager dialog, which is explained in section 3.2.5.

**Preferences**

When this item is selected it invokes the Preferences dialog, explained in section 3.2.7.

The Window menu

When the Imager is active, only the Windows List at the bottom of this menu is active.
3.12.4 Imager network rendering

An Imager set can be sent over the network to a form•Z Render Server for rendering on a rendering farm. The basics are very similar to how form•Z sends jobs to be network rendered, but there are some key differences.

Imager sets can be sent to the form•Z Render Server from either the stand-alone Imager or from an Imager set within form•Z. The process is the same in either case.

Only jobs that are marked as Pending in the status column of the Imager set will be sent to the form•Z Render Server for rendering. Jobs that have been sent to the form•Z Render Server will have their status changed to Network. Once rendering has been completed, the status will change to Complete. As long as the Imager or form•Z application that sent the jobs is running when the job is finished rendering, the completed work will be automatically retrieved. Viewing the completed work is the same as in a normal Imager set: click the View button. Imager sets can also be retrieved from the Network Render Status dialog, which works in exactly the same way as when a normal form•Z job is retrieved: click the Retrieve button in the Network Render Status dialog.

One difference between Imager set renderings and form•Z renderings is the options presented to the user when preparing to send the job to the Server. The options are fewer in an Imager set. Job names are taken from the name of the form•Z file and the view. For example, a form•Z file of “my file.fmz” with a view name of “my view” will be called “my file-my view”. If this name is the same as any other job, it will be renamed by appending a number to the end of the name. For example, “my file-my view-1”.

If there are multiple items in the Imager set that are to be rendered over the network, when the user clicks the Network Render... button, the user will be presented with the Network Render dialog only once. This is because Imager set network renderings use the options found in the Imager Set Up dialog to determine where the completed files are to be stored and what format to save them as. This is in contrast to a normal network rendering where the options are set for each job in the Network Render dialog. The Network Render dialog is only used to select the Server, the Priority of the Imager set, and whether to store the jobs in a Project.

There are some limitations to what can be rendered over the network. Renderings of still images (i.e. not animations) can only be of two types: RenderZone or Shaded. This is how it is with a form•Z rendering. If an image is set with one of the other types of renderings, it will not be sent to the server but will instead be skipped.

Currently, the Print and QTVR options in an Imager set are ignored.
3.13 Importing and exporting object file formats

As discussed in section 3.1.2, in addition to its own project files, form•Z can import and export a variety of other formats. These are of two categories: object file formats, which transfer descriptions of 2D shapes and 3D objects, and image file formats, which contain mostly bit mapped descriptions of scanned, captured, or rendered images. The object file formats are discussed in this section; the image file formats are discussed in the next section (3.14).

The object format files share a number of identical import and export options and therefore their dialogs share a common format. The common options are discussed in the following two subsections and the remaining subsections discuss the options that are unique to each format. Although the dialog formats are similar, there are instances where certain options are unavailable for a format type. In these cases, those options are grayed out (dimmed) and unavailable.
3.13.1 Common object import options

The dialogs invoked when importing object files into form-Z have been constructed to be consistent as they share many common options. There are three sections found in the object import options for each format. The top portion contains options for controlling the units, as shown in Figure 3.13.1.1(a). The next section is either the common options for modeling (Import As Modeling Objects) or drafting (Import As Drafting Elements), as shown in Figure 3.13.1.1(b) and (c). Which section is shown is determined by the selection of Import Model or Import Draft from the Import commands in the standard file Open dialog. The third section at the bottom of the dialog contains the options specific to the format (if any). In instances where an option is not available for a certain format, it is dimmed.

*Figure 3.13.1.1:* The common import options for (a) all formats, (b) modeling objects, and (c) drafting elements.
Setting the units of the imported entities

Most of the format files contain no information about the unit of measurement in which their numerics are written. For example, a 10 in DXF could be inches, feet, centimeters, meters, etc. For these formats it is necessary to tell form•Z what the units are, which is done through the following two pop up menus.

**form•Z Units:** This pop up menu (Figure 3.13.1.2(a)) determines how units will be set in form•Z, when the **Add To Project** option is not selected in the **Import** dialog. When this option is on, this menu is dimmed and unavailable. The units already set in the receiving project are used.

**Format Units:** The item selected from this pop up menu (Figure 3.13.1.2(b)) determines how the units in the imported file will be interpreted. **From File** is available only when the format of the imported file carries unit information and, when selected, it reads the numbers in the imported file as identified in the file. **As In form•Z** causes the units to be interpreted as determined by the selection in the **form•Z Units** pop up menu, or by the settings in the import project. All the other items specify a unit of measurement to be used for interpreting the incoming numbers.

Importing as modeling objects

**Import Grouping:** The items in this pop up menu (Figure 3.13.1.3) set the hierarchy of the imported file. The first item is the default for all formats.

- **Disabled:** When this item (default) is selected, no grouping information is constructed.

- **As Groups:** For FACT files, when this option is on, their grouping hierarchy is converted into form•Z grouping with each child as a nested group to its parent object. For VRML files, form•Z grouping structures are derived from the hierarchy of the VRML separator or grouping nodes.

- **As Layers:** For FACT files, when this option is on, the top level of each group in the file is represented as a form•Z layer, and all the children of the group are placed on the layer. For VRML files, a form•Z layer is constructed for each top level VRML separator or grouping node.
**Import Method:** This group of options controls the method used for converting imported objects to *form-Z* objects.

**Smooth Objects:** There are two options in this pop up menu which decide how parametric objects will be imported into *form-Z*’s modeling environment (Figure 3.13.1.4).

- **As Solids/Surfaces:** When this option is selected, parametric objects will be imported as solids or surfaces, depending on their structure.

- **Preserve Controls:** When this option is selected, parametric objects will be imported as *form-Z* parametric objects, which means their parameters can be manipulated in *form-Z* as with other parametric objects. The table in Figure 3.13.1.5 shows which formats support which parametric object types.

<table>
<thead>
<tr>
<th>Entity/Format</th>
<th>IGES</th>
<th>OBJ</th>
<th>RIB</th>
<th>SAT</th>
<th>3DMF</th>
<th>VRML</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-mesh</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Nurbz curve</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Nurbz object</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Analytic primitives</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓**</td>
</tr>
</tbody>
</table>

Notes: * Except cubes. ** Except toruses

*Figure 3.13.1.5:* Formats supporting parametric import/export and the type they can support.

**Create Object Text From Text Entities:** Text is contained in the DWG, DXF, IGES and Illustrator formats. When this option is on, all imported text is generated as *form-Z* surface text objects. The resolution is determined by the current setting in the *3D Text Editor* dialog. When this option is off, all text entities are generated as plain text. Object text can be redrawn faster, but they consume more memory. Plain text does not store the character outlines. While this conserves memory, it slows the redraw, since the outlines have to be regenerated each time.

---

3 - 284

**Menus** • Importing and exporting object file formats
Transforming files when importing as modeling objects

Different 3D programs may apply different conventions in the way they position the axes of their coordinate systems, the units of measurement they use, and how they position the normals relative to the surfaces that bound objects. To facilitate communications with programs that may follow different conventions than those used in form•Z, the ability to apply certain geometric transformations to data when importing files is available.

Transformation...: Clicking on this button found in the Modeling Import Options dialogs for all the object formats except for STL, invokes the Import/Export Transformation dialog shown in Figure 3.13.1.6, which offers the option to apply a variety of geometric transformations to the data before importing it. Note that the same dialog is used for both importing and exporting data. This dialog consists of three parts.

- the preview area on the left side of the dialog;
- the Coordinate System group of options at the top right of the dialog;
- the Transformations group of parameters at the middle right.

The preview area displays an original object (called the source object), and an object which shows how the object changes by the transformations selected in the dialog (called the destination object). Two systems of axes are displayed: one for the source and one for the destination object. The source object and its axes are displayed black. The destination object and its axes are displayed red. These colors are also shown on two lines displayed under the preview area. When importing, the imported objects are the source and the form•Z objects are the destination. The reverse applies when importing.

![Import/Export Transformation dialog](image)

*Figure 3.13.1.6:* The Import/Export Transformation dialog.
The images displayed in the preview window are refreshed each time a change is made to a parameter in the **Coordinate System** or the **Transformations** group of options. The display view can also be changed using one of the icons in the preview tool palette found under the preview window. These work as for the other preview windows of **form•Z** (see subsection 3.2.5).

**Coordinate System:** This group of options allows you to flip or switch the coordinates of each point in the data.

- **Flip X, Flip Y, Flip Z:** Mirrors data along the X, Y, and Z axis, respectively.
- **Swap Y–Z, Swap X–Z, Swap X–Y:** Interchanges the Y with the Z, the X with the Z, and the X with the Y coordinates, respectively.
- **Inverse Faces:** When this option is selected, the vertex order of each face is reversed. This is useful for formats where the normal order for vertices is the opposite of **form•Z**'s natural order.
- **Normals Point Outward:** When this option is selected, the destination object's normals will point outward from the object.

**Transformations:** The parameters in this group allow you to geometrically transform the data, by rotating, scaling, and/or moving it in one or more directions.

- **Scale:** When this option is selected, the factors entered in the X, Y, and Z fields are used to scale the data. For all fields the default factor is 1.00, which results in no scaling.
- **Rotate:** With this option selected, the angles entered in the X, Y, and Z fields are used to rotate the data about the respective axis. For all axes the default is 0°, which results in no rotation.
- **Move:** When this option is selected, the distances entered in the X, Y, and Z fields are used to move the data. For all fields the default is 0, which results in no move.

**Importing symbols**

**Symbol Options...:** This button invokes the **Symbol Import Options** dialog (Figure 3.13.1.7) that controls how the symbol definitions in an imported file will be treated.

- **Explode Symbols:** When this option is selected, all symbols and insertions are imported as common objects or elements. For each symbol instance, a copy of the objects or elements of the referenced symbol definition is made. This option is used as the default each time a file is imported.
**Import Referenced Symbol Definitions**: When this option is selected, form•Z imports all the symbol definitions that are contained in the file and are referenced. Recall that all placed symbols are automatically imported by the system as independent objects, at the position where they have been placed. This option offers the possibility to also import the definitions of the placed symbols, so that they may be further placed, after they have been imported into form•Z. The objects that are derived from symbol definitions are positioned at the origin (0,0,0) of the world space. They are not rotated and they are not scaled.

**Import Non-Referenced Symbol Definitions**: This option works as the previous for non-referenced symbol definitions, which are definitions that have not been placed.

**Ghost Objects Derived From Symbol Definitions**: This option ghosts the objects which are derived from symbol definitions and is only meaningful when either one or both of the previous two options is on. When importing into the drafting environment, this option is called **Ghost Elements Derived From Symbol Definitions**.

**Group Objects Derived From Symbols**: Symbols are frequently nested entities. That is, a symbol may consist of a number of other symbols or entities. form•Z generates one object for each entity in a symbol. This option preserves the hierarchical structure of a symbol, if any, by generating the corresponding grouping links. This option applies to both objects generated from placed symbols and from symbol definitions. When importing into the drafting environment, this option is called **Make Compound With Elements Derived From Symbols**. Since no grouping is available in drafting, elements generated from symbol definitions or symbol instances are used to create a compound element, which is a type of a grouping.

**Add Symbols To Library**: When this option is on, all symbol definitions in the file are converted to form•Z symbol definitions and stored in the symbol library selected from the pop up menu. All symbol instances are imported as form•Z symbol instances. If no symbol libraries are loaded, this option is not available.

**Add Symbols To New Library**: When this option is on, all imported symbol definitions are stored in a new symbol library. All symbol instances are imported as form•Z symbol instances. The name and location of the new library is defined in the file **Save** dialog, which opens automatically during the import process, when the first symbol is encountered.
Importing text

Text Options...: Clicking on this button invokes the Text Import Default Fonts dialog shown in Figure 3.13.1.8. From this dialog you can select a font for importing text into modeling or drafting. Note that some formats (DXF and DWG, for example) do not contain font specifications. Consequently, this selection is required to instruct the program what font to use when importing text.

Font: The item selected from this menu determines the font to be used when importing.

Constructing solids from closed polylines

Three dimensional entities imported from other applications are typically collections of polylines which, when taken together, enclose an entity equivalent of a solid. There is typically no information about which polylines belong together. To import these entities into the modeling environment of form•Z in a meaningful manner, an option is available that tells the program to construct solid objects.

form•Z collects all the surfaces that share edges and when it is able to derive a closed set of such surfaces, it constructs a solid. By repeating this process, it derives all the solids that are contained in the imported file. The surfaces that fail to incorporate in a closed set of bounding faces are imported as surface objects. Surface objects consist of one sided faces, do not satisfy the closure condition, and have a positive and a negative side.

Composition: There are three options (one with two sub-options) in this section.

Construct 3D Solids: When this option is on, groups of surfaces whose edges touch are compiled and solid objects are constructed from them, provided these sets of surfaces are complete and satisfy the closure requirement. That is, for form•Z to be fully successful in this task, the imported data should contain all the surfaces that are needed to completely enclose a solid object. form•Z does not attempt to create faces which may be missing. It will create as many solid objects as possible. Surfaces that fail to incorporate in the representation of a solid are imported as surface objects.

Same Color Surfaces: When this option is selected, the system will only search and compile surfaces of the same color when reconstructing the solid objects.

Same Layer Surfaces: With this option on, only surfaces on the same layer are compiled. When both options are on, groups of the same color and layer are collected.

Join Adjacent Coplanar Faces: When this option is selected, the system will automatically join all adjacent (sharing an edge) and coplanar faces of the same object into single faces. This option is independent of the Construct 3D Solids option.
Importing colors and rendering parameters

When importing entities through a file format that handles colors, these colors also need to be imported or approximated as closely as possible. Most use an RGB representation for colors, which is consistent with what form•Z uses. Others, such as DXF, use indexes that refer to a color table and no RGB definition is available. In both cases two options are available for handling color. These options can be found in all formats, except SAT and STL, which contain no color information. Notice that the options concerning texture mapping are dimmed within the regular version of form•Z as texture maps are only available in the RenderZone Plus version of the program. The import and export of textures and surface style parameters are discussed in subsection 3.13.3.

Attributes: There are three options in this section. Import Texture Maps and the Texture Options... button are only available in form•Z RenderZone Plus.

Color: The items in this pop up menu determine how the colors of the imported file are treated and how they affect the Surface Styles palette of the window where the file is imported.

Active: When this option is selected, all the imported entities are assigned the color currently selected as active in the Surface Styles palette. This option is not available when the Add To Project option is off.

File's Palette: When this option is selected (default), a new Surface Styles palette is created. It consists of all the colors found in the imported file and these colors are assigned to the respective entities as they are brought into form•Z.

A solid object imported into form•Z from another program is typically constructed from a group of surfaces that enclose it. Each of these surfaces is an independent entity that may carry a different color. Recall that form•Z has colors at both the object and the face level, and when a number of surfaces are joined into a single entity to construct a solid, a single color needs to be identified for assigning it to the object level. form•Z uses the color of the first entity in the group as the object level color. Whenever entities are imported as individual objects, then they simply maintain the color they have in the imported file.

Import Texture Maps and Texture Options...: These options are discussed in section 3.13.3.
Importing as drafting elements

The options contained in the draft version of the Import Options dialogs (Figure 3.13.1.9) determine how form•Z will treat data that is imported into the form•Z’s drafting environment.

Scale Factor: The factors entered in the X and Y fields are used to scale the imported data. Independent scaling factors may be entered for each of the two Cartesian directions, which offers opportunities for scaling transformations to be applied when data is imported. However, to keep the proportions consistent with the original data, the same factor should be entered in both fields. For both fields the default factor is 1.00, which results in no scaling.

Plot Scale: The value entered in this field (1/16” = 1’ by default) determines the scaling of the imported elements in comparison to that of the project it is being imported to.

Text Options..., Symbol Options...: As for modeling (see earlier this subsection).

Color: This pop up menu is similar to the pop up menu with the same name in the Attributes group for importing into modeling, discussed above. Note however, that, while in modeling these options affect the Surface Styles palette, in drafting they affect the Colors palette.
3.13.2 Common object export options

As with importing, the dialogs invoked when exporting modeling object files from form•Z to another program share many common options, which are displayed at the upper portions of the dialogs. The lower sections contain format specific options. Where one of the common options does not apply to a format, it is dimmed in the respective dialog.

Notation and platform types

When exporting to formats that support both ASCII and binary notations (RIB, SAT, STL, 3DGF, and 3DMF) options to select one or the other are available.

**File Type:** This pop up menu contains two items: **ASCII** and **Binary**. Either one can be selected. All the export file formats that can be saved as ASCII text, except RIB, SAT, and Lightscape support the **Platform** pop up menu, shown in Figure 3.13.2.2(a). File formats that are exclusively ASCII always have the **Platform** menu available. For file formats that can be either ASCII or binary the **Platform** menu is available when **ASCII** is selected and dimmed when **Binary** is selected from the **File Type** menu. The default selection of **Platform** is initially set according to the platform form•Z is currently running.

**Export units**

The following option applies when a format does not contain its own unit specification method.

**Units:** This pop up menu, shown in Figure 3.13.2.2(b), sets the linear units of the exported data. When the **Project Units** item is selected, the data is exported in the units set in the **Project Working Units** dialog (see section 3.7.1 of the User’s Manual). When one of the other items is selected, the respective unit is used to export the data.

Note again that the options in this menu only apply to formats that do not have their own unit specification options. The ones that do, such as Lightwave, Lightscape, and VRML use their own specifications. For these formats this pop up menu is dimmed and unavailable.
**Grouping method**

When exporting a **form•Z** project, the objects it contains may be structured as a single group or as many groups. These groups can be structured to contain one object each, all the objects of the same color, all the objects on the same layer, or all the objects that belong to the same **form•Z** group. When exporting to DWG, DXF, FACT, Lightscape, OBJ, RIB, 3DG, 3MF, and VRML, which provide grouping structures, these groups can be exported in a single file, or one file can be constructed per group. When more than one file is exported from a single **form•Z** project, they are placed in a folder.

**Grouping Method**: This pull down menu (Figure 3.13.2.3), contains five grouping methods.

- **Single Group**: With this method (default), all objects become a single group and one file is exported.

- **By Object**: With this option, each object becomes a separate group. When **Separate Files** is also on, then each object is placed into a separate file.

- **By Color**: When this method is selected, all **form•Z** objects or faces of the same color form a separate group. For file formats that do not carry color, such as OBJ, this method allows you to reassign uniform colors to groups of objects after they have been imported into other programs.

- **By Layer**: When this grouping method is selected, all **form•Z** objects that are on the same layer are grouped together, which allows you to preserve the layers set up in the **form•Z** file.

- **By Group**: Selection of this method preserves the **form•Z** groups and transforms them into the grouping structure of the destination program, if one exists, or exports them as separate files.

**Separate Files**: When this option is selected, a separate file is constructed for each of the groups defined by the selection from the **Grouping Method** menu. These separate files are all saved into a newly created folder, which takes its name from the name entered into the name field of the **Save** dialog. The names of the individual files are constructed from the grouping method and an extension that specifies the type of file. For the first part of the name, if the grouping method is **By Object**, the object name is used; if **By Color**, the color name; if **By Layer**, the layer name; and if **By Group**, the group name is used. This option is dimmed and inactive when **Single Group** is selected.
Export method

This group of options specifies how the different types of form•Z objects will be exported.

Plain Objects: The items in this group specify how plain (non-controlled) objects are exported.

Facetted: This menu (Figure 3.13.2.4) controls how facetted objects are represented in the destination format. It has the following items:

As Object: Each form•Z object will be one object in the destination file. This is the default.

As Faces: Each face of a form•Z object will be one object in the destination file. If the format does not support a face or polygon object representation, the item is disabled.

As Polylines: When this item is selected, each face of a form•Z object will become one polyline object in the destination file. If a format does not support a polyline representation, the item is disabled.

As Lines: Each edge of a form•Z object will be one object in the destination file. If the format does not support a line object representation, the item is disabled.

As Points: Each point of a form•Z object will be one point object in the destination file. If the format does not support a point object representation, the item is disabled.

As Trimmed Surfaces: When this item is selected, each form•Z face becomes a trimmed surface in the destination file. If a format does not support a trimmed surface representation, this item is disabled.

Smooth: This menu controls how smooth objects are represented in the destination format. It has the following items (see Figure 3.12.2.5):

As Facetted: The smooth characteristics are ignored and the object is exported as a facetted object using the settings from the Facetted menu.

As Smooth Solids: Each smooth object is exported using the format’s smooth solid representation. If the format does not support a smooth solid representation, the item is disabled.

As Trimmed Surfaces: Each form•Z object will be one object in the destination file. If the format does not support a trimmed surface representation, the item is disabled.
Options...: Clicking on this invokes the Facetting Options dialog shown in Figure 3.13.2.6. It contains options for facetting smooth objects when they are exported.

Use Object’s Display Resolution: When this option is selected, each object’s display resolution is used to control facetting.

Override Objects’ Display Resolution: Selecting this option allows the user to control the facetting of all exported objects using the options available in the Facetting Options dialog. The display resolution options are discussed in section 4.1.1 of the form•Z User’s Manual.

Additional Facetting: When this option is selected, additional facetting is applied to the objects, as described in section 4.21.9 of the form•Z User’s Manual.
**Controlled Objects**: This pop up menu (Figure 3.13.2.7) specifies how controlled objects are exported. Note that if the given export format does not support a type of control object, then it will be handled as a plain object regardless of the menu setting.

**As Plain Objects**: When this item is selected, the parametric information is ignored and the object is exported as a plain object and is affected by all of the settings in the Plain Objects group.

**As Parametric Data**: When this item is selected, the controlled object is exported using its control parameters in the formats native representation.

**Export Visible Layers Only**: All the export formats allow you to export only parts of a project by organizing it into layers that may or may not be visible. When this option is on, only the objects on visible layers are exported.

**Picked Objects Only**: When this option is on, only the objects that appear selected on the screen will be exported.

**Transformation...**: Clicking on this button, which is available for all formats except STL, invokes the Import/Export Transformation dialog (Figure 3.13.1.6), which is the same dialog used with importing.

**Symbol Options...**: Clicking on this button invokes the Symbol Export Options dialog (Figure 3.13.2.8), which contains three options:

**Explode Symbols**: When this option is selected, all form•Z symbol instances are converted to common objects or elements before they are exported.

**Export Symbols And Referenced Definitions**: When this option is selected, all form•Z symbol instances are exported and all symbol definitions referenced by those instances are exported. Symbol definitions which are not referenced are not exported.

**Export Symbols And All Definitions**: When this option is selected, all form•Z symbol instances are exported and all symbol definitions, referenced and not referenced, are exported.
Decomposing faces

form•Z represents holes as true holes. In other 3D applications, especially rendering programs, surfaces that contain holes are typically decomposed and simulated through sets of triangles. Thus the faces of form•Z, especially those that contain holes need to be decomposed before exporting them. Options for two or three levels of decomposition are offered in all export dialogs, except for STL and 3DS, which only accept triangular faces, and Lightscape, which accepts quadrangles and triangles.

Decomposition: This section contains four options.

Connect Holes To Face Edges: When this option is on, form•Z connects holes to their outer face boundaries so that a single polygon is constructed from the original face. This is illustrated in Figures 3.13.2.9 and 3.13.2.10.

Subdivide Concave Faces: When this option is on, all concave faces are decomposed to convex pieces. A shape is concave when a line that is tangent to more than one of its points can be drawn.

Triangulate Faces: When this option is on, faces are decomposed to triangular pieces. This option is only available when the previous is on. It is dimmed otherwise. Thus concave faces are subdivided first, then triangulated.

The process of triangulating a concave face with a hole is shown in Figure 3.13.2.10. Given a concave shape with a hole, as in 3.13.2.10(a), the first step connects the hole to the boundary of the face (3.13.2.10(b)). Then the concave portions of the shape are decomposed to derive convex pieces (3.13.2.10(c)). The results for these two steps are also shown in exploded versions. Next, a decomposition to four-sided pieces process is applied, which is affected by the triangulation method selected through the Triangulate Options dialog. Finally, triangulation is applied, if the Triangulate Faces option is selected. However, the settings in the Triangulate Options dialog are also taken into consideration.

Triangulation Options...: This radio button complements the Triangulate Faces option and clicking on it invokes the Triangulate Options dialog shown in Figure 3.13.2.11. The triangulation is affected by the options set in this dialog. Note that, unless the Triangulate All Faces option is on, only non planar faces will be triangulated.
Exporting colors and rendering parameters

In *form•Z*, different colors may be assigned at the object and face levels, as well as rendering parameters relating to the smooth shading of surfaces. In *form•Z RenderZone Plus* surface style parameters and texture maps may also be assigned. The ability to export these parameters is available for many of the export formats using the following options.

**Attributes**: This section contains options that specify how colors, normals, shading, and textures are exported, as follows:

- **Preserve Face Colors**: When this option is selected, the polygons take their colors from face colors, if such color assignments have been made. If not, they take their colors from the object, which is also the case when this option is not selected. When independent colors have been assigned to faces and this option is selected, polygons resulting from faces with the same color are also grouped as independent entities. By default this option is on.

- **Include Normals**: When this option is selected, a list that contains the normal vectors associated with the vertices of the polygons in the model is also generated, and is included in the export file. These normals are used for smooth shading, and using this option results in better shading, especially around holes and openings.

- **Fix Smooth Shading**: Rendering applications apply smooth shading across surfaces that share common edges. When smooth shading between two neighboring surfaces is not desired, these surfaces should be disconnected and represented as independent surfaces. To achieve these effects, a *form•Z* model can be decomposed into parts using modeling tools available in the program. This option can also be used to determine which faces will be exported as continuous surfaces (and are thus smoothly shaded), and which faces will be exported as independent surfaces. The distinction is made on the basis of the angle at the common edge of two surfaces, which is entered in the **Angle** field, typically found next to this option in the export dialogs. When this option is selected, only the faces whose angle is larger than that shown are exported as continuous surfaces and are smooth shaded by the rendering application. Neighboring faces with smaller angles are exported as discontinuous surfaces.

- **Textures and Options...**: This pop up menu and the dialog invoked from the button offer options for specifying how textures are exported. They are discussed in subsection 3.13.3. Note that these options are only available in *form•Z RenderZone Plus*. 


Exporting drafting data

form-Z can export drafting data to a variety of file formats, including three object formats: DWG, DXF, and IGES. These formats share some common export options, which are set in a section of the Draft Export Options dialog, as shown in Figure 3.13.2.12.

Export Dimensions As Symbols: When this option is selected, a draft dimension is exported as a symbol instance, and a symbol definition referenced by the instance. Since dimensions are not directly supported by some formats, a form-Z draft dimension must be converted to plain elements. The Export Dimensions As Blocks option allows for a grouping of these elements into a single entity.

Export Visible Layers Only: This option is the same as that in the Modeling Export Options dialogs.

Symbol Options...: Clicking on this radio button invokes the Symbol Export Options dialog (Figure 3.13.2.13), which contains three options.

Explode Symbols: When this option is selected, all form-Z symbol instances are converted to common elements before they are exported.

Export Symbols And Referenced Definitions: When this option is selected, all form-Z symbol instances are exported as instances, and all symbol definitions referenced by those instances are exported as definitions. Symbol definitions which are not referenced are not exported.

Export Symbols And All Definitions: When this option is selected, all form-Z symbol instances are exported as instances, and all symbol definitions, referenced and not referenced, are exported as definitions.

Attributes: The single option in this section contains a pop up menu, Entity Colors From, with two options available.

Element Colors: When this option is selected, an entity is assigned a color from the program’s color table that is the closest match with the element color used in form-Z.

Line Weight Colors: With this option on, the entity is assigned a color table index that corresponds the best to the element’s line weight color.
3.13.3 Importing and exporting texture and surface style parameters

In the RenderZone Plus version of form•Z, the ability to import and export texture mapping and other surface style parameters is available. Note that since the regular version of form•Z does not support texture mapping, this is not available. The formats that support such parameters are: FACT, OBJ (in conjunction with MTL files), Lightscape, RIB (export), VRML, Art•lantis, Lightwave, Shockwave W3D, Viewpoint, 3DMF, and 3DS.

Note that the type of texture mapping and surface parameters supported by different rendering programs and export/import formats vary widely. Consequently, not all the options available in form•Z RenderZone Plus can be exported or can only be exported after some adjustments are made. Most notably importing applications can accept only pre-captured (image based) textures and, in most cases, only the UV mapping type. This would normally exclude procedural textures and mapping types other than UV from being exported. However, form•Z RenderZone Plus offers the ability to convert procedural textures to images and all mapping types to UV coordinates before exporting.

The following table (Figure 3.13.3.1) summarizes the form•Z surface style and texture mapping parameters that can be exported for each of the formats that support such parameters. For the definitions of these parameters see the form•Z RenderZone Plus User’s Manual.

<table>
<thead>
<tr>
<th>Color</th>
<th>Art•lantis</th>
<th>FACT</th>
<th>OBJ</th>
<th>RIB</th>
<th>VRML</th>
<th>3DMF</th>
<th>3DS</th>
<th>Lightscape</th>
<th>W3D</th>
<th>VET</th>
<th>Lightwave</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain Color</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Image Map</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reflection/Transmission</th>
<th>Art•lantis</th>
<th>FACT</th>
<th>OBJ</th>
<th>RIB</th>
<th>VRML</th>
<th>3DMF</th>
<th>3DS</th>
<th>Lightscape</th>
<th>W3D</th>
<th>VET</th>
<th>Lightwave</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Reflection</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Diffuse Reflection</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Specular Reflection</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Roughness</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Specular Color</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Transmission</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Refraction</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Reflectivity</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transparency</th>
<th>Art•lantis</th>
<th>FACT</th>
<th>OBJ</th>
<th>RIB</th>
<th>VRML</th>
<th>3DMF</th>
<th>3DS</th>
<th>Lightscape</th>
<th>W3D</th>
<th>VET</th>
<th>Lightwave</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transparency Map</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bumps</th>
<th>Art•lantis</th>
<th>FACT</th>
<th>OBJ</th>
<th>RIB</th>
<th>VRML</th>
<th>3DMF</th>
<th>3DS</th>
<th>Lightscape</th>
<th>W3D</th>
<th>VET</th>
<th>Lightwave</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bump Amplitude</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Bump Map</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

* Note: Transferred with the image map only.

Figure 3.13.3.1: The import/export texture parameters that each file format supports.
Importing textures

For importing texture parameters, the **Modeling Import Options** dialogs of the formats that support textures include the **Import Texture Maps** check box. When on, clicking on the **Texture Options...** button next to it invokes the **Texture Map Import Options** dialog, shown in Figure 3.13.3.3.

The **Texture Map Import Options** dialog sets the destination location of the support texture maps. It also sets the source location, if the texture is not found in the source location specified in the **Project: File Search Paths** section of the **Preferences** dialog. These locations contain the texture maps of the imported file. The destination location is where the texture maps for the **form•Z** project are found, copied, or created.

**If Source Texture Not Found:** This group contains the options that determine what **form•Z** should do if a source texture map is not found in any location specified in the **Project: File Search Paths** section of the **Preferences** dialog. If a source texture map can not be found, the action **form•Z** takes is determined by the following options.

- **Ask For Path To Texture:** When this option is selected and a source texture map can not be found, **form•Z** will display a standard folder selection dialog. The location of the texture map file should be entered. This location will be added to the locations to search for the duration of this file import. It will not be added to the preferences. If the texture map file is not found in the supplied location, the texture is replaced with the default **form•Z** texture. This is the default option.

- **Edit Search Paths...**: This button brings up a dialog containing the **Project: File Search Paths** section of the **Preferences** dialog. This can be used to modify the search paths for the source texture maps. This will change the preferences.

- **Use Default Texture:** When this option is selected and a source texture map can not be found, the texture is replaced with the default **form•Z** texture.
**Project Texture Maps:** These options specify where the textures of the form-Z project are.

**Use Original File:** When this option is selected, the support textures of the project are expected to be in the same folder with that of the imported file, at the location specified in the **Imported File Texture Maps** section. When there is no such section, as with 3DMF, which has embedded textures, this option is dimmed.

**Copy To File:** When this option is selected, the imported textures are copied into new files, whose type is specified by the **Save Texture Maps As** menu. They are placed in the application folder or in a user defined location. These new texture map files are the ones that will be used by the form-Z project.

**Save Texture Maps As:** This pop up menu (Figure 3.13.3.4) is only available when **Copy To File** is selected. It allows you to select the image type for the newly created texture maps. The image type can be the same as the original file, or any of the image types supported by the particular format. In the latter case a conversion of the original type to the selected type is made.

**Use The Application’s Folder:** This option is only available when **Copy To File** is selected. When on, the textures used by the project are saved in the form-Z folder. This is the default.

**Custom:** Only available when **Copy To File** is on, the textures used by the project are saved in a location you specify, as with the same option in the **Imported File Texture Maps** section.
Exporting textures

For exporting texture mapping and surface styles parameters, the **Modeling Export Options** dialogs of the format files that support them include the **Textures** pop up menu shown in Figure 3.13.3.5. It contains three items. Note that this menu is only available in **RenderZone Plus**.

**Off**: When this item is selected, no texture map information will be exported.

**Wrapped**: When this item is selected, the images of textures and their mapping information is exported. Note that this option cannot handle procedural textures, but the next item in the pop up menu can by converting them to image based textures. Selection of **Wrapped** also activates the **Options...** button next to the **Textures** pop up menu. Clicking on it invokes the **Wrapped Textures Options** dialog, shown in Figure 3.13.3.6. Its options are as follows:

**Image File Format**: This menu lists all the available image formats that are supported by the object format. The image format that is selected from this menu will be used for all wrapped textures.

**Options...**: This button invokes the image export options for the selected format.

**Use Original Of Supported Formats**: When this option is selected, if an image file is of a format supported by the object file format, then the original file is used.

**Make A Copy**: When this option is selected, a copy of the file is made in the destination location. When this is not selected, the exported file references the original file.

**Rendered**: When this item of the **Textures** pop up menu is selected, each object will be rendered into a separate texture image map. It also activates the **Options...** button, which is unavailable when this option is off.

This option makes it possible to export textures to all 3D formats that support texture image maps with uv texture coordinate mapping.
File formats such as VRML which are geared toward interactive viewing of geometry, can be rendered without any lighting calculations. Note that certain rendering effects are view dependent. For example, specular highlights or reflections will change with a different view position. When the exported model is rendered from different viewpoints, those effects will not appear correct. For a more accurate display, rendered textures can also be generated as separate color, transparency and bump maps. In this case, each rendering effect is rendered into a separate texture.

The texture files created will be named as described in section 6.5 of the RenderZone Plus User's Manual and will be placed in the same folder as the exported 3D file.

**Options...**: Clicking on this button invokes the Rendered Textures Options dialog (Figure 3.13.3.7), where the parameters for rendering the textures are set.

**Render Composite Textures**: When this option is selected, the objects are rendered as they are in RenderZone Plus and the resulting image is stored in a single image. This image may contain rendering effects such as shadows, shading, reflections, and transparencies.

**Render Separate Textures**: When this option is selected, the selected texture map types will individually be generated. Note that textures will only be generated for the selected texture type if a texture of that type is assigned to the object's or any of the object's faces' surface styles.

**Color Map**: When this option is selected, color map texture images will be generated.

**Transparency Map**: When this option is selected, transparency map texture images will be generated.

**Bump Map**: When this option is selected, bump map texture images will be generated.

**Render All Objects**: When this option is selected, texture images are generated for all textured objects.

**Render Picked Objects**: When this option is selected, texture images are generated only for picked objects which are textured.

**Constant Texture Size**: When this option is selected, square texture images of the specified size will be generated for each object.
**Constant Texture Resolution**: When this option is selected, square texture images will be generated such that the pixel size for all textures will be roughly the same. This will create larger texture images for objects with a large surface area and small textures for objects with a small surface area. Due to distortions caused by the mapping of a texture to an object, the pixels sizes are only approximate.

**Nominal Pixel Size In World Space**: The approximate pixel size.

**Minimum Texture Size**: If a texture image generated for an object is smaller than this size, the texture will be scaled up to this size. In this case, the pixel size will be less than the size specified in the **Nominal Pixel Size In World Space** option.

**Maximum Texture Size**: If a texture image generated for an object is larger than this size, the texture will be scaled down to this size. In this case, the pixel size will be greater than the size specified in the **Nominal Pixel Size In World Space** option.

**Image File Format**: The generated texture images are saved in the format selected from this pop up menu.

**Options...**: Pressing this button invokes the export options dialog for the currently selected image format.
The rendered textures are stored in a folder, which has the same name as the active project, with “.txr” appended to the name. This folder is located in the same folder as the project file. If the current project has not been saved yet, the textures folder will be created in the application folder. The name of a composite texture is formed by the string “tex” followed by a unique number and the file extension. For example, “tex12.tga”. Separate textures are named, “col,” “trn,” and “bmp” followed by a unique number and the file extension. The color, transparency, and bump texture for the same face will share the same number.

Recall that, in addition to UV mapping, form•Z RenderZone Plus also supports flat, cylindrical, cubic, and spherical texture mapping types. Among the formats that support texture mapping parameters, only FACT, Lightscape, and Lightwave support mapping types other than UV. For these formats, texture mapping types other than UV that may be used in a form•Z RenderZone Plus project are first converted to UV coordinates and are exported as such. When this happens, the warning message in Figure 3.13.3.8 is displayed.

For FACT, Lightscape, and Lightwave, the texture mapping types are translated to their counterparts in that format, as closely as possible. Note that, once a texture mapping type, other than UV, is translated to UV coordinates and is exported, when it is re-imported into form•Z, the UV mapping can not be automatically converted back to the original mapping type. For example, saving to 3DMF a form•Z object that uses spherical texture mapping, causes this mapping to be converted and exported in UV coordinates. Importing this 3DMF file back into form•Z will produce an object with UV texture mapping.
Tip and warnings

The following are some recommendations that will produce the best results, when exporting/importing texture map parameters:

• When exporting, set the Export Method for Facetted to Object, Smooth to As Facetted, and Controlled Objects to As Plain Objects. When using these settings each texture group (group of faces to which the same texture has been assigned) is exported as a separate group of faces, which is desirable. The alternative settings produce less desirable results. That is, if you set Facetted to Faces, every face is exported with its own mapping controls. If you set Smooth to Trimmed Surfaces, the mapping controls are derived from default NURBS parameters for the surface and their conversion to UV may not always be accurate. Note that controlled meshes with texture mapping other than UV, drop their controls and are exported as plain meshes. The same applies to parametric primitives that carry incompatible texture map controls, i.e. a sphere with flat mapping or a cylinder with a cylindrical texture map rotated about X by more than 0°.

• When importing, click the Add To Project option on, select File’s Palette from the Color menu. Only this option produces correct image maps. Recall that the first option in the Color menu uses the current color which is not desirable.

When importing a file without importing its textures, only the file you select is imported. When you import with the Import Texture Maps option on, in addition to the main file, a number of other support files, mostly image maps, also need to be imported. Texture map importing relies heavily on the validity of these support resources. Make sure that all the support files (image maps and material files) required by the main file are present; that they are compatible with the main file; that they are of an image type recognized by form-Z, if they are images; and that they have correct file names. When an image map can not be located at import it is substituted by a default “form-Z logo” image. If one or more of the above mentioned conditions are not satisfied, error messages are issued and importing may terminate. Needless to say that when you have an incomplete set of support files, the main file can still be imported by turning off the Import Texture Maps option.

When exporting spherical or cylindrical texture maps applied to relative large surfaces, which are converted to UV coordinates, the exported models may not render accurately in other applications. You may improve the rendering quality these applications can produce from such maps by increasing the resolution of the form-Z models, using one of the Mesh tools. When special cases are encountered as you import/export files with texture maps, you are warned at the end of the operation. These cases include file name changes due to file format syntax restrictions, image map conversions, failure to locate image maps, or exporting files with unsupported image formats. Also note that decals are not imported/exported with file formats, when using the “wrapped” method. The following discussion refers to the particular file formats that are affected by the texture import and export options. Recall that the parameters supported by each format have been summarized in the table of Figure 3.13.3.1.
3.13.4 3DGF

3DGF is a file format created by Macromedia, Inc. and is intended to be used for the interchange of 3D data. A typical 3DGF file contains polygon lists and color information and may contain representations of surface meshes. While a polygon list may be used to represent the faces of a solid, it also may be used to represent any collection of polygons. When such lists are imported, form•Z constructs solid objects from them, if they constitute closed collections of polygons. It constructs surface objects otherwise. When form•Z objects are exported to 3DGF, they are exported as polygon lists and may or may not be triangulated. In general, the process of exporting to 3DGF is very similar to that used for DXF, including the automatic connection of the holes.

3DGF offers some limited means for exporting c-meshes. To be able to export them, meshes must satisfy the following conditions:

1. Open c-meshes must be open and closed c-meshes must be closed in both directions (length and depth).
2. Open c-meshes must be of type Broken Bezier and closed c-meshes must be B-Splines in both directions.
3. Degree must be 3 (cubic) for both length and depth.
4. The number of control points for both depth and length must be a multiple of 3 plus 1.

Importing 3DGF files

A 3DGF file is imported into form•Z in the standard manner (using the Import... item of the File menu). All the options in the Modeling Import Options: 3DGF dialog that is invoked are the common options discussed in subsection 3.13.1.

Exporting 3DGF Files

The contents of a form•Z project are exported to a 3DGF file in the standard manner. Note that drafting elements cannot be exported to 3DGF files.

All the options in the Modeling Export Options: 3DGF dialog are the common options discussed in subsection 3.13.2. However, note that 3DGF does not support grouping structures. When exporting to 3DGF, form•Z combines the geometry of grouped objects into one to simulate the type of grouping that may have been selected.
**3.13.5 3DMF**

QuickDraw 3D Object Metafile (3DMF) is a file format developed by Apple to support the exchange of 3D data.

**Importing 3DMF files**

A 3DMF file is imported into form•Z in the standard manner and all the options in the **Modeling Import Options: 3DMF** dialog are as discussed in subsection 3.13.1.

**Exporting 3DMF files**

The content of a form•Z project is exported as a 3DMF file in the standard manner. Only modeling objects can be exported to 3DMF. The **Modeling Export Options: 3DMF** dialog that is invoked (Figure 3.13.5.1) contains the common options discussed in subsection 3.13.2 plus one 3DMF specific option.

**Make Trimeshes**: When this option is on, solid and surface objects are exported as trimeshes, and the 3DMF file is marked as version 1.5. Otherwise the solids and surfaces are saved as meshes, and the 3DMF file is marked as version 1.0. This option is only available when the **As Object** is selected from the Facetted pop up menu of the **Modeling Export Options** dialog.

**How are textures handled**

3DMF supports plain color and image maps, which are internal to the 3DMF files. At import, the **Imported File Texture Maps** section of the **Texture Map Image Options** dialog is not available, since the support image maps are embedded in the imported file. The image maps are converted according to the specified image type from the **Save Texture Maps As** pop up menu in the **Project Texture Maps** section of the dialog. The image maps used by the form•Z project are saved with the prefix “Image” followed by a unique number and a suffix denoting the file type. This happens because the internal (embedded) 3DMF image maps do not contain any names. One image file is saved per unique image location in the 3DMF file. At export, the maps referenced by the form•Z project are converted and embedded into the 3DMF file. Each map is only converted once and referenced thereafter.

The form•Z texture map controls are converted to UV coordinates at export and imported as UV coordinates from the 3DMF files. The 3DMF reader will recognize UV coordinates coming from meshes, trimeshes, trigrids, general polygons, and triangles. In addition it will associate parametric texture map controls with NURBS patches, cubic controls with boxes, spherical controls with ellipsoids and toruses, and cylindrical controls with cones and cylinders.
3.13.6 3DS

3DS is a file format used by Autodesk’s 3D Studio® to store modeling and animation information. A 3DS file can have an object section and a keyframe section, which transfers animation parameters. form•Z reads and writes only the object section of a 3DS file and does not support its animation portion. 3DS is a binary encoded format.

Importing 3DS files

A 3DS file is imported into form•Z in the standard manner and all the options in the Modeling Import Options: 3DS dialog are as discussed in subsection 3.13.1. The 3DS lights, cameras, and materials are converted as closely as possible to form•Z lights, views, and surface styles.

Exporting 3DS files

A form•Z project is exported as a 3DS file in the standard manner and only modeling objects can be exported to 3DS files. All the options in the Modeling Export Options: 3DS dialog are as discussed in subsection 3.13.2.

All the form•Z modeling objects are triangulated when they are exported to 3DS. Also, all the c-meshes and other control objects lose their controls, since they are not supported by 3DS. The form•Z lights, views, and surface styles are converted to the closest 3DS lights, cameras, and materials.

How are textures handled

The 3DS file format supports plain color, image maps, ambient, diffuse, and specular reflections, roughness, specular color, and transmission. It also supports transparency maps and bump maps.

When importing a 3DS file, and the file contains references to texture maps, all the image types recognized by form•Z are considered valid texture maps.

The form•Z texture map controls are converted to UV coordinates (or rendering coordinates in 3DS terms) at export and are imported as UV coordinates from the 3DS files.
3.13.7 Art•lantis

This is a native format of Art•lantis Render, a rendering/animation application by ABVENT, available on both MacOS and Windows. form•Z supports versions 3.0 and 3.5. This format is comprised of two ASCII files that typically have the same name with different extensions. The first file’s extension is .OPT. It contains information about window sizes, animation paths, cameras, rendering options, and a link to a second file, whose extension is .DB. This contains descriptions of objects such as geometry, shaders attached to faces, and lights.

The Art•lantis file format supports plain color, image maps, diffuse and specular reflections, transmission, refraction, and reflectivity.

**Importing Art•lantis files**

There is a single format specific option for importing (Figure 3.13.7.1). All the other options are as for the other formats (see section 3.13.1).

**Minimum Keyframes Per Bezier Segment:** Art•lantis and form•Z represent paths differently. Art•lantis uses end keyframes and a Bezier path stored with the animation. form•Z uses keyframes on a spline path that is not stored explicitly but is reconstructed from the keyframe positions. When importing Art•lantis paths, form•Z is required to generate the necessary keyframes per Art•lantis Bezier segment. How many frames are generated per segment is determined by the value entered in this field. Default is 3, the minimum number of controls needed for a Bezier segment.

![Modeling Import Options: Art•lantis dialog.](image)

*Figure 3.13.7.1:*
The **Modeling Import Options: Art•lantis** dialog.
Exporting Art-lantis files

Two format specific options, in addition to the common, are set when exporting Art-lantis. Also, the Grouping Method has a limited range, because Art-lantis does not support groups. Any group resulting from the method selected is exported as a separate file. That is, groups are simulated by exporting separate files.

Version: This menu specifies in which version of Art-lantis to save. The default is 4.5. Files saved in 3.0 can be read by Art-lantis version 3.0 and newer. However, v. 3.0 imposes stricter limits to the numbers of items that can be exported. If you hit such a limit, you should consider using v. 3.5 or v. 4.5.

In Art-lantis version 4.5, the file extension of that application was changed from .opt to .atl. If the version is changed from 3.x to 4.5 or from 4.5 to 3.x, after a file name has been entered, form-Z will automatically change the file extension.

Export Data File Only (.DB): When this option is on, only the .DB file is exported. When off (default), both files are saved. It is quite typical to preliminarily save an animation without the objects in the scene being completely developed yet. This allows you to check the animation paths, then finalize the details of the models, and then save the .DB file.
3.13.8 Collada

The Collada file format takes its name from Collaborative Design Activity and its extension is ".dae" that stands for digital asset exchange. This format may be used to transfer object geometry, textures, and other 3D data.

Collada can be used for both importing and exporting.

To import invoke the Collada Import Options dialog, which is a standard import dialog. That is, it only contains general type of options and has no Collada specific options. In this dialog, the Import Textures option is on by default.

To export invoke the Collada Export Options dialog, which is again a standard export dialog. That is, it only contains general type of options and has no Collada specific options.
3.13.9 DEM

Digital Elevation Model (DEM) files are distributed by the United States Geological Survey (USGS), and they consist of an array of sampled elevations for ground positions that are, normally, at regularly spaced intervals.

form•Z reads DEM data files (usually carrying the extension .cdo), and constructs a surface object out of them, which consists of quadrangles or is triangulated. Since DEM file sizes can easily grow to extensive sizes, form•Z offers the option to sub-sample the files or to extract only a portion of the total array of elevations. form•Z only imports DEM files.

A DEM file is imported into form•Z in the standard manner. The common options in the Modeling Import Options: DEM Data dialog (Figure 3.13.9.1) are as discussed in section 3.13.1. The dialog also contains a number of DEM specific options, as follows:

**Place At Origin:** If this option is selected (default), the array of elevations starts at the origin of the world axes. If it is not selected, the array starts at the true geographic coordinates.

**Triangulate Model:** If this option is selected the constructed surface is triangulated.

**Use Models Extends:** If this option is selected, the entire model is imported into form•Z.

**Use Part Of Model:** Only a portion of the elevation grid array is imported limited by the First/Last Column, First/Last Row fields. The default values in these fields are set for the full extends of a typical 1-Degree USGS Digital Elevation Model.

**Sub-sample:** When this option is selected (default) the imported data is sampled at a lower resolution, to reduce the amount of memory required for the constructed surface. The resolution can be constrained either by the Step that elevations will be sampled, or by limiting the Max Obj. Memory.

In the second case (default) a step is calculated automatically to fit the memory requirements.

![Figure 3.13.9.1: The Modeling Import Options: DEM Data dialog.](image)
3.13.10 DXF

DXF (Drawing Interchange Format) is the format in which AutoCAD® saves and exports its drawings. AutoCAD was initially an exclusively 2D drafting system. It was later extended to include 3D capabilities. However, its internal data structures and its output format retained, to a large extent, the characteristics of the original drafting oriented representations. The AutoCAD entities which appear as 3D objects when displayed, are in fact groups of separate surfaces. Furthermore, they are surfaces of at most four sides and they cannot contain holes.

DXF files may also contain block definitions and insertions. The blocks (called symbols by other programs) represent the ability to define entities which can then be placed as copies or instances, called insertions. An insertion is not a complete representation of the entity, but rather points to the definition of that entity in some central library of representations, called block definitions. A DXF file may contain block definitions that have been placed (inserted) one or more times, as well as definitions that have not been placed. A placed block definition is called a referenced block. Otherwise, it is called a non-referenced block.

Symbol definitions and symbol instances in form•Z are the equivalent to DXF blocks and insertions. When importing DXF data into form•Z, DXF blocks and insertions may be brought in as common objects or elements, or may be converted to form•Z symbol definitions and symbol instances. The respective options are set in the Symbol Import Options dialog, which can be invoked from the Modeling/Drafting Import Options: DXF dialog. Since form•Z symbol definitions are stored in a separate library file, this dialog also allows the user to select an existing library or to create a new library file to which the converted DXF blocks will be added. When exporting a form•Z project to a DXF file, form•Z symbol definitions and symbol instances can be saved as DXF blocks and insertions, or exploded into common objects or elements before exporting. The respective options are set in the Symbol Export Options dialog, which can be invoked from the Drafting Export Options: DXF dialog or the Modeling Export Options: DXF dialog.

When exporting a 3D model to a DXF file and the Export Method for Facetted is not Object, each solid is decomposed to its faces and each face is exported as a separate surface. Faces that consist of at most four sides and contain no holes, can be exported directly as they are. However, faces with more than four points or faces with holes have to be decomposed to the type of surfaces that DXF can handle.

Contrary to the modeling environment, the importation of a DXF file into the drafting environment of form•Z and the exportation of the content of a drafting window as a DXF file is straightforward. This is because the drafting representations of form•Z are inherently similar to those handled by DXF and require no special processing.
Importing DXF files

A DXF file is imported into form•Z in the standard manner and all the common options in the Modeling Import Options: DXF dialog (Figure 3.13.10.1) are as discussed in subsection 3.13.1.

Form Object By Layer (Polylines, 3D Faces): When this option is on, simple objects are constructed from all the polylines and 3D faces on a layer.

Import Hatch Patterns: When this option is selected, hatch entities are read as individual lines. This can result in a large number of line objects/elements being generated. If this option is not selected, hatch patterns are not imported.

Make Smooth Objects: When this option is selected, objects which can be represented as smooth objects (e.g. splines, arcs, circles, etc.) will be imported as smooth objects. Selecting this option for a large file can greatly increase the import time. If this option is not selected, all entities except REGION, SOLID3D, and BODY will be imported as faceted.

Explode Dimensions: When this option is selected in the Draft Import Options: DXF dialog, dimensions will be exploded into text, and lines. This has the advantage that the imported geometry of the dimension will match the geometry displayed in AutoCAD. However, the imported entity will not be a dimension object in form•Z. If this option is not selected, AutoCAD dimensions will be imported as form•Z dimensions. However the dimensions will not be drawn exactly as in AutoCAD.

Figure 3.13.10.1: The Modeling Import Options: DXF dialog.

Figure 3.13.10.2: The Draft Import Options: DXF dialog.
Exporting DXF files

The content of a form-Z project is exported as a DXF file in the standard manner. Depending on whether the active window is a modeling or drafting window, form-Z invokes the **Modeling Export Options: DXF** or **Drafting Export Options: DXF** dialog, shown in Figures 3.13.10.3 and 3.13.10.4, respectively. The common export options are as discussed in section 3.13.2. There are also a number of DXF specific to exporting options.

**Export Facetted Objects As ACIS Objects:** When this option is selected, facetted objects are written to the DXF file as embedded ACIS entities rather than native DXF entities. Note that not all software that can read DXF files supports the ability to read the embedded ACIS entities. This option is off by default.

**Version:** This pop up menu (Figure 3.13.10.5), allows you to indicate which version of AutoCAD the DXF file will be exported to. The default is **2000**.

**Edge Visibility:** This pull down menu (Figure 3.13.10.6) contains three options that control the visibility of the edges of the exported 3D faces.

**All Visible:** Selecting this option causes all the edges, both the original edges and the edges resulting from the splitting, to be marked as visible.
**All Invisible:** Selecting this option causes all the edges, both the original edges and the edges resulting from the splitting, to be marked as invisible.

**Actual Invisible/Split Invisible:** This is the default option and causes the split edges to be invisible, while the original edges of the faces remain visible.

**Export Groups As Blocks:** When this option is selected, form•Z groups will be exported as blocks. The groups are defined by the Grouping Method in the common options (e.g. if the Grouping Method is set to By Color, objects of the same color will be considered a group and will be exported as a block). To export groups, defined in the object palette as blocks, set the Grouping Method to By Group.

**Explode Dimensions:** When this option is selected in the Drafting Export Options: DXF dialog, dimensions will be exploded into text and lines. This has the advantage that the exported geometry of the dimension will match the geometry displayed in form•Z. However, the exported entity will not be a dimension object in AutoCAD. If this option is not selected, dimensions will be exported as AutoCAD dimensions. However the dimensions will not be drawn in AutoCAD exactly as in form•Z.
3.13.11 DWG

DWG is the native format of AutoCAD®. While it is more efficient than DXF, the other AutoCAD format, the structure of its entities is quite similar to DXF’s. Consequently, the interface of the DWG translator is very similar to DXF’s and the following discussion makes frequent references to the ways DXF works.

Supported entity types

form•Z saves and reads DWG files in both the modeling and drafting environments. When exporting to DWG, a variety of versions can be selected. When importing, the information from the above versions is automatically extracted and placed in a form•Z project file. The following table summarizes the AutoCAD entity types supported by the form•Z DWG format translator:

<table>
<thead>
<tr>
<th>Entity types:</th>
<th>Importing</th>
<th>Exporting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>modeling</td>
<td>drafting</td>
</tr>
<tr>
<td>TEXT</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>MTEXT</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>INSERT</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>MININSERT</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>POLYLINES</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>3DFACE</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LINE</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>MLINE</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>CIRCLE</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>ARCS</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>ELLIPSE</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>DIMENSION</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LEADER</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>SOLID</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>TRACE</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>SPLINE</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>POINT</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>PROXY</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>HATCH</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>ACIS:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REGION</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>SOLID3D</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>BODY</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>tables:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAYER</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>BLOCK</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LINE TYPE</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>LIGHTS</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>VIEWS</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>VIEWPORTS</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

* Text entities are exported only when exporting to version 12. Otherwise text entities are not exported.
When exporting DWG files from modeling, what type of entity is exported also depends on selected export options. When exporting from drafting types of elements that do not have an equivalent in AutoCAD, they are transformed to the closest types of entities available. For example, form-Z rectangles are exported as polylines, dimensions are saved as a set of the graphic entities that comprise them, and hatches are exported as lines.

**Importing DWG files**

The import options are set in the **Modeling Import Options: DWG** dialog (Figure 3.13.11.1). The common importing options are discussed in section 3.13.1. Options specific to importing DWG files are found in the **DWG Options** section of the dialog. These options are identical to those used when importing DXF files.
Exporting DWG files

The export options are set in the **Modeling Export Options: DWG** dialog for modeling (Figure 3.13.11.3) and in the **Drafting Export Options: DWG** dialog (Figure 3.13.11.4) for drafting. The common options are discussed in section 3.13.2.

**Figure 3.13.11.3:**
The **Modeling Export Options: DWG** dialog.

**Figure 3.13.11.4:**
The **Drafting Export Options: DWG** dialog.
3.13.12 FACT

FACT is the file format used by ElectricImage™, a high end rendering and animation program. FACT files describe a model as a collection of polygons. Polygons are organized in groups. Once imported, ElectricImage can assign further attributes, such as textures or colors, to groups of polygons. Individual polygons cannot be manipulated. The polygons can be complex polygons or triangles (3-sided) and quadrangles (4-sided).

**Importing FACT files**

A FACT file is imported into form•Z in the standard manner. The **Modeling Import Options: FACT** dialog contains one FACT specific option and is shown in Figure 3.13.12.1, along with the common import options that are discussed in subsection 3.13.1.

**Keep Sub Polygons**: FACT surfaces may be represented as complex polygons consisting of more than four points or as sub-polygons which consist of at most four points (triangles and quadrangles). Complex polygonal faces are typically decomposed to sub-polygons by the ElectricImage Transporter utility. In a FACT file, a surface may be represented either as a complex polygon, or as a group of sub-polygons, or both representations may be simultaneously available. When only one of the two types of representation is available for a specific surface, then this representation is imported into form•Z. When both representations are available, only one is imported. If **Keep Sub-Polygons** is selected, then the sub-polygons of the surfaces will be imported and the complex polygonal representations will be discarded. The reverse happens when this option is not selected.

*Figure 3.13.12.1:* The **Modeling Import Options: FACT** dialog.
Exporting FACT files

The content of a form-Z project is exported as a FACT file in the standard manner. Drafting elements can not be exported to FACT files. The common options in the Modeling Export Options: FACT dialog are as discussed in section 3.13.2. There are also a few FACT specific options, as follows:

**Electric Image Version:** This menu determines the version of Electric Image with which the written FACT file will be compatible. The default is version 2.9.

**Include Hierarchy:** This option is only available when **Grouping Method** is set to **By Layer** or **By Group.** When this option is on, the grouping hierarchy is included in the FACT file. When forming groups by layer, the hierarchy is a single level deep, with each object as a child of the layer on which it is located. When forming groups by form-Z groups, the FACT hierarchy reflects the nested grouping of the objects in the form-Z project.

**Include Face Decomposition:** When this option is selected, all faces are decomposed into sub-polygons before they are exported. They are not decomposed otherwise. However, the holes, when they are present, are always connected to the outer outline. ElectricImage Camera will only render triangles and quadrangles. Thus, faces that are not decomposed when exported will have to be decomposed by the ElectricImage Transporter utility. When this option is selected, a few additional options determine how the faces will be decomposed.

**Include Group Centroids:** When this option is selected, the group centroids (average X, Y, and Z coordinates) are included in the FACT file. When this option is not selected, the world origin is used as the group reference point.

**Extended Coordinate Precision:** When this option is on, coordinate data is written as 10 byte floating point numbers. When this option is off, it is written as four byte floating point numbers. With **Extended Coordinate Precision** on, files are bigger but more accurate. If you plan to open the FACT file with form-Z or any other modeling application, this option will give you the best results. For rendering applications, the extended accuracy is usually not required.
How textures are handled

FACT files support, in addition to plain colors and image maps, roughness, specular color, transmission and refraction parameters. These are transferred with the image maps only. The format also supports bump amplitude and bump maps.

When importing a FACT file that contains references to texture maps, all the image types that are recognized by form•Z, on the platform on which it runs, are considered valid texture maps (in particular PICT and Targa are supported by both form•Z and ElectricImage). Texture maps that are in ElectricImage’s Image format have to be converted beforehand using ElectricImage’s Projector utility, and have to be associated manually with the form•Z surface styles.

The texture map control parameters that are converted are, origin and rotation. Convertible mapping types are flat, cubic, cylindrical, and spherical. Any other mapping type defaults to cubic. From the tiling parameters size, center (except vertical cylindrical), and mirror are converted.

Some other considerations, when converting FACT files with texture maps, are:

- A FACT file may have multiple references of the same texture map. When importing a FACT file, form•Z creates a unique surface style (containing maps) for every unique combination of an image map name and a bump map name encountered in the FACT file. If this combination of texture maps is referenced again in the FACT file, form•Z reuses the surface style already created.

- In form•Z the texture maps that are read from the FACT file are applied to the currently accessed FACT group.

- If a FACT texture map includes both an image map and a bump map, form•Z will convert only the controls of the image map; the bump map will share the same controls.

- For exported bump maps to render correctly in ElectricImage, they must be in the alpha channel of the bump map.

- Specular Color, Roughness, Transmission, and Refraction are only exported with surface styles which contain image maps.
3.13.13 Google Earth

This file format, which is similar to Collada, can be used for transferring Google Earth models. The Google Earth files have the extension ".kmz" and can be used for both importing and exporting.

To import invoke the **Google Earth Import Options** dialog, which is a standard import dialog. That is, it only contains general type of options and has no Google Earth specific options. In this dialog, the **Import Textures** option is on by default.

To export invoke the **Google Earth Export Options** dialog, which is a standard export dialog with two additional Google Earth specific options:

**Latitude** and **Longitude**: The values entered in these fields represent the location in Google Earth where the exported model will be placed.

3.13.14 Google SketchUp

This file format is only available for importing and is used for bringing into form•Z models from SketchUp. Note however, that not all SketchUp models can come into form•Z as solid models. form•Z will do its best to convert the information imported from SketchUp to solids, but it is not always successful due to the fact that some objects may be incompletely enclosed.

The Google SketchUp files use the extension ".skp."

To import a SketchUp file invoke the **SketchUp Import Options** dialog, which is a standard import dialog. That is, it only contains general type of options and has no SketchUp specific options. In this dialog, the **Import Textures** option is on by default.
3.13.15 IGES

The Initial Graphics Exchange Specification (IGES) is a file format intended to exchange data for drafting, modeling, and manufacturing. Unlike other formats, IGES is not directly related to one particular application, but is designed to allow the transfer of data between different systems with as little loss of information as possible. In an IGES file, information is stored as entities. These are organized in classes of geometric, annotational, associativity, and property entities. When data is saved from a program to an IGES file, each piece of information from the exporting system is translated into the most closely matching IGES entity. When an IGES file is imported, the reading system must find the best interpretation of the IGES entities in its own internal representations. Version 5.1 of IGES supports 91 different entity types. Commonly, a program that reads an IGES file skips the entities it cannot use, if a particular entity does not apply to the type of the system, or if the system does not have the structure to handle it. For example, the IGES entity “Linear Dimension” may be skipped by a system which does not have drafting capabilities.

Importing IGES files

An IGES file is imported into form•Z in the standard manner. The Modeling Import Options: IGES dialog (Figure 3.13.15.1) contains common import options, which are as discussed in section 3.13.1, and IGES specific options.

The IGES entities that form•Z can read are summarized in Figure 3.13.15.2. They are sorted by entity number. The table shows what type of an entity is created by form•Z for each IGES type. Certain entities, such as the Transformation Matrix Entity (#124), do not exist by themselves and do not generate any form•Z element. However, they are necessary for the construction of other entities. Entities not listed in the table are skipped when an IGES file is imported.

The IGES Log File contains more detailed information. The IGES log file is always located in the form•Z application folder and is named “iges.log”.

Heal Corrupt Geometry: When this option is on, the translator detects and corrects accuracy/tolerance discrepancies that may be encountered in the imported data. This option is on by default.
### Figure 3.13.15.2

The IGES entities that can be imported and the corresponding *form-Z* entities.

<table>
<thead>
<tr>
<th>Ent #</th>
<th>Entity name</th>
<th><em>form-Z Model</em></th>
<th><em>form-Z Draft</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Circular Arc</td>
<td>Surface/Wire</td>
<td>Arc/Circle</td>
</tr>
<tr>
<td>102</td>
<td>Composite Curve</td>
<td>Surface/Wire</td>
<td>Polyline</td>
</tr>
<tr>
<td>104</td>
<td>Conic Arc</td>
<td>Surface/Wire</td>
<td>Ellipse/Polyline</td>
</tr>
<tr>
<td>106</td>
<td>Copious Data</td>
<td>Surface/Wire</td>
<td>Polyline</td>
</tr>
<tr>
<td>108</td>
<td>Plane</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>110</td>
<td>Line</td>
<td>Open Wire</td>
<td>Polyline</td>
</tr>
<tr>
<td>112</td>
<td>Parametric Spline Curve</td>
<td>Surface/Wire</td>
<td>Polyline</td>
</tr>
<tr>
<td>114</td>
<td>Parametric Spline Surface</td>
<td>Surface</td>
<td>Polyline</td>
</tr>
<tr>
<td>116</td>
<td>Point</td>
<td>Point</td>
<td>Point</td>
</tr>
<tr>
<td>118</td>
<td>Ruled Surface</td>
<td>Surface</td>
<td>Polyline</td>
</tr>
<tr>
<td>120</td>
<td>Surface of Revolution</td>
<td>Surface/Solid</td>
<td>Polylines</td>
</tr>
<tr>
<td>122</td>
<td>Tabulated Cylinder</td>
<td>Surface/Solid</td>
<td>Polylines</td>
</tr>
<tr>
<td>123</td>
<td>Direction</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>124</td>
<td>Trans.Matrix</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>126</td>
<td>Rat. B-Spline Curve</td>
<td>Nurbz Curve</td>
<td>Polyline</td>
</tr>
<tr>
<td>128</td>
<td>Rat. B-Spline Surface</td>
<td>C-Mesh(Nurbs)/Nurbz</td>
<td>Polylines</td>
</tr>
<tr>
<td>130</td>
<td>Offset Curve</td>
<td>Surface/Wire</td>
<td>Polyline</td>
</tr>
<tr>
<td>140</td>
<td>Offset Surface</td>
<td>Surface/Wire</td>
<td>Polyline</td>
</tr>
<tr>
<td>142</td>
<td>Curve on a Parametric Surface</td>
<td>Surface</td>
<td>Polyline</td>
</tr>
<tr>
<td>144</td>
<td>Trimmed Surface</td>
<td>Surface</td>
<td>Polylines</td>
</tr>
<tr>
<td>150</td>
<td>Block</td>
<td>Solid</td>
<td>Polylines</td>
</tr>
<tr>
<td>152</td>
<td>Right Angular Wedge</td>
<td>Solid</td>
<td>Polylines</td>
</tr>
<tr>
<td>154</td>
<td>Right Circular Cylinder</td>
<td>Solid/Cylinder</td>
<td>Polylines</td>
</tr>
<tr>
<td>156</td>
<td>Right Circular Cone</td>
<td>Solid/Cone</td>
<td>Polylines</td>
</tr>
<tr>
<td>158</td>
<td>Sphere</td>
<td>Solid/Sphere</td>
<td>Polylines</td>
</tr>
<tr>
<td>160</td>
<td>Torus</td>
<td>Solid/Torus</td>
<td>Polylines</td>
</tr>
<tr>
<td>162</td>
<td>Solid of Revolution</td>
<td>Solid/Revolved Obj.</td>
<td>Polylines</td>
</tr>
<tr>
<td>164</td>
<td>Solid of Linear Extrusion</td>
<td>Solid</td>
<td>Polylines</td>
</tr>
<tr>
<td>166</td>
<td>Ellipsoid</td>
<td>Solid/Sphere</td>
<td>Polylines</td>
</tr>
<tr>
<td>186</td>
<td>Manifold Solid B-Rep Object</td>
<td>Solid</td>
<td>Polylines</td>
</tr>
<tr>
<td>190</td>
<td>Plane Surface</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>202</td>
<td>Angular Dimension</td>
<td>Skipped</td>
<td>Leaders/Text</td>
</tr>
<tr>
<td>206</td>
<td>Diameter Dimension</td>
<td>Skipped</td>
<td>Leaders/Text</td>
</tr>
<tr>
<td>208</td>
<td>Flag Note</td>
<td>Skipped</td>
<td>Text</td>
</tr>
<tr>
<td>210</td>
<td>General Label</td>
<td>Skipped</td>
<td>Leaders/Text</td>
</tr>
<tr>
<td>212</td>
<td>General Note</td>
<td>Skipped</td>
<td>Text</td>
</tr>
<tr>
<td>214</td>
<td>Leader</td>
<td>Skipped</td>
<td>Leader</td>
</tr>
<tr>
<td>216</td>
<td>Linear Dimension</td>
<td>Skipped</td>
<td>Leaders/Text</td>
</tr>
<tr>
<td>218</td>
<td>Ordinate Dimension</td>
<td>Skipped</td>
<td>Leaders/Text</td>
</tr>
<tr>
<td>220</td>
<td>Point Dimension</td>
<td>Skipped</td>
<td>Leaders/Text</td>
</tr>
<tr>
<td>222</td>
<td>Radius Dimension</td>
<td>Skipped</td>
<td>Leaders/Text</td>
</tr>
<tr>
<td>228</td>
<td>General Symbol</td>
<td>Skipped</td>
<td>Draft Elements</td>
</tr>
<tr>
<td>230</td>
<td>Sectioned Area</td>
<td>Skipped</td>
<td>Polylines with Hatch</td>
</tr>
<tr>
<td>308</td>
<td>Subfigure Definition</td>
<td>Model Objects</td>
<td>Symbol Definition</td>
</tr>
<tr>
<td>314</td>
<td>Color Definition</td>
<td>Color</td>
<td>Color</td>
</tr>
<tr>
<td>406</td>
<td>Definition Levels (Form 1)</td>
<td>Layer</td>
<td>Layer</td>
</tr>
<tr>
<td>408</td>
<td>Singular Subfigure Instance</td>
<td>Model Objects</td>
<td>Symbol Instance</td>
</tr>
<tr>
<td>412</td>
<td>Rectangular Array Subfigure Instance</td>
<td>Model Objects</td>
<td>Symbol Instances</td>
</tr>
<tr>
<td>502</td>
<td>Vertex</td>
<td>Solid Object Point</td>
<td>N/A</td>
</tr>
<tr>
<td>504</td>
<td>Edge</td>
<td>Solid Object Segment</td>
<td>N/A</td>
</tr>
<tr>
<td>508</td>
<td>Loop</td>
<td>Solid Object Outline</td>
<td>N/A</td>
</tr>
<tr>
<td>510</td>
<td>Face</td>
<td>Solid Object Face</td>
<td>N/A</td>
</tr>
<tr>
<td>514</td>
<td>Shell</td>
<td>Solid Object Volume</td>
<td>N/A</td>
</tr>
</tbody>
</table>
In IGES, most entities are described in a parametric form. That is, instead of storing the detailed (evaluated) representation of an entity, IGES stores the parameters which can generate it. For example, the Cylinder Entity (#154) is represented by a radius, axis, and height parameter. Whether objects retain their parameters or not is controlled by the item selected in the Smooth Objects pop up menu, as discussed in subsection 3.13.1. Controlled objects get their attributes from the values set in the corresponding tool options dialog. Note that, while frequently all the parameters required for the generation of a form-Z entity are contained in the IGES file, there are also cases where parameters are missing and, to import them, parameters set in form-Z dialogs are used. Thus different variations of objects may be derived from the same IGES file, if different parameters are used. The following paragraphs summarize how IGES entities are interpreted, when imported into form-Z.

Circular Arc Entity (Type 100): The resolution of an Arc Entity, when imported into the modeling environment is determined by the settings in the Display Resolution dialog. Arcs, circles, and ellipses generated from IGES entities in the modeling and drafting environments maintain their parameters.

Conic Arc Entity (Type 104): These entities are used to generate ellipses, hyperbolas, and parabolas. The Display Resolution dialog is used to determine the resolution of these shapes. Ellipses generated in the modeling and drafting environments maintain the IGES parameters.

Parametric Spline Curve (Type 112): The spline curve is generated as an open or closed wire object. The number of its segments is given by the respective parameter in the IGES file.

Parametric Spline Surface (Type 114): The spline surface is generated as a surface object. The number of faces it consists of is given by the number of patches parameter in the IGES file.

Ruled Surface (Type 118): An IGES ruled surface is represented by two open or closed source shapes, corresponding points of which are connected in a manner that forms a surface. When imported into form-Z, an equal number of segments is first generated on both source shapes, if necessary. Then faces are created between corresponding segments of the source shapes. The result is a form-Z surface object.

Solid of Revolution Entity (Type 162), Surface of Revolution Entity (Type 120): In IGES, this entity is represented by the source shape, the axis, and the angle of revolution. To generate an object of revolution, form-Z requires a value for the density of the revolution. This value is provided by the # Of Steps parameter in the Revolve Options dialog. When the source shape is an open surface object whose end points do not lie on the axis of revolution, a surface solid is generated, otherwise a solid is generated. If the Preserve Controls is selected, the object becomes a parametric revolved object.

Tabulated Cylinder (Surface of Extrusion) (Type 122): This IGES entity is represented by the source shape, the direction, and distance of the extrusion. From this entity form-Z generates an extruded solid.
Rational B-Spline Curve Entity (Type 126): From this IGES entity form•Z generates a Nurbz Curve. The control points, weights, knots, and degree are provided by the IGES parameters.

Rational B-Spline Surface Entity (Type 128): From this IGES entity form•Z creates a controlled mesh (c-mesh) or a nurbz object, if the Preserve Nurbz option is on. The control lines, weights, knots, and degree are provided by the IGES representation.

Offset Curve (Type 130): This IGES entity is represented by the source shape, the type of offset curve, and the offset distance. form•Z creates an open or closed wire, by applying a uniform offset distance relative to the source shape. IGES offset curve entities with varying distances or with distances specified by functions are not processed.

Trimmed Surface (Type 143 or 144): This IGES entity is represented by the parameters of a surface, and the exterior and interior trimming curves. It is imported as a trimmed nurbz object.

Right Circular Cylinder (Type 154), Right Circular Cone (Type 156), Sphere (Type 158), Torus (Type 160), Solid of Revolution (Type 162), Ellipsoid (Type 168): The type of objects generated by form•Z for these IGES entities depends on the selection in the Smooth Objects pop up menu, as discussed in subsection 3.13.1.

Manifold Solid B-Rep Object (Type 186): The IGES MSBO entity most closely resembles the representation used for the form•Z objects. It consists of points, segments, outlines, and faces, and can be directly translated into form•Z objects. If the form•Z ACIS Module is not installed, form•Z will only import MSBO entities whose faces are planar. MSBO entities with non planar surfaces are skipped.

Angular Dimension (Type 202), Diameter Dimension (Type 206), Linear Dimension (Type 214), Ordinate Dimension (Type 218), Point Dimension (Type 220): These IGES dimension entities can only be imported into the drafting environment and are translated into leader and text elements. Since these entities do not include information about the elements which are dimensioned and with which they are associated, form•Z cannot generate true dimensions.

General Note (Type 212): This IGES entity is imported as a text element. The IGES parameter for the text font is ignored and the default font on your system is used.
Exporting IGES files

A form•Z project is exported as an IGES file in the standard manner. The Modeling or Drafting Export Options: IGES dialogs (Figure 3.13.15.3) contain common export options, found in the upper sections of the dialogs. They are as discussed in subsection 3.13.2. Options that are specific to IGES files are found in the IGES Options section of the dialog.

Certain items in the Facetted and Smooth pop up menus, both of which contain common export options, have the following specific effects when applied to IGES files:

**Facetted**: The items in this pop up menu apply to facetted solid surface objects only.

  **As Faces**: With this item selected, each outline is exported as Entity Type 106, Form 12 (Copious data). If the IGES specific option Make Composite Curves is on, each outline is exported as Entity 102 (composite curve) with each segment as Type 110 (straight line).

  **As Lines**: With this item selected, each segment is exported as Type 110 (straight line).

**Smooth**: The items in this pop up menu apply to smooth solid/surface objects only.

  **As Trimmed Surfaces**: With this item selected, each face is exported as Type 143 (trimmed surface). If the IGES specific option Export Trimmed Surfaces As 144/142 is selected, each face is exported as Type 144.

  **As Solids**: With this item selected, each smooth object is exported as Type 186 (manifold solid b-rep).

Closed and open wire objects are exported as Type 100 (arc), Type 110 (line), Type 104 (ellipse), or Type 126 (spline curve) with each segment as an independent entity. If the IGES specific option Make Composite Curves is on, each wire is exported as Type 102 (composite curve) consisting of sub-types 100, 110, 104, and 126.
Make Composite Curves: When this option is selected, the \textit{form-Z} objects are exported as composite curves. One Line entity (type 110) is created for each edge and one Composite Curve entity (type 102) is created for each outline of the object.

Export Color Table: When this option is selected, the \textit{form-Z} Surface Style palette is also saved in the IGES file and the exported entities retain their original colors. Otherwise, the IGES base color that is the closest to the \textit{form-Z} color is assigned to the exported entity. The IGES base colors are: black, white, red, green, blue, magenta, cyan, and yellow.

Export Trimmed Surfaces As 144/142: The IGES format contains two different representations of trimmed surfaces: 144/142, and 143/141. Some programs only support one of these formats when reading IGES files. Select the proper setting based on which type of IGES entities the destination program supports. When this option is on, trimmed surfaces are exported using the 144/142 format. When this option is off, which is the default, the 143/141 format is used. This option is only applied when the \textit{As Trimmed Surfaces} option is selected from the \textit{Smooth} menu.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
\textbf{Object Type} & \textbf{IGES Representation} \\
\hline
\textit{C-Curve NURBS, Nurbz Curve, Tangent Curve, Spline Curve} & one Rational B-Spline (Type 126) \\
\hline
\textit{C-Curve B-Spline} & one Rational B-Spline (Type 126) with equal weights and default knots, where degree = number of control points - 1 \\
\hline
\textit{C-Curve Bezier} & one Rational B-Spline (Type 126) for each (degree + 1) set of control points with equal weights and default knots, using the midpoints of the control lines as the control points, except for the first and last point \\
\hline
\textit{C-Curve Continuous Bezier} & one Rational B-Spline (Type 126) for each (degree + 1) set of control points with equal weights and default knots \\
\hline
\textit{C-Curve Broken Bezier} & one Rational B-Spline (Type 126) of degree 2 with equal weights and default knots \\
\hline
\textit{C-Curve Quick Quadratic} & handled as a Continuous Bezier Curve of degree 3 \\
\hline
\textit{C-Curve Quick Cubic} & one Rational B-Spline (Type 126) of degree 2 with one additional control point for each existing control point except the first and last. The control points are located such that the spline curve passes through the initial control points \\
\hline
\textit{Parametric Cube Block} & Block (Type 150) \\
\hline
\textit{Parametric Cone Right Circular Cone} & Right Circular Cone (Type 150) \\
\hline
\textit{Parametric Cylinder Right Circular Cylinder} & Right Circular Cylinder (Type 154) \\
\hline
\textit{Parametric Sphere Sphere} & Sphere (Type 158) \\
\hline
\textit{Parametric Torus Torus} & Torus (Type 160) \\
\hline
\end{tabular}
\end{table}

\textbf{Figure 3.13.15.4}: How different object types are translated into the IGES format.
The options in the Draft Export Options: IGES dialog (Figure 3.13.15.5) are similar to those in the dialog for exporting from modeling. The different types of drafting elements are translated into different IGES entities; each element is translated to its most similar type of IGES entity. Figure 3.13.15.6 summarizes how each drafting element is translated into IGES.

<table>
<thead>
<tr>
<th>Draft Element</th>
<th>IGES Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangle</td>
<td>one Copious Data (type 106, form 12)</td>
</tr>
<tr>
<td>Polygon</td>
<td>one Copious Data</td>
</tr>
<tr>
<td>Circle</td>
<td>one Circular Arc (type 100)</td>
</tr>
<tr>
<td>Arc</td>
<td>one Circular Arc</td>
</tr>
<tr>
<td>Ellipse</td>
<td>one Conical Arc (type 104, form 1)</td>
</tr>
<tr>
<td>Polyline</td>
<td>one Copious Data</td>
</tr>
<tr>
<td>Text</td>
<td>one General Note (type 212) for each line in the text</td>
</tr>
<tr>
<td>Leader</td>
<td>one Leader (type 214)</td>
</tr>
<tr>
<td>Dimension</td>
<td>one General Note, two leaders (type 214)</td>
</tr>
<tr>
<td>Symbol Definition</td>
<td>one Subfigure Definition (type 308)</td>
</tr>
<tr>
<td>Symbol Instance</td>
<td>one Subfigure Instance (type 408)</td>
</tr>
</tbody>
</table>

Figure 3.13.15.6: How different drafting elements translate into the IGES format.
3.13.16 Lightscape

Lightscape is a 3D rendering application, which specializes on generating radiosity based images like form•Z RenderZone Plus. form•Z exports modeling data and lighting information to Lightscape’s preparation file format. If a file is exported from form•Z RenderZone Plus, the compatible radiosity parameters are also exported. The Lightscape files extension is .lp.

Importing Lightscape files

The options for importing are set in the Modeling Import Options: Lightscape dialog, which only contains the common import options discussed in section 3.13.1. When importing into form•Z RenderZone Plus, some of the radiosity parameters in the Lightscape file are also imported and cause changes to the following parameters of form•Z dialogs:

- In the Radiosity Options dialog: Adaptive Meshing On/Off, Apply Reflected Light As Ambient, and Adaptive Meshing Threshold.
- In the Radiosity Preview Options dialog: Update After n Iterations.

Exporting Lightscape files

The export options are set in the Modeling Export Options: Lightscape dialog (Figure 3.13.16.1). All the common export options are as discussed in subsection 3.13.2. For Lightscape, objects are always exported as four or three sided polygons. The latter are generated when Triangulate Faces is on and the Triangulate All Faces option is on in the Triangulate Options dialog. There is one format specific option:

- Export All Views As Separate Files: When this options is on, all views currently defined in the views palette are exported to separate view files. The view files have the same name as the corresponding view plus a .VW extension. They are located in the same directory as the Lightscape file.

Note, that isometric, oblique, straight up perspective and panoramic views are not supported by Lightscape and are converted to normal perspective views. Axonometric views are only supported if they are a top, bottom, front, back, right, or left view (orthographic projections). Axonometric views from any other direction are also converted to perspective views. The current view is always saved in the Lightscape (.lp) file itself.

**Figure 3.13.16.1:** The Modeling Export Options: Lightscape dialog.
Exporting lights
• Lightscape only supports area lights that consist of a single polygon. Consequently, when form-Z area lights are exported to Lightscape, they are first decomposed to triangles, each of which is then exported as a separate area light in the Lightscape file.
• Projector lights are not supported in Lightscape and are not exported.
• The Intensity distribution of a custom light is written as a separate photometric data file in IES format. If an IES file with the same name already exists, it will be overwritten without warning.
• Lightscape supports only one distant light (the sun). Consequently only the distant light flagged in form-Z as the sun is exported. All other distant lights are ignored.
• Lightscape specifies the output power of lights only in photometric units (Lumens). When exported to a Lightscape file, only the photometric light output quantity of a form-Z light is used, even if Radiometric is selected in the Radiosity Intensity dialog.

Exporting textures and surface styles
• Surface styles in Lightscape only support plain color and image textures.
• Flat, Cubic, and UV coordinates texture map controls are supported in Lightscape and are exported as such. Cylindrical and Spherical texture map controls exist in Lightscape, but are not compatible with the representation in form-Z. They are converted to UV coordinates.

Exporting symbols and parametric objects
• Symbol definitions in Lightscape are called blocks. Symbols and definitions in form-Z can be exported to Lightscape in the same fashion as to DXF. The name of a symbol definition exported as a Lightscape block is a composite of library name, definition name, and level number, separated by colons (for example: “furniture:chair:1”)
• Parametric objects are not supported in Lightscape and are written as polygons. If parametric mapping is used with an object, it is converted to UV coordinates.

Exporting radiosity parameters
• The form-Z RenderZone Plus parameters that are compatible with Lightscape parameters are exported as they are. The remaining Lightscape parameters are assigned default values.
• form-Z RenderZone Plus parameters that are also supported by Lightscape:
  • Radiosity Preview Options dialog: Update After n Iteration.

Exporting system colors
• Lightscape does not save its wire frame display color in the .lp file. By default it is white. However, the background color is saved. When exporting to a Lightscape file, the project’s background color is written to the file. If the background in form-Z is white, the wire frame display of the file after opening it in Lightscape will show white lines on a white background, resulting in a blank screen. You need to switch to Colored Wireframe in the Display menu, or change the wire frame color in the Properties dialog, invoked from the Edit menu in Lightscape.
• Lightscape supports form-Z’s Background Color, Fog Depth Effect on/off and Fog Color.
3.13.17 LightWave

LightWave is a modeling and animation application by NewTek. It is available for both MacOS and Windows. form•Z reads all versions of LightWave object data, but writes only in Version 6.

A LightWave project consists of a scene file (.lws) and one or more object files (.lwo). The scene file contains object connections, views/cameras, lighting, and animation information. The object files contain object, geometric, and surface texture information. form•Z can read and write the object files. The scene files are not documented by NewTek, hence form•Z can not read or write these files.

Importing LightWave files

There is one format specific option for importing LightWave files (Figure 3.13.17.1). All the other options are as for the other formats and as discussed in section 3.13.1.

Separate Objects By Volume: Selecting this option separates LightWave data into separate form•Z objects based on their volume. This operation is similar to that of the Separate tool (see section 4.18.2 in the form•Z User's Manual). The Separate Objects By Volume option is used to break the volumes of the model into separate objects so that they can be easily manipulated in form•Z.

In addition to plain color and image maps, other texture parameters supported are diffuse and specular reflections, transmission, refraction, and reflectivity. Also supported are transparency maps, bump amplitude, and bump maps.

Modeling Limitations

• LightWave’s MetaNURBS surfaces can not be imported.

Texture Map Limitations

• Procedural, layered, and shader-generated surface styles can not be imported. However, surfaces generated with the basic parameters of LightWave’s Surface Editor are supported. Surfaces that have textures applied to the color, transparency, and/or bump channels are supported as well.
• LightWave allows multiple texture controllers to contribute to an object’s surface style. **form•Z** does not support this. When importing, **form•Z** will look for one applied to the color, transparency, or bump shader (in that order) and use the first one it finds.

• LightWave offers five types of texture controllers. The UV map is fully supported. However, the others are not. Specifically, all non-UV texture controllers have an orientation controlled by the Rotation tab in the LightWave Texture Editor. Rotation values in excess of 60 degrees give unpredictable results. The import of texture maps into **form•Z** always causes at least one 90-degree rotation. This results in texture controllers that may not export back to LightWave properly.

  • The Falloff tab is not supported.
  • The use of Reference Objects is not supported.
  • Non-integer Width Wrap Amount values are not supported for Cylindrical and Spherical controllers.
  • Non-integer Height Wrap Amount values are not supported for Spherical texture controllers.
  • For Cubic texture controllers, LightWave uses three Size values. **form•Z** uses only the first two.
  • LightWave allows image-processing effects to be applied to images before they are used as Surface Styles. **form•Z** does not support this.
  • All Major Axis settings are supported, as well as settings for the Scale and Position tabs.

Guidelines for successfully “round-tripping” (exporting then importing, or vice versa) surface styles:

  • Do not use multiple texture controllers on a surface in LightWave.
  • Do not do texture-image-processing in LightWave.

### Exporting LightWave files

The LightWave export options include a format specific option, in addition to the common options discussed in subsection 3.13.2. The **Modeling Export Options: LightWave** dialog is shown in Figure 3.13.17.2.

**Make Quads**: When this option is selected, all the faces exported to the LightWave format will be quadulated (transformed to 4-sided polygons while triangular faces remain as they are).
Limitations

- All geometry data is saved as polygons.
- Texture controller limitations are as discussed for the import of LightWave object files.
- Not all Surface Style Shaders are supported when Textures is set to Wrapped.

The supported Shaders for this option are as follows:

- **Color**: Plain and Image Map.
- **Reflection**: Matte, Chrome, Constant, Metal (Simple), Plastic, and Glass (Simple).
- **Transparency**: None, Simple, and Transparency Map.
- **Bump**: None, Bump Map.

Guidelines for successfully “round-tripping” (exporting then importing, or vice versa) surface styles:

- Do not use procedural textures or decals.
- Do not use environment maps.
- Use UV coordinates for texture controllers whenever possible.
- Be prepared to reset all rotation values with each importing or exporting.
- Avoid cubic and spherical controllers.
3.13.18 OBJ

OBJ is a data file format used by Wavefront. It contains representations of the geometry and other properties of objects. It supports polygonal objects, controlled curves, and controlled meshes. OBJ files are always ASCII files.

Importing OBJ files

An OBJ file is imported into form•Z in the standard manner. The **Modeling Import Options: OBJ** dialog contains common options as discussed in subsection 3.13.1. It also contains one format specific option, **Use Absolute Indices**. The same option is also found in the **Export** dialog and is discussed below.

Exporting OBJ files

The content of a form•Z project is exported as an OBJ file in the standard manner. The **Modeling Export Options: OBJ** dialog, shown in Figure 3.13.18.1, contains all common options (see subsection 3.13.2) and one format specific option.

**Use Absolute Indices**: The OBJ format offers the option to export with either absolute or relative indices. When this option is off, which is the default, data will be exported with relative indices. If on, absolute indices will be exported. Note that some programs that import OBJ files only read absolute indices. Data intended for such programs should be exported with this option on.

How textures are handled

The OBJ format supports plain color, image maps, ambient, diffuse, and specular reflections, roughness, specular color, transmission, refraction, transparency maps, and bump maps.

When importing a MTL file associated with an OBJ file, and the material file contains references to texture maps, all the image types recognized by form•Z under the active platform are considered valid texture maps.

The form•Z texture map controls are converted to U,V coordinates at export and are imported as U,V coordinates from the OBJ files.

---

*Figure 3.13.18.1:* The **Modeling Export Options: OBJ** dialog.
3.13.19 PLY

The PLY format was initially created at the Stanford University Computer Graphics Laboratory. It has since become a file format used by 3D scanners, most notably Cyberware scanners. The PLY format is a simple polygonal format. It contains vertex locations, and colors and face definitions. Faces can have any number of vertices.

Importing PLY files

A PLY file is imported into form•Z in the standard manner. The only options in the **Modeling Import Options: PLY** dialog are the common options discussed in subsection 3.13.1.

Exporting PLY files

The contents of a form•Z project are exported to a PLY file in the standard manner. The only options in the **Modeling Export Options: PLY** dialog are the common options discussed in subsection 3.13.2. Note that drafting elements cannot be exported to PLY files.
3.13.20 RIB

RIB is the format used by Pixar’s RenderMan™, a high end rendering program. A RIB file contains a set of instructions describing the scene to be rendered and includes the lighting and the viewing parameters. The RIB file is not structured as a data file, the way DXF is, but describes the scene with a set of instructions. For example, one instruction may define the brightness of a light source, while another may specify the geometry of an object to be rendered. form•Z will export its models to the RIB format so that they may be rendered in RenderMan. The ability to import RIB files into form•Z is not currently available.

form•Z creates RIB files which contain descriptions of the objects to be rendered, viewing parameters, and the position of one light source, which corresponds to the sun position of the current form•Z window. The viewing parameters are generated from the view of the current window. These parameters can be further manipulated and more light sources may be added by taking the RIB file into Pixar’s Showplace, before rendering it with RenderMan. The form•Z objects are exported as bounded polygons. C-meshes and nurbz can be exported as bounded polygons or NURBS meshes. Consequently, form•Z c-meshes that use other types of curves are parametrically adjusted to NURBS before they are exported. This may affect the shape of the mesh. Analytic primitives can be exported with their controls preserved or as bounded polygons.

Exporting RIB files

The content of a form•Z project is exported as a RIB file in the standard manner. All the options in the Modeling Export Options: RIB dialog are as discussed in subsection 3.13.2.

How textures are handled

The RIB format supports plain color and image maps.

When exporting form•Z projects as RIB files and the Textures option is Wrapped or Rendered, the form•Z texture map controls are converted to U,V coordinates at export.
3.13.21 SAT and ACIS data

ACIS is a 3D solid and surface modeling library, published and marketed by Spatial Technology, Inc. It is used by a number of popular modeling programs as their primary modeling engine. The native file format of ACIS is called SAT and applications that incorporate ACIS can import and export it. ACIS data may also be embedded in DXF or DWG files. This is done by directly importing SAT files or by importing DXF or DWG files that contain ACIS entities.

The form•Z ACIS module is a library that contains the necessary functions for supporting smooth modeling and the SAT format. The smooth modeling module can be optionally installed with the form•Z Installer. It is installed by default. If this module is not installed, or if the module file has been moved or deleted from the application directory, the ACIS format is disabled.

ACIS supports two types of entities: solids/surfaces and wires. It represents its solids and surfaces by a boundary representation, using parametric surfaces. That is, objects are defined by a set of faces, whose surfaces may be planar, conical, cylindrical, spherical, toroidal, or spline based. Each face is delineated or “trimmed” by one or more curves, the first of which is the outer curve. The other curves are optional and represent holes. The edges which define the trimming curves may be straight lines, elliptical, or spline curves.

The ACIS wires are equivalent to the open or closed shapes (called surface objects) in form•Z. That is, they contain no surface information. The edges of wires may be straight lines, elliptical, or spline curves.

Importing SAT

SAT files are imported into form•Z in the standard manner. The Modeling Import Options: SAT dialog, shown in Figure 3.13.21.1, contains all common options (see subsection 3.13.1) and one format specific option:

Heal Corrupt Geometry: When this option is on, the translator detects and corrects accuracy/tolerance discrepancies that may be encountered in the imported data. This option is on by default.

Figure 3.13.21.1:
The Modeling Import Options: SAT dialog.
Exporting ACIS entities

The content of a form•Z project is exported as an SAT file in the standard manner. The **Modeling Export Options: SAT** dialog (Figure 3.13.21.2), in addition to the common export options (see subsection 3.13.2), it contains one format specific option:

**Version:** This pop up menu (Figure 3.13.21.3) allows you to choose which version of ACIS you would like to export as. The **Default** option will export using the latest ACIS version.

---

**Figure 3.13.21.2:**
The **Modeling Export Options: SAT** dialog.

**Figure 3.13.21.3:**
The **Version** menu.
3.13.22 STL

STL is a file format that is used by rapid prototyping machines for producing 3D physical models from computer models. The rapid prototyping process typically consists of two stages. A preprocessing computer is first used that reads the STL file, reconstructs the 3D model, and slices it into horizontal cross sections, at a thickness of approximately 1/1000". Then a manufacturing device is used to physically build each cross section. These physical “slices” are placed on top of each other in the order they are derived and then glued, resulting in a layered 3D model.

The rapid prototyping process imposes a series of restrictions on the 3D model to be manufactured. The model must be described by one or more closed volumes. Independent surfaces and overlapping volumes are not allowed. STL accepts triangular faces only; consequently, form•Z decomposes all the faces of an object into triangular faces before the object is exported. Each triangular face cannot exceed a given minimum size. The maximum size of the model is also limited by the manufacturing apparatus. Sometimes, support structures also need to be added to the model in order to prevent structurally weak parts from bending during the construction process. Once the model is finished, the support structures are usually removed.

Importing STL files

An STL file is imported into form•Z in the standard manner. The Modeling Import Options: STL dialog, shown in Figure 3.13.22.1, contains one set of format specific parameters.

**Scale Factor: X, Y, Z:** The scale factors entered in these fields are applied to the geometry of the imported model. If no scaling is needed, these values should be kept to 1.0, which are the default values.

![Figure 3.13.22.1:](image)

The Modeling Import Options: STL dialog.
Exporting STL files

The content of a form•Z project is exported as an STL file in the standard manner. When exporting to STL, the active window must be a modeling window, as drafting elements cannot be exported to STL. The desired options are set in the **Modeling Export Options: STL** dialog (Figure 3.13.22.2), which contains a number of format specific options.

**Scale**: Typically, rapid prototyping machines are limited to building models of a certain size, which may make it necessary to scale the computer model when it is exported. This can be done by selecting the **Scale** option and entering the appropriate scaling parameters.

- **Scale Factor**: When this option is selected, the scale factor contained in its field will be uniformly applied in all directions to the model.
- **Fit Box**: When this option is selected, the model is scaled to fit in a bounding box whose size is determined by the numeric values contained in the X, Y, and Z fields.

**Use Current Reference Plane As Base Plane**: Typically, the direction in which STL models are sliced is determined by the actual coordinates of the model. That is, the slices are parallel to the XY plane. This option allows you to manipulate the direction of the slicing. When selected, the model will be exported with its coordinates defined relative to the reference plane (the reference plane becomes the XY plane). This will result in slices parallel to the reference plane.

**Position In Positive XYZ Octant**: When this option is selected, the position of the model is automatically adjusted so that all its coordinates take positive values and no coordinate value is less than that specified in the **Min. Coord. Value** field. Some rapid prototyping systems require the model data to be located entirely in the positive XYZ octant. That is, they do not accept negative and zero coordinates.

**Supports Located On Layer**: Selection of this option informs form•Z that a support structure will be created and its description (model) is in the layer whose name appears in the field next to the option. All the objects found in the indicated layer are interpreted as support structures and are saved into a separate STL file, which is labeled by adding “sup” to the name used for the model file (for example: mod1.stl becomes mod1sup.stl). If the rapid prototyping process requires support structures, they need to be saved in a separate STL file, which is done automatically by form•Z. Note that the model itself should neither be nor have any parts on the layer where the support structures are. If it does, the result will be different from what was intended.
**Export Solids Only:** When this option is on, only solid objects are exported to the STL file. This option is on by default. When this option is off, solids, surfaces, and closed wire objects are exported.

Note that a properly formatted STL file should describe a solid object. This can be guaranteed by turning this option on. However, some rapid prototyping and 3D printing systems can, to a certain degree, deal with surfaces and have the ability to fix gaps or stitch surfaces together to produce solids. If this is not the case, the **Export Solids Only** option should be turned on.

**Diagnosis:** This group of options may be used to instruct the system to check your data before actually producing an STL file. The diagnostic process is applied as soon as you click on the **Diagnosis...** button. What is checked is determined by the currently selected options, all of which are on by default. The diagnostic results are posted in the **STL Diagnosis** dialog (Figure 3.13.22.3). In addition to counts of entities that were found in violation of certain STL requirements, this dialog posts error messages in a boxed area, when irregularities are detected in the data.

- **Minimum Coordinate Values:** When this option is selected, all coordinates are checked against the value in the **Minimum Coordinate Values** field. If values less than that are found, the smallest is posted in this numeric field.

- **Minimum Triangle Size:** When this option is selected, the lengths of the sides of all the triangles that resulted from the triangulation of the object are calculated. If sides smaller than 1/1000" are found, an error is issued and the smallest side length is posted in the numeric field. These triangle size restrictions are dictated by the current state of technology of the rapid prototyping machines.

- **Number Of Intersecting Triangles:** When this option is selected, all triangles that resulted from the triangulation are checked for intersections. If intersections are found, an error message is issued and how many triangles were found to intersect is posted in the numeric field.

**Preview...:** Clicking on this button invokes a preview dialog where a wire frame display of the model is shown. If any of the **Scale**, **Use Current Reference Plane As Base Plane**, or **Position In Positive XYZ Octant** options are selected, the model is shown with the respective transformations applied. The overall dimensions and the minimum and maximum coordinates of the model are also drawn. The exported objects are drawn in black while the support structures are shown in the highlight color. The preview dialog works as other preview dialogs in **form•Z** and includes tools for manipulating the view parameters of the display.
Requirements and recommendations

Before exporting to STL files, you should carefully review the instructions of the manufacturer of the rapid prototyping machine you will be using and follow their recommendations. As an example, following are the requirements and recommendations for STL files intended to be used with a Stereo Lithography Apparatus (SLA), manufactured by 3D Systems, Inc., of Valencia, California. The SLA machines use a liquid polymer that is cured by the exposure to a laser beam.

About the model:

Note that, because the STL format always exports the models triangulated, the only triangulation option that is available for STL is the Triangulate Method. All the other options in the dialog are ignored by STL.

• The minimum recommended wall thickness is 0.02 inches (0.5 millimeters).
• The model must reside entirely in the positive XYZ octant.
• The distance between the model and the origin should be minimized.
• The height of the model should be minimized, by selecting the most efficient slicing direction. This reduces the number of layers required to be built, which reduces the construction time.
• The number of enclosed volumes (cavities) should be minimized.
• The number of sloped surfaces should be minimized. Such surfaces tend to result in a stair-step appearance, with the thickness of each layer being the height of the step. Special effort should be made to orient the most important surfaces vertically or horizontally.
• The part must fit in the tank holding the liquid polymer. Large models can be built in parts through several runs and assembled after they are constructed.

About the support structure:

• The supports should be located in a manner that provides a rigid foundation for the model. This includes supporting corners and edges. The model should be separated from the machine platform by support structures.
• Bottom and side supports should overlap the model by 2 to 3 layers (0.04 to 0.06 inches). This ensures a strong bond between the supports and the model.
• Diagonal supports beginning and ending on the part should be designed as buttresses rather than extending into a corner of the model, where they are hard to remove.
• Supports typically are spaced 0.1 to 0.8 inches apart. In general, they must be located at sufficiently close intervals so that no appreciable sag or curl occurs.
• The supports must be at least 0.65 inches long where they contact the platform to span platform drain holes.
3.13.23 VRML

The Virtual Reality Modeling Language (VRML) has become a standard for exchanging 3D information and browsing virtual worlds on the Internet. There are currently two major VRML releases: 1.0 and 2.0 (or VRML 97). Most of the products now available (especially for the MacOS platform), including plug-ins for popular Web browsers, support only VRML 2.0. form•Z supports both versions as described.

VRML is an extended and expandable standard that contains prototyping mechanisms to support user defined structures. The VRML standard relies on other standards, such as JPEG, MPEG, PNG, MIDI, UTF8, or other proprietary formats, like GIF and WAV, and on other programming languages, such as Java or JavaScript. For accessing data on the Internet, it also borrows from syntactic constructs such as RURLs, URLs, and URNs. Generally, form•Z supports the necessary data nodes from both versions of the VRML standard, which can describe static scenes. This includes geometric nodes, materials, cameras, lights, and grouping.

At export, form•Z writes objects as “IndexedFaceMesh” nodes, if the Export Method for Controlled Objects is set to As Plain Objects. If it is set to As Parametric Data, form•Z additionally exports “box,” “cone,” “cylinder,” and “sphere” nodes. In addition to the geometric information, views, lights, and material related nodes are exported. Texture maps are supported only in the RenderZone Plus version, provided that the user converts the images used in the form•Z project to JPEG images.

When importing VRML 1.0 files, form•Z extracts information from the following nodes:

**Shape Nodes:** Cone, Cube, Cylinder, IndexedFaceSet, IndexedLineSet, and Sphere.
**Property Nodes:** Coordinate3, Material, Texture2 (image conversions should be manual if images are GIFs), MatrixTransform, Rotation, Scale, Transform, and Translation.
**Group Nodes:** LOD (only the first child node is used), Separator, TransformSeparator, Switch (lights are not affected when they are inside a switch node), and WWWAnchor (partially, only for its grouping properties).
**Miscellaneous Nodes:** DirectionalLight, OrthographicCamera, PerspectiveCamera, PointLight, PointSet, SpotLight, and TextureCoordinate2.

Additionally, user field definitions are used to skip over custom data fields inside standard nodes. Undefined nodes are assumed to be user nodes, and they can contain only user defined fields.

When importing VRML 2.0 files, form•Z extracts information from the following nodes:

**Grouping Nodes:** Transform.
**Common Nodes:** DirectionalLight, PointLight, Shape, SpotLight.
**Geometry Nodes:** Box, Cone, Cylinder, IndexedFaceSet, IndexedLineSet, Sphere.
**Geometric Properties:** Coordinate, TextureCoordinate.
**Appearance Nodes:** Appearance, ImageTexture, Material.
**Bindable Nodes:** Viewpoint.
Importing VRML files

A VRML file is imported into form•Z in the standard manner. The **Modeling Import Options: VRML** dialog contains only common options, which are discussed in subsection 3.13.1.

Exporting VRML files

The content of a form•Z modeling project is exported as a VRML file in the standard manner. The export options are selected from the **Modeling Export Options: VRML** dialog (Figure 3.13.23.1), the common export options of which are discussed in section 3.13.2. There is also one format specific option.

**Version**: This pop up menu determines the VRML version to which the project will be exported.

![Figure 3.13.23.1: The Modeling Export Options: VRML dialog.](image)

How textures are handled

The VRML file format supports plain color, image maps, ambient, defused, and specular reflections, roughness, specular color, and transmission.

When importing a VRML file and the file contains external references to texture maps, all the image types recognized by form•Z are considered valid texture maps. If a VRML file references GIF files (JPEG and PNG are supported), they have to be manually converted to image types recognized by form•Z, using an image conversion utility, and to associate each one of them with the form•Z surface style that is tagged with the original image name. Internally embedded texture maps in the VRML files are not currently supported.

The form•Z texture map controls of solids and surfaces are converted to U,V coordinates at export and are imported as U,V coordinates from the VRML files. In addition, cubic texture map controls are associated with boxes, spherical controls with ellipsoids, and cylindrical controls with cones and cylinders.
3.13.24 ZPR

ZPR is a file format that is used by 3D printers from Z Corporation (www.zcorp.com). 3D prints are made by spreading a thin layer of powder on the print bed. Then, using inkjet printer technology, a binder is “printed” on the powder which solidifies the powder. This printing process is done one cross-section at a time until the full 3D model is printed. To print a form•Z model, the model is exported as a ZPR file. The exported file is then loaded into Z Corporation’s Zprint software which drives the printer.

This process imposes many of the same restrictions as any other rapid prototyping process. The model must be described by one or more closed volumes. Independent surfaces are not allowed. Overlapping or intersecting volumes are allowed but in some cases may result in anomalies in the printed model. To avoid any anomalies, volumes should intersect at facetted face edges. ZPR files only contain triangular faces; consequently, form•Z decomposes all the faces of an object before the object is exported. The maximum size of the exported model is limited by the specific printer model to be used. Unlike STL, supporting structures are not needed in ZPR files. Z Corporation’s Zprint software will add these.

Importing ZPR Files

A ZPR file is imported into form•Z in the standard manner. The **Modeling Import Options: ZPR** dialog, shown in Figure 3.13.24.1, contains two format specific options, in addition to the common options discussed in subsection 3.13.1 of the form•Z User’s Manual.

**Ignore File’s Scale Data:** ZPR files contain a transform which scales (and can also rotate and translate) all geometry in the file. This allows original model vertex coordinates to be preserved while transforming the model to fit within the physical limits of the printer. Normally when a ZPR file is imported into form•Z, the geometry is transformed as specified by the file. When this option is selected, the geometry will not be transformed and the original vertex coordinates will be imported.

**Scale Factor: X, Y, Z:** The scale factors entered in these fields are applied to the geometry in addition to the file’s transform (unless **Ignore File’s Scale Data** is selected). If no scaling is needed, these values should be kept to 1.0, which are the default values.
Exporting ZPR Files

The content of a form•Z project is exported as a ZPR file in the standard manner. When exporting to ZPR, the active window must be a modeling window, as drafting elements cannot be exported to ZPR. The desired options are set in the Modeling Export Options: ZPR dialog (Figure 3.13.24.2), which contains a number of format specific options.

**ZPR Version:** The version of the format can be selected from this pop up menu. ZPR v.3 and later contain a specification for the type of units used in the file (Metric or English). Version 2 files do not have this information, thus when version 2 files are opened, the ZPR File Units dialog is invoked for telling form•Z what units the ZPR data is in. Select Inches, Feet, Centimeters, or Meters and click OK.

**Printer Model:** A printer model can be selected from this pop up menu. This information is used to determine the size of the printer bed and may cause the model to be sectioned into several zpr files. Known printer models are listed in the menu, as shown in Figure 3.13.24.3.

**Embed Textures In File:** When this option is selected, the model’s textures are embedded (contained) in the exported ZPR file. This can result in larger ZPR files, however, it has the advantage that all of the information necessary to print the model is contained in a single file. When this option is not selected, the textures are kept in separate texture files. The location and format of these is controlled by the options in the Textures section above. The Embed Textures In File option is enabled and is on by default when ZPR version 3 (or later) is selected. This option cannot be disabled when exporting ZPR version 2 files as this version only supports embedded textures.

**Scale:** Models printed on Z Corporation printers are limited to a certain size, which may make it necessary to scale the model when it is exported. This can be done by selecting the Scale option and entering the appropriate scaling parameters.

**Scale Geometry:** ZPR files contain a transform which scales (and can also rotate and translate) all geometry in the file. This allows original model vertex coordinates to be preserved while transforming the model to fit within the physical limits of the printer. When this option is selected, this transform is set to unity and the geometry of the model is scaled.
**Write Scale Data To File**: When this option is selected, the scale data defined here is written to the ZPR transform and original vertex coordinates are exported.

**Scale Factor**: When this option is selected, the scale factor contained in its field will be uniformly applied in all directions to the model.

**Fit Box**: When this option is selected, the model is scaled to fit in a bounding box whose size is determined by the numeric values contained in the X, Y, and Z fields.

**Scale To Printer**: When this option is selected, the model is scaled to fit within the printer bed as specified by the printer chosen in the **Printer Model** menu. When this option is selected, the model will not be split up and will be exported as a single ZPR file.

**Use Current Reference Plane As Base Plane**: When this option is selected, the model is rotated so that the current reference plane is aligned with the print bed. Layers of the printed cross-sections will be parallel to this plane.

**Position In Positive XYZ Octant**: When this option is selected the model is translated such that all its coordinates take on positive values and no coordinate value is less than that specified in the **Min. Coord. Value** field.

**Export Solids Only**: When this option is selected (the default), only solid objects are exported.

The **Diagnosis** and **Preview** options are identical to STL. Refer to subsection 3.13.22 of this volume of the form•Z User’s Manual for a description of these options.
3.14 Importing and exporting image file formats

In addition to the object formats, form•Z includes the ability to import and export eleven image formats, namely, BMP, EPS, Illustrator, JPEG, Metafile, PICT, PNG, QuickTime Image, Targa, TIFF, and WMF. form•Z can also export to HPGL and Piranesi. Additional image formats may be available through QuickTime. The dialogs invoked when image files are imported or exported have been structured as uniformly as possible so that switching from one format to an other should not require extensive adjustments.

Like the object formats, the image file formats also share a number of identical import and export options and therefore their dialogs share a common format. The common options are discussed in the following two subsections. The remaining subsections discuss the options that are unique to each format. Although the dialog formats are similar, there are instances where certain options are unavailable for a format type. In these cases, those options are grayed out (dimmed) and are unavailable.
3.14.1 Common image import options

The dialogs invoked when opening or importing image files into *form-Z* have been structured consistently, as they share many common options. These common options are in the upper section of the dialog (Figure 3.14.1.1), while the lower section contains options specific to that format. When an option is not available for a format, it is dimmed.

There are three distinct variations of the common import options. When **Import Model** is selected from the **Open** dialog (invoked from the **Import** item in the **File** menu), the **Import As Modeling** section is displayed as shown in Figure 3.14.1.1(a). When **Import Draft** is selected, the **Import As Drafting Elements** and/or **Import As Drafting Image** sections is displayed as shown in Figure 3.14.1.1(b) and (c). A few formats (such as PICT) support both vector and pixel information. For these formats, the **Import As Drafting Elements** and the **Import As Drafting Image** sections are shown with radio buttons for selection of the desired import method.

**Import As Modeling**: This group of items controls how vector based images are imported into the modeling environment.

**Scale Factor**: This group of items determines the X and Y scale that is applied to the vector image data when it is imported. The default is 1.0.

**Color**: This menu determines how colors are imported into the modeling environment. This is the same as for object formats imported into modeling as described in subsection 3.13.1 and shown in Figure 3.13.1.1.

**Display Resolution**: This invokes the **Display Resolution** dialog where the display parameters for imported smooth objects are set. See subsection 4.1.1 for the details on the display resolution parameters.
Import As Drafting Elements: This group of items controls how vector based images are imported into the drafting environment. The Scale Factor, Plot Scale, and Color options are as for the object formats imported into drafting, as described in subsection 3.13.1 and shown in Figure 3.13.1.9.

Import As Drafting Image: This group of items controls how pixel based images are imported into the drafting environment as draft image elements.

Image Position: These options decide the position of the image when it is imported into the drafting environment.

X, Y: The coordinates entered into these fields determine where in the drafting window the image will be placed. The default is 0'-0" which places the image at the origin of the world axis.

Image Size: These options decide the size of the image when it is imported into drafting.

Width, Height: The coordinates entered in these fields determine the size of the image. The default size is 1'-0" x 1'-0".

Keep Original Size: If this option is selected, the X,Y, Width, and Height options are dimmed and unavailable, and the image is imported at its original size.

Keep Proportional: If this option is selected, the image will retain the proportions it had before importing.

Store Image In Project: When this option is selected, images added to a drafting project as image elements are stored with their full resolution directly in the project. If this option is off, these images are not stored in the project, but the project is linked to a file that contains the image. In either case a low resolution image is stored in the project. If this option is on and the image requires more that 256 K of space, a warning is issued offering the option to either store the image or link to it. Which option is preferable depends on how much memory and disk space you have available. Images stored with a project redraw and print faster.
3.14.2 Common image export options

The dialogs invoked for exporting image files also share common options, which are at their upper portion (Figure 3.14.2.1). The lower section contains format specific options. When an option does not apply to a certain format, it appears dimmed.

**Platform**: File formats that are exclusively ASCII always have the Platform pop up menu available. Selection of one of these options produces text files with line terminations that are acceptable by Apple’s Macintosh, DOS, or Windows, and UNIX computers respectively. Binary file formats that can be exported optimized for different platforms (i.e. TIFF) have this pop up menu available too.

**Image Options**: This group contains options that affect the export of images.

- **Include Background**: If the current scene is exported as vector data, selecting this check box includes the background in the file. The background is a filled rectangle placed behind the entities shown on the screen. When this option is off, vector lines are exported without a background. In other words, deselecting this option exports the scene with a transparent background, which is desirable when a file is intended to be included on top of some other image. Note that pixel images always include the background when exported.

- **Include Preview**: When this is on, a pixel based representation of the saved scene is included in the file. This image may be displayed as a preview of the content of a file, when the file is included by another application. This option is on by default for the formats that support it.

- **Export As Image Map**: If this option is selected, the file is exported as a bitmap file.

- **Export As Vector/Polygons**: If this option is selected, the file is exported as vector lines.

- **Drafting Text As Paths**: When this option is on, the outlines of all drafting text elements that use TrueType™ or PostScript™ fonts are exported as vector line paths. This is useful when a particular plotter does not support TrueType™ or PostScript™ fonts, but the text needs to be plotted accurately. While it needs more disk space, it produces text as it appears on the screen, regardless of whether the font used is available on a printer. When this option is off, or a bitmap font is used, it is always exported in a parametric format. This option is not available in modeling.

- **Line Weight**: This pop up menu (Figure 3.14.2.2) is active only when exporting from the modeling environment. If the current scene is exported as vector data, all lines are assigned the line weight indicated in the pop up menu. The Hair Line option generates lines at the smallest resolution which a format can support. All other line weight options generate lines at the indicated thickness in points. One point is equal to 1/72 of an inch. The default is set to 0.5 point.

![Figure 3.14.2.1: The common image file format export options.](image1)

![Figure 3.14.2.2: The Line Weight menu.](image2)
3.14.3 BMP

BMP is the standard bitmap image format for MS-DOS and Windows applications. It is an exclusively pixel based format similar to TIFF and Targa. It is available on the Windows and Macintosh versions of form•Z and form•Z RenderZone Plus.

Importing BMP files

A BMP file can only be imported into the drafting environment of form•Z in the standard manner, which is through the Import... command of the File menu.

In addition to importing BMP images as drafting elements, they can also be imported as form•Z underlays and as image maps by form•Z RenderZone Plus.

Exporting BMP files

The form•Z image displayed in the current window is exported as a BMP file in the standard manner, which is by using the Export Image command of the File menu. BMP files will always export the image displayed on the window, which can be a modeling or a drafting window.
3.14.4 EPS

An EPS (encapsulated PostScript™) file contains PostScript language commands which describe the appearance of a single page. The commands in an EPS file generate 2D graphic information in the form of text, vector graphics and pixel images. Vector shapes may be lines, arcs or bezier curves. Connected paths can be constructed using those basic shapes. Paths can be drawn with different line styles and line thicknesses. Closed paths may be filled with a solid color or predefined patterns. Typically, the purpose of an EPS file is to be included, or “encapsulated,” in another document. That is, when an EPS file is included in a document, the content of the EPS file is not interpreted and translated to the graphic format of the including document, but a direct copy of the commands in the EPS file is included. When the document is printed on a PostScript compatible output device, such as a PostScript Laser Printer, the encapsulated PostScript commands are executed by the device and they will generate the graphic information as it is described in the initial EPS file. EPS files may contain a preview image. On the Macintosh, this preview image is stored as PICT; on Windows it is stored as a TIFF image. A document including the EPS file is able to show the content of the EPS file by displaying the preview image. However, the final content of the included EPS file usually cannot be revealed until the document is printed.

Importing EPS files

An EPS file can only be imported into the drafting environment of form•Z as an image element. Importing an EPS file format requires no options to be set and invokes no dialog.

Exporting EPS files

The content of a form•Z window is exported as an EPS file by executing the Export Image command (File menu), which invokes the Save dialog and then the Export Options: EPS dialog, shown in Figure 3.14.4.1. This dialog contains a few format specific options, in addition to the common options.

![Figure 3.14.4.1:](image)

The Export Options: EPS dialog.
Saving as an EPS file exports the content of the active window. When a modeling window is active, the 2D image displayed on the screen is exported, rather than the 3D representation of the modeling objects. How the exported image is represented in the EPS file (vector or pixel data) depends on the type of display that is on the screen. Wire frame and hidden line images are exported as vector lines. Renderings produce with Shaded Render and RenderZone are exported as pixel images. Quick paint and surface renderings may be exported as vector or pixel data.

When a modeling scene is exported as vector lines, the line thickness is determined by the active Line Weight option in the pull down menu of the Export Options: EPS dialog. All lines are exported as solid lines. When a drafting window is active, vector line data is exported for the drafting elements currently displayed on the screen. The line weights, line styles and hatch patterns are exported to match the attributes of the respective drafting elements.

When a modeling scene is exported as pixel data, the maximum number of colors in the pixel image is determined by the Image Color Depth selected in the Image Options dialog, invoked from the Display menu. For example, if 8 bit (256) is selected in the Image Options dialog, the pixel image can contain up to 256 different color values. When a scene is exported as vector data, the color information of an entity exported to an EPS file is determined by the red, green and blue values of the color attribute of the respective object, face or drafting element, regardless of the current Image Color Depth setting.

The drafting image elements and the modeling underlays that are produced from image formats cannot be described by EPS commands. Therefore, these entities are not included in the graphic information of an EPS file.

Write Black As: One of the three available items can be selected from this pull down menu:

- **RGB**: When this item is selected, black is exported as red=0, green=0, and blue=0.

- **Gray**: When this item is selected, black is exported as gray=0.

- **CMYK**: When this item is selected, black is exported as cyan=0, magenta=0, yellow=0, and black=1. This option is only available when PostScript Compatibility is set to Level 2.

Which option is selected may be significant when producing plates for color printing and black need to be generated as a spot color. In such cases, the CMYK and Gray options usually work best.

**Postscript™ Compatibility**: When the Level 1 option is selected, the EPS file can be printed on a Postscript device which only supports Postscript Level 1. Postscript Level 1 does not support color images and hatch patterns. Level 1 files are supported by Level 2 devices but Level 2 files cannot be printed on Level 1 machines. The default option is set to Level 2.
3.14.5 HPGL

An HPGL (Hewlett Packard Graphics Language v2) file contains commands which control pen plotters (mostly HP plotters, although several other plotter manufactures have also adopted this language). The commands in an HP-GL/2 file represent 2D graphics information in the form of stick text and vector graphics. Vector shapes may be lines, arcs, or Bezier curves. Connected paths can be constructed using those basic shapes. Paths can be drawn with different colors, line styles, and line widths. Paths may be filled with a solid color or a predefined pattern. Typically the purpose of an HP-GL/2 file is to be sent to a pen plotter to be plotted/printed on paper. HP laser printers which support HP’s PCL5 (Printer Control Language v5) can also print HP-GL/2 by setting their print mode to HP-GL. There are also several HPGL software viewers available.

Exporting HPGL files

The content of a form•Z window is exported as HP-GL/2 by executing the Export Image command (File menu), in the standard manner. The Export Options : HPGL dialog is shown in Figure 3.14.5.1. It contains a number of format specific options.

Saving an Hp-GL/2 file exports the content of the active window. When a modeling window is active, the 2D image which is currently displayed on the screen is exported, rather than the 3D representation of the modeling objects. Since HP-GL/2 does not support pixel images, Shaded Render*, RenderZone*, and Sketch Render* renderings can not be exported. Text objects are exported as outlines since HP-GL/2 does not support fonts.

Figure 3.14.5.1: The Export Options: HPGL dialog.
When a modeling scene is exported, the line thickness is determined by the active Line Weight option on the pull down menu of the Export Options : HPGL dialog. All lines are exported as solid lines. When a drafting window is active, vector line data is exported for the drafting elements currently displayed on the screen. The line weights, line styles, and hatch patterns are exported to match the attributes of the respective drafting elements.

**HPGL Options**: These are the options specific to the HPGL format, as follows:

- **Plot Scale**: Sets the scale of the drawing on the printed page.
- **Scale To Fit Page**: When this option is selected the drawing scale is automatically set to fit the drawing on a single sheet of paper defined by the Page Size options.
- **Page Size**: This menu contains several standard drawing sizes. The Custom item allows you to define your own page size.
- **Height, Width**: The height and width of the page.
- **Margins**: These values determine the space between the plot area and the edge of the page.
- **Right, Left, Top, Bottom**: The spaces between the right, left, top, and bottom edges of the page and the plot area.
- **Orientation**: Orientation specifies the orientation of the printed page.
  - **Portrait**: With this option on, the page is plotted as defined.
  - **Landscape**: This option swaps the width and height of the image to be plotted; in other words, it plots the image sideways.
- **Origin**: This value specifies where the origin of the drawing (plotter coordinate 0, 0) is.
  - **Lower-Left**: This option places the plotter origin in the lower-left corner of the page.
  - **Center Of Media**: This option places the plotter origin in the center of the page.
- **Add PCL Commands**: When this option is selected, PCL5 commands are added to the beginning and end of the HP-GL/2 file. The exported file can then be sent to an HP laser printer. The PCL5 commands set the printer’s mode to HPGL before HP-GL/2 commands and resets it to PCL mode after the HP-GL/2 commands.
- **Pen Selection Method**: Since plotters use physical pens to draw and HP-GL/2 files refer to these pens by number (not color), there needs to be a mapping from colors in form•Z to pen numbers. Pen tips can also have different thickness resulting in different line widths. This menu and the dialog invoked from the Pen Definitions... button (see Figure 3.14.5.2) are used to set up this mapping.
One Pen: The plotter has only one pen. All colors are mapped to this pen.

Best Match - Color First: The form-Z color is mapped to the Pen Colors dialog by comparing colors and line widths. The pen color has a higher priority than the line width; thus a pen will be selected for an object or draft element which has the best color match, even if it has a different line width.

Best Match - Width First: The form-Z color is mapped to the Pen Colors dialog by comparing colors and line widths. The pen's line width has a higher priority than the color; thus a pen will be selected for an object or draft object which has the best line width match, even if it has a different color.

Dynamic Pen: This is only useful for HP-GL/2 files that will be viewed in a software viewer or printed on a color printer. This option embeds color information on the HP-GL/2 file. Pens are assigned dynamically to these colors.

**Pen Definitions...**: This button invokes the Pen Definitions dialog, shown in Figure 3.14.5.2. This dialog is used to define the pens that are available on a plotter.

The top portion of this dialog is a scrollable list of defined pens. It shows the pen number, color, and line (pen tip) width. Clicking on a pen definition in this list will select it for editing. The bottom portion of the dialog contains fields that are used to set pen definitions.

**Number Of Pens**: This sets the number of pens the plotter has. The number of pen definitions shown in the list will match this number.

**Edit Pen**: This area contains fields for editing the definition of the selected pen.

Pen #: The numeric index of the selected pen (not editable).

Color: The color of the selected pen. Clicking on this item will pop up the Color Selection dialog and the pen color can be set in this dialog.

Line Width: The line (pen tip) width. The value in this field is always in millimeters.
3.14.6 Illustrator

Adobe Illustrator™ is a 2D drawing and page designing program. Users of the program can generate open and closed paths and add text. Paths are defined by a series of connected segments. Each segment can be a straight line or a curved Bezier segment. Paths can be filled with a solid color or a pattern. Text elements in Illustrator can be placed at a point, inside a path, or along a path. The Illustrator file format describes the geometry and attributes of all paths and text elements in a PostScript® language page description. When imported into form•Z, the PostScript commands contained in the Illustrator file are interpreted and modeling objects or draft elements are generated. Currently, form•Z imports Illustrator files versions 3 to 8, and exports Illustrator files version 3.

Illustrator paths can be composed of straight line segments and curved Bezier segments. When imported into the modeling environment, an Illustrator path is created as a composite curve. In drafting the path is converted to a segmented polyline representation.

Given the 2D character of the Illustrator file format and the 3D nature of the form•Z modeling structures, only 2D images of modeling scenes can be exported from the modeling environment and only surface objects can be imported into it. When exporting from the drafting environment, all elements visible in the drafting window, except the bitmap entities, will be exported. Since the Illustrator file format does not support image elements and bitmap fill patterns, image elements are skipped and bitmap patterns are substituted with a solid fill pattern.

As a comprehensive page design program, Illustrator offers a great variety of text attributes. Some of those attributes are not supported in form•Z, since they exceed the scope of a drafting and modeling program. When text elements are imported, those attributes are ignored for the generation of form•Z text. In addition, Illustrator text is stored both in a parametric form and in a final form, the latter of which is displayed on the screen. Since form•Z does not support several of the Illustrator text parameters, the final form representation of the text is used when importing. That is, all parts of a text element which share the same attributes are generated as separate form•Z entities. Illustrator also offers text which follows the shape of a path or fills the area of a path. When imported into form•Z, each part of the text which bends at a corner of a path or wraps to a new line will be generated as a separate text element in form•Z.

Importing Illustrator files

An Illustrator file is imported into either the modeling or drafting environment of form•Z in the standard manner. The Import Options: Illustrator dialog, shown in Figure 3.14.6.1, contains some of the common options discussed in section 3.14.1 and a few format specific options, as follows:
Maintain Grouping: This option is only available when importing into a modeling project. When selected, the elements grouped in the Illustrator file will be grouped when imported into form-Z. Otherwise no groups will be generated.

Position On Current Reference Plane: With this option, which is only available when importing into modeling, the 2D data in the Illustrator file will be placed on the current reference plane, which may even be an arbitrary plane. When this option is off, the Illustrator data is placed on the world XY plane.

Create Object Text From Text Elements: As discussed in subsection 3.13.1.

Make Solid Hatch Elements From Filled Paths: This option is only available when importing into drafting. When it is on, each path that is filled generates a polyline, representing the path outline, and a solid hatch element, representing the path fill. When this option is not selected, only the polyline is generated.

Make Smooth Objects: When this option is on, which is the default, curves will be imported as smooth objects. Selecting this option for large files can greatly increase the import time. When this option is off, all entities will be imported as faceted.

Exporting Illustrator files

The content of a form-Z window is exported as an Illustrator file in the standard manner. The Export Options: Illustrator dialog only contains common options, which are as discussed in section 3.14.2.

Saving an Illustrator file exports the content of the active window. When a modeling window is active, the 2D image which is currently displayed on the screen is exported, rather than the 3D representation of the modeling objects. However, since Adobe Illustrator does not support pixel images, pixel based renderings cannot be exported. If a modeling scene contains plain text objects, the outline of the plain text is not exported to the Illustrator file, since the outline data is not stored with plain text objects.

Figure 3.14.6.1: The Import Options: Illustrator dialog for (a) modeling and (b) drafting.
3.14.7 JPEG

The Joint Photographic Experts Group (JPEG) format is commonly used to display photographs and other continuous-tone images over the Internet. JPEG files retain all their color information in an RGB image and also use a compression scheme that effectively reduces file size by identifying and discarding nonessential image display data.

Because it discards data, the JPEG compression scheme is referred to as “lossy.” Once an image has been compressed and then decompressed, it will not be identical to the original image. A high level of compression will result in lower image quality, while low compression levels result in higher quality images.

Importing JPEG files

A JPEG file is imported into form•Z in the standard manner. The Import Options: JPEG dialog, contains common import options only (see subsection 3.14.1). There are no format specific importing options.

Exporting JPEG files

The content of a form•Z window is exported as a JPEG file in the standard manner. The Export Options: JPEG dialog (Figure 3.14.7.1), in addition to the common options that are as discussed in section 3.14.2, contains one format specific option.

**Quality Factor**: The items in this pop up menu specify which JPEG compression scheme will be used when exporting. Selecting the maximum quality setting, **5 (High)**, will produce the best image, but will result in a larger file size. Selecting the lowest setting, **1 (Medium)**, will produce the lower quality image, but will result in a smaller file size. The default setting is **2**.

*Figure 3.14.7.1: The Export Options: JPEG dialog.*
3.14.8 Metafile

The Metafile format is a Windows standard, where it is used for exchanging graphic information between applications. A Metafile is a collection of 2D integer based Windows drawing commands. A Metafile image may contain vector graphics, such as lines, rectangles, and polygons, as well as bitmap or pixelmap images. The Metafile format is analogous to the PICT file format for the Macintosh computers and all the options available in its import and export dialogs work as for PICT files. The Metafile format is only available on Windows.

**Importing Metafile files**

The content of a Metafile is imported into form•Z in the standard manner. The *Import Options: Windows Metafile* dialog (Figure 3.14.8.1) contains common options only, which are as discussed in section 3.14.1. Note that Metafile images can also be imported as underlays and by form•Z RenderZone Plus to be used as image maps.

**Exporting Metafile files**

The content of a form•Z window is exported as a Metafile file in the standard manner. The *Export Options: Windows Metafile* dialog, shown in Figure 3.14.8.2, contains common options only. They are as discussed in subsection 3.14.2. Saving as Metafile exports the content of the active window, which may be a modeling or a drafting window. When a modeling window is active, the 2D images currently displayed on the screen are exported, rather than representations of the modeling objects.

How the exported images are represented in the Metafile format (vector lines or bit/pixel maps) depends on the type of display that is currently on the screen. Shaded render, and RenderZone renderings are exported as bit maps. Wire frame, hidden line, quick paint, and surface renderings may be exported as vector or bit map representations.
3.14.9 PICT

PICT is Apple’s standard format through which Macintosh applications can exchange graphic information. The PICT file is essentially a collection of QuickDraw drawing commands and it contains integer based 2D information. PICT may consist of both vector line representations and bit/pixel maps. The vector shapes may be simple lines, arcs, circles/ovals, rectangles, round-corner rectangles, or polygons. The bit/pixel maps may be text or any picture represented as a collection of dots (pixels). The term “bit” is used for black and white images, while “pixel” is used for color images.

Macintosh versions of form•Z can import and export PICT format files, and also Windows versions with QuickTime 3D installed, where PICT is available only as a QuickTime pixel image. Given the 2D character of the PICT format and the 3D nature of the form•Z modeling structures, only 2D images of modeling scenes can be exported from the modeling environment, and only surface objects can be imported into it. The exchange of data between PICT files and the drafting environment of form•Z is more straightforward, since both are of a 2D character.

Importing PICT files

The content of a PICT file is imported into form•Z in the standard manner and can be imported either into modeling or drafting. The Import Options: PICT dialog, shown in Figure 3.14.9.1 contains common options only, which are as discussed in section 3.14.1. There are two variation of the dialog, one for modeling and one for drafting.

PICT images can be used as underlays and as image maps in form•Z RenderZone Plus.

Figure 3.14.9.1: The Import Options: PICT dialog for (a) modeling and (b) drafting.
Exporting PICT files

The content of a form-Z window is exported as a PICT file in the standard manner. Saving as a PICT file exports the content of the active window, which may be a modeling or a drafting window. When a modeling window is active, the 2D images which are currently displayed on the screen are exported, rather than 3D representations of the modeling objects. How the exported images are represented in the PICT file (vector lines or bit/pixel maps) depends on the type of display that is currently on the screen. Wire frame and hidden line images are exported as vector lines. Shaded render, and RenderZone renderings are exported as bit maps. Quick paint and surface renderings may be exported as vector or bit map representations.

The Export Options: PICT dialog, shown in Figure 3.14.9.2, contains common options (see subsection 3.14.2) and a group of format specific options, as follows:

Compress: When this option is selected, the exported image is compressed using the QuickTime compression engine. If QuickTime is not installed on your machine or has been disabled, this item is dimmed and unavailable. PICT images exported with QuickTime compression can only be used on machines that have QuickTime installed. When this option is off, which is the default, the PICT image is exported in the normal uncompressed fashion.

Options...: Clicking on this button invokes the Compression Settings dialog (Figure 3.14.9.3), which allows selection of the compression algorithm and settings to be applied to the exported image. Note that this is a standard dialog provided by QuickTime and is discussed in subsection 3.6.10.
3.14.10 Piranesi

The Piranesi File Format (EPX) is a pixel based format used by Piranesi, a stylistic render and painting program that can be used to apply artistic treatment to computer renderings. form•Z exports renderings to EPX files for use in Piranesi. For more information on Piranesi, please see the informatix web site at www.informatix.co.uk.

Exporting Piranesi files

The image contents of a form•Z window are exported into a Piranesi file in the standard manner. The Export Options: Piranesi dialog contains common options only. They are as discussed in subsection 3.14.2.

The size and resolution of the EPX file is determined by the settings in the Image Options dialog (see subsection 3.6.12). Piranesi export works with all rendering modes, however, it is most useful when used with shaded renderings and RenderZone renderings.
3.14.11 PNG

The Portable Network Graphics (.png) image format is a generic cross platform format designed to be compact and transmit quickly across networks and the Internet. It is frequently used in association with VRML and is one of the formats that VRML uses for representing texture map images. It is fully supported in form•Z as all other pixel based image formats (TIFF, TARGA, BMP). PNG images can be used as texture maps and can be viewed using the View File menu item in the File menu.

Importing PNG files

A PNG file can be imported into form•Z in the usual manner. The Import Options: PNG dialog contains common options only, which are as discussed in subsection 3.14.1.

Exporting PNG files

The graphics in a form•Z window can be exported in PNG format in the standard manner. The Export Options: PNG dialog contains common options only, which are as discussed in subsection 3.14.2.
3.14.12 QuickTime™ Image

QuickTime supports a single frame QuickTime Image format as well as additional formats through component plug-ins. form-Z can import and export the QuickTime Image format as well as any format that is supported through the available Apple or third party QuickTime plug-ins. This format is not available when QuickTime is not installed.

The minimum version of QuickTime supported is QuickTime 2.5 for Macintosh and 3.0 for Windows. It is recommended to use the latest version of QuickTime available. Please visit www.QuickTime.com for details on the latest versions.

Importing QuickTime Image files

The content of a QuickTime Image is imported into form-Z in the standard manner. The Import Options: QuickTime™ Image dialog only contains common options, which are as discussed in subsection 3.14.1.

Exporting QuickTime Image files

The content of a form-Z window is exported as a QuickTime Image file in the standard manner. The Export Options: QuickTime™ Images dialog (Figure 3.14.12.1) contains both common options (see subsection 3.14.2) and a group of format specific options.

Compress: When this option is selected, the image will be compressed using the standard QuickTime compression options which are accessed from the adjacent Options... button, as discussed in subsection 3.14.12.

QuickTime plugin components may add formats to the Export Image submenu of the File menu. These formats can be identified by the string “(via QuickTime)” written next to the format name. If one of these items is selected at export time, the data to be exported is sent to QuickTime which produces the export file.
3.14.13 Targa

The Targa (TGA) file format is commonly supported by MS-DOS and Windows color applications. It is a pixel based format similar to TIFF and pixel based PICT files. It is available on both Windows and Macintosh versions of form•Z.

Importing Targa files

A Targa file can only be imported into the drafting environment of form•Z, which is done in the standard manner. The Import Options: TARGA dialog contains common options only, which are as discussed in subsection 3.14.1. There are no format specific options to be set when importing Targa files.

In addition to importing Targa images as drafting elements, they can also be imported as underlays and as image maps by form•Z RenderZone Plus.

Exporting Targa files

The form•Z image displayed in the current window is exported as a Targa file in the standard manner. The Export Options: TARGA dialog (Figure 3.14.13.1), in addition to the common export options (see subsection 3.14.2), contains a group of format specific options.

Compress: Selecting this item causes the Targa file to be run length encoded, which results in a smaller size without losing any image information. Targa files will always export the image displayed on the window (which can be a modeling or a drafting window) and they are always written as 24 bit color pixel images.
3.14.14 TIFF

The Tagged-Image File Format (TIFF) is a pixel based format, used to exchange image documents between different applications and different computer platforms. Typically, in applications that support TIFF, the scene currently displayed on the screen is saved into a TIFF file. Each screen pixel is saved with the color information as shown on the screen. TIFF files can be quite large. To keep their size reasonable, there is an option for compressing them.

Importing TIFF files

A TIFF file can only be imported into the drafting environment of form•Z as an image element. This is done by executing the Import... command from the File menu in the standard manner. The Import Options: TIFF dialog contains common options only, which are as discussed in section 3.14.1.

In addition to importing TIFF images as drafting elements, they can also be imported as form•Z underlays and as image maps by form•Z RenderZone Plus.

Exporting TIFF files

The contents of a form•Z window are exported into a TIFF file in the standard manner. The Export Options: TIFF dialog (Figure 3.14.14.1) contains both common options, which are as discussed in subsection 3.14.1, and one format specific option.

Compress: When this option is selected, the TIFF file is compressed to the smallest size possible without losing any image information.

Figure 3.14.14.1:
The Export Options: TIFF dialog.
Modeling

4.0 Introduction

Modeling and composing objects in 3D space is the main thrust of form•Z. There are two types of models: smooth and facetted or polygonal. These exist in parallel to two object types, which are the plain and parametric objects. There are many different types of parametric objects, such as analytic primitives, revolved objects, nurbs, patches, splines, and text objects, to mention some. The difference between plain and parametric objects is that the latter are stored with their parameters and they can subsequently be edited and changed, typically through a preview or edit dialog.

All form•Z objects exist in the same 3D environment. Which type of object is generated depends on the tool used. Objects can be generated directly or can be derived from other objects. A variety of operations can be applied to objects to transform either their geometry, their topology, or both. In general, especially given the ability to combine facetted with smooth objects, it is not an exaggeration to suggest that there is no form, real or imaginary, that cannot be generated in form•Z, using some combination of its tools.

Which type of model (facetted or smooth) is generated depends on the tool used or on the selection of an option. For example, all primitive objects (cylinder, cone, sphere, torus) except the cube are always generated smooth. For all directly drawn objects (extrusions, convergences, enclosures) there is an option to be selected and may be either facetted or smooth. The same objects, generated as derivatives, follow the type of the source shape. Derivatives, such as objects of revolution, helixes, and sweeps offer options which allow you to generate them either as facetted or smooth.

It is also possible to apply operations to mixed types of operands. For example, you can Boolean a facetted object with a smooth. The model type of the resulting object depends on the selection of an option, accessible through the Preferences dialog (see section 3.2.7).
4.0.1 Types and representations of objects

There are two types of object models in form-Z: facetted (also known as polygonal) and smooth. Internally, they are both stored through a boundary representation (b-rep). However, the b-rep used for each type of model is to some extent different. It is important to have some understanding of how these representations work, since many attributes that are associated with objects depend on them.

In solid modeling, a faceted object such as the cube in Figure 4.0.1.1, is essentially a collection of points or vertices in 3D space, called world space, and information about how these points are connected. The points are defined through their coordinates (x,y,z), called world coordinates. They are the part of an object known as its geometry. The connections of points are known as the topology of an object, which is what associates form to an object. That is, the same set of points can be connected in different ways, resulting in a variety of forms. The topology also associates attributes to an object. The main attribute we are concerned with in solid geometry is solidity.

The boundary representation consists of points, segments, outlines, faces, and volumes.

• A point (or vertex) is simply a coordinate triplet (x,y,z).
• A segment (or edge) is a directed connection between two points.
• An outline is an ordered sequence of segments.
• A face is a collection of outlines.
• A volume is a closed collection of faces.
• An object is a collection of volumes.
• Objects can be linked to form a group.

These elements constitute the topological levels of an object’s structure. They were given in an order from the lowest level to the highest and are shown in Figure 4.0.1.2.

From a practical point of view, the surface of a solid volume is subdivided into a number of portions, the boundaries of which are delineated by closed and ordered sets of segments. Such a bounded surface constitutes a face, which is delineated by one or more outlines. In its simplest form, a face is delineated by a single outline, such as in the case of the cube, where all six faces are delineated by single outlines. But a face may also contain one or more holes, in which case more than one outline is required to accurately delineate its boundary. An example of an object with faces that consist of more than one outline is the 3D enclosure shown in Figure 4.0.1.3.
The distinct difference between facetted and smooth objects is that the faces of the former need to be planar, while those of the latter are curved surfaces, which may include flat surfaces as a subset. An implication of this is that other topological levels, such as segments, are straight lines in facetted objects, but typically curved lines in smooth objects. There are also implications relative to the number of faces required to represent objects which are inherently of a curved shape. For example, it takes many faces to approximate and represent the surface of a facetted sphere, but it takes a single face to represent a smooth sphere. Interestingly, since the single face of a smooth sphere is a continuous surface, it has no points and no segments. By comparison, a smooth cylinder has three faces, two edges, and two vertices.

For both facetted and smooth objects, the direction in which the segments that delineate a face are connected is significant. By convention, the clockwise direction is defined to be positive and the counterclockwise to be negative. When a face is viewed from its front, its outer boundary should be positive, while its inner outlines (holes) should be negative (Figures 4.0.1.4 and 4.0.1.5). These directions are exactly the opposite, when a face is viewed from its back. A face can only have a single outer outline, but it may contain any number of inner outlines. All outlines of a face should be disjoined, meaning that one outline cannot intersect another outline of the face.

By the boundary representation, each pair of points is connected twice, in opposite directions. In other words, for an object to be solid, each segment in its topological structure should have a reversely coincident segment and should only have one such segment (Figure 4.0.1.6). This is otherwise known as the condition of closure, which guarantees that a solid is completely enclosed by a continuous surface, or otherwise it does not qualify as a solid. When the condition of closure is satisfied, an object is called well formed. It is called incompletely formed otherwise. The solids of form•Z are always well formed.

While form•Z is above all a solids modeler, it is also a surface modeler, which makes it a hybrid system. This is true for both its facetted and smooth personalities. When viewed from the point of view of solids modeling, the surface objects are incompletely formed, yet they are quite appropriate as surfaces or open lines and their respective operations, which are also included in form•Z.
The behavior of the different types of models and objects in form•Z may vary significantly. It is thus frequently important to know the type. When in doubt, the Query tool can be used to find out the type of an object. The different categories that the Query tool will distinguish are summarized in Figure 4.0.1.7.

Figure 4.0.1.7: Types of form•Z objects.
4.0.2 Parametric objects and their representations

As already mentioned, in addition to the plain objects, form•Z includes tools for generating a variety of parametric objects. These objects are internally stored with their parameters and they can subsequently be revisited or edited and changed, without having to recreate them from the beginning. Which type of representation is preferable really depends on the modeling task at hand. form•Z is unique in that it supports a rather extensive repertoire of representational methods under a single roof. In addition, it routinely allows conversions from one representation to another. For example, smooth objects can be converted to facetted, which allows you to take advantage of a different set of modeling tools that are available for facetted geometry.

The following paragraphs summarize the parametric representations available in form•Z. However, this is by no means a complete discussion and more is covered in the respective sections. Yet, especially with the parametrics, formal definitions are not that important. Our recommendation is that you try the different parametric objects, explore their behavior, and try to identify what suits best your modeling tasks.

**Parametric splines** and **controlled curves**: These are curved lines that are generated from control points. They are available in two variations. (1) The parametric splines are drawn directly and can later be edited using the Edit Controls tool. (2) The controlled curves (or c-curves) are generated as derivative curves from previously drawn vector lines. They can be Bezier, B-splines, or NURBS of any degree. The control points and other parameters are stored with the curve and can later be edited, using the same tool that created them.

**Analytic primitives**: These are the basic forms of analytic geometry, namely the cube, cone, cylinder, sphere, and torus. There are tools that generate them directly and they are internally stored through their analytic parameters. They can easily be converted to other parametric personalities as well as facetted representations.

**Spherical primitives**: These include two types of spheres (revolved and geodesic), the soccer ball, and the set of Platonic solids (tetrahedron, hexahedron, octahedron, dodecahedron, and icosahedron). While they are ultimately facetted objects, they are parametric primitives, whose size, shape, and type can be changed after their initial generation.

**Parametric derivatives**: These are the extrusions, enclosures, objects of revolution, the surface/solid helixes, the screws/bolts, the spiral stairs, the stairs from path, and the sweeps. Some of these can be generated as either facetted or smooth objects, including nurbs. When they are facetted they are parametric with respect to the procedural parameters that generated them, which includes any source and path shapes or axes that were used.
**Controlled meshes** and **nurbz**: These are two variations of NURBS surfaces. (1) The control meshes (c-meshes) are hybrid NURBS/facetted objects generated from previously drawn control lines. By this representation, parts of an object are represented by the full range of NURBS controls (including Bezier and B-splines) and parts are planar facets. While this hybrid method offers major advantages toward the representation of virtually any 3D form, it has the disadvantage of not being exportable as a pure NURBS representation, which is what the next type offers. (2) Nurbz are purely spline based meshes. Such nurbz objects can be generated from analytic primitives, from most of the parametric derivatives, and also directly from control lines. After their generation they can be edited with both the Edit Controls and Edit Surface tools.

**Bezier** and **Coons patches**: The *patch* is a rectangular or triangular tile, whose four or three sides are control lines from which curved surfaces can be generated. This makes it similar to the NURBS surfaces, however it also has some distinct properties of its own. As the representation of a complete object consists of a set of patches, which may completely enclose an object to produce a solid, patches do not have to be all of the same size, but can be subdivided at any depth desired. Thus, they are particularly useful for modeling objects whose surface detail may vary widely from one area to another.

**Metaformz**: These are known as implicit surfaces generating entities, which have the property of blending smoothly with each other, when they intersect or their regions of influence interact. They are known as metaballs, which, in **form•Z**, can be generated directly as primitives. In addition, derivative metaformz can be generated from any facetted shape, which represents a broad generalization of the metaball algorithms. You can change their parameters, relative positions, and blending attributes, which results in different shapes.

**Controlled rounding** and **subdivisions**: These are operations that may be applied to polygonal objects in a parametric manner. The former rounds or bevels the edges and/or points of solids and the latter curves the surfaces of objects as they are meshed. In both cases the parameters that were applied can be revisited and revised.

**Text**: This is a special type of a parametric object. The content, style, and shape, which includes the shape of the line on which the text may have been placed, can be changed after the initial generation of a text string.

**Symbols**: These can be considered another special type of a parametric object, since their editing and other operations frequently resemble those of parametric objects. Symbols are collections of entities that are repeated in a project. They are defined and stored in libraries, from which they are placed as instances. The instances themselves do not contain any geometry or topology, but rather references to symbol libraries from which they can be displayed with a project.
4.0.3 Local coordinate systems and centroids of objects

All objects of form•Z, including those that may have a two dimensional shape, exist within the 3D world space, which has a global origin. Recall that origin is the (0,0,0) point of a Cartesian coordinate system from which all other coordinate points are measured. In addition to their global coordinates, all form•Z objects also have local coordinates, which are relative to a local origin and coordinate axes. They also have a centroid, which is by its nature always local.

The local coordinate axes and centroid of objects can be displayed by turning on the Show Axes and Show Centroid options respectively, in the Wire Frame Options dialog (Figure 4.0.3.1).

A local coordinate system is one relative to which an object is defined and follows the object when operations, especially geometric transformations are applied to it. The position of the origin of the local coordinate system is generated at the time the object is created, according to an option selected from the Objects dialog (invoked from the Objects... item in the Options menu), which is shown in Figure 4.0.3.2.

Per Object Type: When this option is on, which is the default, the origin of the local coordinate axes is placed where it appears to be normal for the respective form. For example, for the sphere it will be placed at its center, for the cone and cylinder it will be placed at the center of their circular base, etc.

Center Of Gravity, Average Of Points, Center Of Bounding Volume: When one of these options is selected, the origin is placed according to a centroid definition (see below). However, it may or may not coincide with the centroid, which depends on where the centroid is placed.

The initial positions of the local axes can be changed through the Query Object dialog, where their world coordinates and orientation are displayed. A significant by-product of the local coordinate system is that objects now have widths, depths, and heights represented by the local X, Y, and Z axes respectively. These follow the object as it may be rotated in space. Examples of local axes displayed are shown in Figure 4.0.3.3(a).
A centroid is a point somewhere at the center of an object (Figure 4.0.3.3(b)). It can be mathematically calculated as one of three available types, which is selected in the Objects dialog.

**Per Object Type:** When this option is on, whatever type of centroid appears natural for each type of object is used rather than using the same type throughout.

**Center Of Gravity:** With this option selected the centroid is placed at the exact center of the volume (mass) of an object, which is independent of the surface resolution of the object (Figure 4.0.3.4(a)).

**Average Of Points:** With this option the centroid is calculated as the average the coordinates of all the points of the object. If an object has a higher resolution (more points) on one side, the average centroid will be closer to that side (Figure 4.0.3.4(b)).

**Center of Bounding Volume:** With this option the centroid is positioned at midpoint between the minimum and maximum ends in each direction of the bounding box (Figure 4.0.3.4(c)).

Note that the example in Figure 4.0.3.4 was specifically selected to be able to illustrate the three variations of calculating the positions of centroids. These options have no effect on completely symmetrical objects, such as spheres. The centroid is calculated according to one of the above options at the time an object is created. Its position, expressed in world coordinates, can be inspected in the Query Object dialog.

A local bounding box is a cuboid that extends between the minimum and maximum ends of an object. The term local signifies that it has an orientation parallel to the local axes of an object and follows the object as it may be repositioned and rotated in 3D space. The local bounding boxes may be displayed in wire frame, instead of the real objects, by turning on the Show Objects As Bounding Volumes option in the Wire Frame Options dialog.
4.0.4 The modeling tool bar

By default, at start up time, the modeling tool bar is arranged vertically in two columns and occupies the left edge of the screen. Its tools can be expanded or popped out as shown in Figure 4.0.4.1.

Figure 4.0.4.1: The torn off palettes of the modeling tools.
The tool bar is by no means fixed. It can be arranged horizontally by clicking on the third button found at its upper left corner, it can be moved and positioned anywhere on the screen, its tools can be popped out and torn off as separate tool palettes, and most importantly the content and structure of these palettes can be completely customized (see section 1.9). In other words, for each individual user, the tool bar and its palettes may look completely different than the default set up with which form•Z is shipped.

Whatever the case, the discussion throughout this chapter and throughout this User’s Manual assumes the default shape, structure, and position of the tool bar. Where necessary, we refer to it as left and right tool bar column, and as nth row. The default modeling tool bar has 15 rows and 30 individual palettes. Their content in icons range from a minimum of one icon (left 15th row) to a maximum of 14 (right 3rd row).

The icons from which more icons can be pulled out are marked with a little arrow. Tool icons marked with a little dot in their upper right corner have a dialog associated with them that can be invoked by double clicking on the icon. It can also be invoked by pressing option (Macintosh) or ctrl+shift (Windows) while clicking on the icon. On Windows, the dialog can also be invoked by right-clicking on the icon. The dot is shown in the highlight color (default red).

The rows of icons that pull out and are torn off become stand alone palettes, which can be positioned anywhere on the screen. These are actually the palettes shown in Figure 4.0.4.1. Torn off palettes can be collapsed by clicking on the hide box in their upper right corners. This works the same as for the other form•Z palettes (see section 2.3).

The form•Z tool bar contains different types of tools, which are color coded and produce different types of results. These types are discussed in the next subsection, which also summarizes what the groups of tools in each default tool palette do.
4.0.5 Types of tools

The modeling tool palette contains three types of tools: *primitives*, *modifiers*, and *operators*. In the standard tool bar, the primitives are colored black-and-white and are used to draw directly and generate different types of primitive solids. The modifiers, colored turquoise or magenta, set a mode that affects the actions of the operators. The operators, shown in black-and-white, are the action producing commands. All the operators generate or manipulate objects.

*Modeling primitives*

There are three types of primitives positioned in two palettes on the first row.

**Primitives**

These tools generate five analytic primitives, namely cube, cone, cylinder, sphere, and torus, and four quadratic surfaces, namely paraboloid, single and double paraboloid, and hyperbolic paraboloid. They are all analytic objects, and thus parametric. They can be edited after their initial generation.

**Balls**

These three tools generate a range of spherical objects, which may be transformed to solid stars, and metaballs, which are parametric and can be edited and changed after their initial creation.

*Modeling modifiers*

There are four types of modifiers positioned in four palettes. The modifiers are mutually exclusive within their palette. That is, in each palette, one icon is always *active* or *selected*.

**Object Type**

These modifiers, also called Direct Object Generation modifiers, determine the type of object that will be generated after the execution of one of the drawing operators.

**Insertions**

These four modifiers determine the type of insertion that will be applied after the execution of a drawing operator.
Topological Levels

These modifiers establish the topological level to which subsequent operators will be applied. While seven of them set specific levels, the last modifier is “intelligent” and specifies a topological level on the basis of how close to an entity the mouse cursor is.

Self/Copy

These modifiers determine whether geometric transformations are applied to the object itself, to a single copy, or to multiple copies and whether the copied objects will form a family of clones. Also included in their palette is the Define Macro modifier, which is of a special character, but is mutually exclusive with the Self/Copy modifiers.

Modeling operators

The tools in the remaining palettes of the modeling tool bar are operators. They either generate new objects or they execute operations on existing objects and their parts. Contrary to the modifiers, all primitives and operators taken together are mutually exclusive. Only a single action producing tool (which is done by the primitives and operators) can be active at any given time. Selecting a new primitive or operator deselects the one previously active.

As already mentioned, to the extent possible, the default tool bar has been structured in a manner that corresponds to the inherent order in which operations are expected to be executed in form•Z. This refers to the way the tools have been grouped in the palettes, the order of the palettes, and the positions of the modifiers. With only very few exceptions, the modifiers affect tools that are at a lower palette, but they do not necessarily affect all the tools below them. Which modifiers affect which operators is discussed in the following paragraphs and throughout this manual.

Polygons and Circles

These operators generate polygonal shapes, circles, or ellipses from two or three points entered through graphic or numeric input.
Point, Lines, Splines, and Arcs

The operators on this palette allow you to draw single points (one click), single segments (two clicks), open or closed vector lines (many mouse clicks), spline curves, and arcs. By switching icons as you draw, straight lines, curves, and arcs may be combined in a single shape. The shapes you draw result in the generation of new objects or insertions, depending on the active modifier.

Picking and Editing

The first tool in this palette is used to select objects or their parts. Which part of an object is picked is determined by the active topological level. This tool can also be used to directly transform objects. The other four tools allow you to edit parametric objects.

Derivatives 1

The operators in this group are used to generate a new object from one or more existing objects. For example, a cube may be generated from a surface object or a pyramid from the face of a cube. Some of the operators generate types of objects that are the same with those generated by the direct object generation procedures. Other operators produce types of objects that can only be generated as derivative objects; for example, terrain models. In many instances, the active topological level affects the result produced by the derivative object generating procedures, by determining whether a complete object or a part of an object will be used as the source shape for the generation of the derivative object.

Derivatives 2

The group of tools in this palette contains a special set of derivative operators, namely objects of revolution, helixes, two types of screws and bolts, gears, two types of stairs, two types of sweeps, skins, lofts, and cap. They typically create new objects from parts of other objects.
Meshes and Deform

The group of tools in this palette support operations that affect the topological resolution of objects as well as their geometry. Rectangular, triangular, flat or smooth meshes can be generated on objects, or the density of such meshes can be reduced. Once meshes are in place, their geometry can be smoothly moved according to a profile shape, it can be deformed, bent, morphed, or it can be displaced according to the shades of an image.

Round and Draft Angles

The first three tools in this palette can be used to apply either bevels or round fillets to points, segments, or both points and segments of objects. The next two tools blend the ends of two surfaces by creating a new surface or a fillet. The last tool can be used to apply a special type of beveling known as draft angles, which are useful for molding.

Smooth Curves

The tools in this palette generate splines from vector lines and manipulate parametric spline curves through operations such as breaking, merging, attaching, etc. The last tool generates curves from preset or user provided formulas.

Nurbz Surfaces

The tools in this palette generate and manipulate NURBS surfaces, called nurbz. The next to the last tool generates a special hybrid type of an object, called c-mesh, which is partly a parametric surface and partly a faceted object. The last tool generates surfaces from preset or user provided formulas.

Patches

The four tools in this palette generate and manipulate a special type of parametric surfaces and solids known as patches.
Metaformz

The two tools in this palette supplement the metaballs generated as primitive objects. They can be used to generate derivative meta-objects, called metaformz, from virtually any form•Z object, and to evaluate the implicit surfaces of any grouped combination of such objects.

Booleans and Intersections

These tools are used to produce new objects from the intersection of other objects. The first four tools are the Boolean operators that are used to derive objects that are the union, intersection, or difference of two objects, or a combination of these operations. They can be applied to either solid or coplanar 2D shapes. The next tool is the Trim/Split operations that separate surfaces at their line of intersection, which can also be returned as an independent object by the next tool. The next tool stitches open edges of objects that touch, the third and second tools from the right end of the palette generate 2D/3D sections from solids and a sequence of such sections known as contours. The last tool on the right applies a special intersection operation to objects derived from the projections of a selected object. This is a low resolution substitute of the original object, called cage.

Join and Group

The first two pairs of tools in this palette merge/unmerge and link/unlink objects into one entity, respectively. The last tool removes clones from the clone family.

Text

These tools are used for the generation and editing of two and three dimensional text.

Symbols

These tools are used for the generation, placement, and editing of 3D modeling symbols. Included is also a tool that is used to “explode” a symbol, which returns it to its components.
Line Editing

These tools offer operations for editing 3D lines, including breaking them, joining them, and beveling/filleting them. Also included are tools for inserting points or segments in surfaces or solids.

Structures

The tools on this palette are used to change topological attributes of objects, such as repositioning the first point of closed shapes, changing the direction of objects, converting one-sided to two-sided objects and vice versa, and marking points. The third from the last tool can be used to convert objects from one type to another and the next to the last tool to convert wire objects to surfaces. The last tool is used to extract the control lines of parametric objects.

Query

The first three tools in this palette are used to retrieve information about objects or their relationships. The first displays information about the geometry and structure of an object; the second about its attributes; the third about its parameters; the fourth is used to measure distances and angles. The fifth and sixth tools diagnose and correct certain types of irregularities, whenever possible. The last tool detects conditions that may present difficulties to 3D printing.

Geometric Transformations

These tools are used to transform the geometry of objects and are the only ones that are affected by two modifiers: the Topological Level and the Self/Copy modifiers. They execute translations (moves), rotations, scalings, and reflections (mirroring). There are also three tools that execute predefined macro transformations.

Relative Transformations

The tools in this palette transform objects relative to other objects. The first tool attaches objects or parts of objects to other objects, the second aligns groups of objects, the third extends segments relative to surfaces, the fourth places objects on other objects, and the last replaces objects.
Attributes

These tools are used to assign attributes such as color, smooth shading, shadows, and textures. There are tools for applying singular or multiple attributes and for copying attributes from one object to another.

Animation

These tools are used to set up the different types of animation available. They are discussed in Chapter 7 and not in this volume.

Animation Utility

These tools also support the animation procedures and are discussed in Chapter 7.

Ghost and Layers

The first two tools in this palette are used to make objects inactive and active. The third tool is used for placing objects on layers.

Delete Objects

The single tool on this palette is used to delete objects, which is the same result produced with the delete key.

Delete Parts

The two tools in the last palette apply deletions at topological levels other than Object or Group. That is, they can be used to delete points, segments, outlines, faces, or holes. The first tool deletes the topology (connections) and the second the geometry of the selected entity. Note that, when deleting points, both tools produce the same results.
4.1 Generating and editing primitives

With the tools on the first row of the modeling tool bar, form•Z offers the ability to directly generate parametric primitives, which are cubes, cylinders, cones, spheres, and tori, as well as four types of quadratic surfaces. They can be generated either through numeric or graphic input, and can be edited and changed after their initial generation, by simply changing the value of their parameters, or through graphic editing. The following tools for generating primitives are available and are discussed in this section:

- Cube
- Cone
- Cylinder
- Sphere
- Torus
- Paraboloid
- Single Hyperboloid
- Double Hyperboloid
- Hyperbolic Paraboloid
- Spherical Objects
- Star
- Metaballs

The first five tools (icons in the first row on the left) generate what are more specifically known as analytic primitives, due to that their generation is based on classic analytic geometry formulas. The next four tools generate quadratic surfaces, which can also be treated as special types of primitives. The last three tools (icons in the first row, right) generate a set of spherical objects, solid stars, and metaballs. Each of these types is parametric but behaves differently when it is edited, as discussed in more detail in the respective sections.

The analytic primitives, as implemented in form•Z, after their initial generation, can be changed to a different personality, including another parametric type. This is done using the Convert tool. The significance of this is that each personality allows the primitives to behave differently, when they are edited and reshaped. In other words, different shapes and forms can be derived from different personalities. The other personalities to which they can be changed are nurbs, patches, and faceted objects.
4.1.1 Representation and display of analytic primitives

Analytic primitives are generated by five tools located by default in the top-left position of the modeling tool bar. Each of these tools is supported by a dialog that is invoked from it and contains the parameters of the respective type of primitive. All these dialogs, except for that of the cube, contain a **Display Resolution** tab (Figure 4.1.1.1), whose options control how a primitive is displayed on the screen. This is done with either a **Simple** or a **Detailed** method.

**Simple**: This is a slider bar whose left and right ends represent “low” and “high”. Positioning the slide marker between the two ends determines how dense the representation graphics of the primitive will be. This is shown in Figure 4.1.1.2.

**Detailed**: The options in this group allow you to set the different parts of the representation of a primitive individually.

The analytic primitives and other smooth objects are graphically represented in two ways: (1) By *wires* that run along their parametrically defined surfaces, more specifically called *edges* and *iso lines* (Figure 4.1.1.3). Edges are the boundary lines of surfaces. Iso lines are the wires that run inside the edges along a surface and help to visualize it in wire frame mode. (2) By *facets* that divide the analytic surfaces into a number of planar, triangular or quadrangular polygons.

Both representations may be used simultaneously or facet display may be turned off. This is controlled by options in the **Wire Frame Options** dialog. The actual structure and density of these representations is set in the **Display Resolution** tab, which contains sections for the wire and faceted representation.

---

**Figure 4.1.1.1**: The **Display Resolution** tab.

**Figure 4.1.1.2**: The **Simple** method for setting the display resolution, shown (1) with and (2) without facets: The slider bar marker set at (a) left end, (b) halfway between ends, and (c) at high end.

**Figure 4.1.1.3**: Edges and iso lines.
**Edges And Iso Lines:** The options in this section determine how many iso lines are drawn for a particular primitive and how smooth the edges and the iso lines are.

**Iso Line Density:** The selection from this pop up menu determines how many iso lines are drawn for a primitive. There are three items in the menu, Low, Medium, and High. Their effect is illustrated in Figure 4.1.1.4.

**Point Density:** The options in this group control the smoothness or resolution with which the edges and iso lines are drawn.

**# Of Segments:** When this option is selected, the resolution of the edges and iso lines is determined by the number of segments entered in its numeric field. The default is 48.

**Max Normal Deviation:** When this option is on, the edges and iso lines are facetted at a density such that the angle between the normals of two adjacent segments is not larger than the angle specified in its text field. This angle is illustrated in Figure 4.1.1.5. With this option the resolution of the edges and iso lines depends on their curvature. The closer the curve the denser the points, and vice versa.

**Max Edge Length:** When this option is selected, the resolution of the edges and iso lines is not a constant number, but depends on their length. The longer they are, the higher the resolution. The program generates as many segments of the size $s$ indicated in its numeric field as it can fit.
Facets: The options in this section determine the density and nature of the facets drawn for a primitive. Facets are used to visualize primitives in all rendering modes, except Shaded Render* and RenderZone*. One or more facetting criteria can be selected in this section, each of which generates a facetted mesh of a certain density. If more than one criterion is selected, the facetting will satisfy all of them, which typically results in a denser mesh. Note that facetting is only applied to the curves of a primitive. For example, only the curved surface around a cylinder’s axis will be facetted, and not its top and bottom faces.

Max # Of Facets: When this option is selected, the numbers entered in its Length and Depth fields determine the resolution of the facets to be created, as follows:

- Sphere, torus have Length x Depth facets.
- Cone, cylinder have Length facets.

If Max # Of Facets is selected by itself, the exact number of faces entered in its Length and Depth fields are generated. If it is combined with any of the other options below, Length and Depth specify an upper limit. For example, if the Max Normal Deviation criterion would normally generate 100 facets around the length direction of a sphere and the Length parameter is set to 24, 24 facets will be generated. On the other hand, if Max Normal Deviation generates only 10 facets for length, 10 facets are created. This mechanism is provided as a safeguard against excessive meshing.

Max Edge Length: When this option is selected, facets with dimensions no larger than the number entered in this field are generated.

Max Normal Deviation: With this option, the primitive is facetted at a density such that the angle between the normals of two adjacent facets is not larger than the angle specified in the text field. This angle is illustrated in Figure 4.1.1.5.

Max Surface Deviation: When this option is selected, the primitive is facetted at a density such that the distance between the centers of the facets and the curved surface is not larger than the distance specified in the text field. This method is illustrated in Figure 4.1.1.6.
4.1.2 Drawing cubes

A primitive cube can be generated using the cube tool. While the generated cube is an analytic primitive, its appearance is not really any different than a regular facetted cube. When in doubt about the type of a cube, the Query tool will tell you what it is.

**Cube**

This tool is used to create a cube graphically or from numeric input. How it is done is determined by options selected in the Cube Options dialog (Figure 4.1.2.1), which can be invoked directly from this tool. This dialog (and all the analytic primitives dialogs) contain a number of mutually exclusive icon button commands. Each represents a different method for generating the primitive.

**PRESets:** When this button is on, form-Z creates a cube using the parameters in the **Width**, **Depth**, and **Height** fields in the Preset section of the dialog. As soon as you click in the graphics window the cube is generated with its local origin positioned where you clicked (Figure 4.1.2.2(a)). The position of the cube’s local origin is at the center of its bottom face.

**DIAGONAL:** When this button is selected, the cube is generated graphically by inputting three points (Figure 4.1.2.2(b)).
- The first click sets one corner point of its base face.
- The second click sets the diagonally opposite point of the base face. After the first click, a rectangle is rubber banded whose sides are constrained to an orientation parallel to the axes of the reference plane. Note that this draws a base as a 2-point rectangle.
- The third click sets the height of the cube.

**AXIAL:** When this button is selected, the cube is generated from four points (Figure 4.1.2.2(c)).
- The first mouse click sets one point of the base face.
- The second click sets the direction of the base rectangle (which is not constrained to the orientations of the reference plane axes).
- The third click sets the size of the base face. Note that this method draws the base as the 3-point rectangle.
- The fourth point sets the height of the cube.
4.1.3 Drawing cones and cylinders

Complete or truncated cones, as well as fully or partially revolved cones can be generated with the Cone tool.

**Cone**

This tool allows you to generate a primitive cone using either numeric or graphic input. The available options are selected from the Cone Options dialog (Figure 4.1.3.1), which is invoked directly from the tool. This dialog (and all the primitive Options dialogs, except Cube Options) has two tabs, Options and Display Resolution. The options in the former tab are discussed below. The options in the latter tab were discussed in section 4.1.1.

**PRESET:** With this button selected, form-Z draws a cone according to the parameters set in the Radius X, Radius Y, and Height fields in the Preset section of the dialog. A single click generates the cone, whose origin is positioned at the click point (see Figure 4.1.3.2(a)). If the Truncated Cone option is selected, the size of the truncated top is expressed as a percentage in the Truncated Top Is n % Of Base numeric field (default is 50%, see Figure 4.1.3.2(b)).
RADIUS: With this button selected, a cone is generated with three mouse clicks (Figure 4.1.3.3(a)).
- The first click sets the center of the circular base.
- The second click sets the radius.
- The third click sets the height of the cone, which can be positive or negative.

DIAMETER: With this button on, a cone is again drawn with three points, but they are different from the points of the previous method (Figure 4.1.3.3(b)).
- The first and second clicks set the diameter of the circular base face. Note that this method draws the base as a circle by diameter.
- The third click sets the height of the cone, which can be positive or negative.

3 POINTS: When this button is selected, a cone is drawn with four clicks (Figure 4.1.3.3(c)). The base of the cone is drawn as a 3 point circle.
- The first three clicks set three points on the circumference of the circular base face.
- The last click sets the height of the cone.

ELLIPSE: With this button selected, the cone is drawn with four clicks, and may have an elliptical base face (Figure 4.1.3.3(d)).
- The first click sets the center of the base.
- The second click sets one radius of the ellipse.
- The third click sets the other radius.
- The last click sets the height of the cone.

Figure 4.1.3.3: Drawing a cone with the (a) RADIUS, (b) DIAMETER, (c) 3 POINTS, and (d) ELLIPSE method.
Cones created with these methods can be truncated and/or generated partially, using the following options:

**Truncated Cone**: Selecting this option causes form•Z to cut off the top of the cone and to replace it with a planar face. Note that this face may be smaller or larger than the base face (Figure 4.1.3.4). With the PRESET method this is set through numeric input. With the other methods its size is set by one additional mouse click.

**Partial**: Selecting this check box allows you to create partial cones whose total circumference is determined by the angles typed in its numeric fields or by using the interactive icon to the left of the fields. The icon is automatically updated when you change the values in the numeric fields. It is a representation of the base of the cone, in top view.

To use the interactive icon, simply click the mouse on one of the small squares found on the perimeters of the larger circle. The hollow square corresponds to the **Start**, while the solid square corresponds to the **End** numeric field. While holding the mouse button down, move the squares until they reach the angles you desire. Pressing the **shift** key while you move the squares, causes both angles to be adjusted in equal numbers of degrees. Whether form•Z will close the first and last points, depends on the following options.

**Closure**: These three buttons determine whether the cone will be closed to become solid or open. Note that these options affect both complete and partial cones.

- **NONE**: Selecting this option causes form•Z to create a surface object that has the outer shape of the cone, but remains open as shown in Figure 4.1.3.5(a).

- **PLANE**: When this button is selected, which is the default, a solid cone is generated for both complete and partial cones. When complete, the conic surface is closed with a base face. When partial, two faces are added to close the solid: one for the base and one that joins the open ends of the cone (Figure 4.1.3.5(b)).

- **CENTER**: When this option is selected, a solid cone is generated again. For complete cones, the resulting object is identical to that created by the previous option. For partial cones, three faces are used to close the solid. The open ends of the cone are closed using two faces that extend to the axis of the cone. A third face is used to close the base (Figure 4.1.3.5(c)).

*Figure 4.1.3.4*: A cone drawn with the **3 Points** method and the **Truncated** option on.

*Figure 4.1.3.5*: The different cone closure types: (a) NONE, (b) PLANE, and (c) CENTER. (1) Complete and (2) partial cones.
Complete or partial cylinders can be generated using the Cylinder tool, which works exactly like the Cone tool does, except that it creates objects by applying a parallel extrusion to a circular base, as opposed to an extrusion to a point that is applied for cones.

Cylinder

The options that affect this tool are selected from the Options tab of the Cylinder Options dialog (Figure 4.1.3.6). The Display Resolution tab is discussed in section 4.1.1. Note that the Options tab is identical to its counterpart in the Cone Options dialog except that it does not have a Truncated option, and icons of the buttons show cylinders rather than cones.

Generating a cylinder is identical to the processes applied for the generation of cones and will not be repeated. The four graphic methods corresponding to iconic buttons (RADIUS, DIAMETER, 3 POINTS, and ELLIPSE) are illustrated in Figure 4.1.3.7. The Closure types are shown in Figure 4.1.3.8.

Figure 4.1.3.6: The Cylinder Options dialog.

Figure 4.1.3.7: Drawing a cylinder with (a) RADIUS, (b) DIAMETER, (c) 3 POINTS, and (d) ELLIPSE method.

Figure 4.1.3.8: Cylinder Closure types: (a) NONE, (b) PLANE, and (c) CENTER.
4.1.4 Drawing spheres

Analytic spheres can be generated through both numeric and graphic input using methods similar to those used for the generation of cone and cylinder primitives.

The options that control the generation of spheres are selected from the Options tab in the Sphere Options dialog, which can be invoked directly from the tool. It is shown in Figure 4.1.4.1. The Display Resolution tab is as discussed in section 4.1.1.

PRESST: Selection of this button generate a sphere according to the parameters set in the Radius X, Radius Y, and Radius Z numeric fields in the Preset section of the dialog. A single click generates the sphere. Its center and local origin are positioned at the click point (Figure 4.1.4.2(a)).

RADIUS: Selection of this button allows you to draw a sphere with two clicks (Figure 4.1.4.2(b)).
• The first click sets the center of the sphere.
• The second click sets its radius (size).

DIAMETER: Selecting this button also allows you to draw a sphere with two clicks (Figure 4.1.4.2(c)).
• The first click sets one point on the surface of the sphere.
• The second click sets the opposite end of a diameter of the sphere that starts at its equator.

Figure 4.1.4.1: The Sphere Options dialog.

Figure 4.1.4.2: Drawing a sphere using the (a) PRESET, (b) RADIUS, and (c) DIAMETER option.
3 POINTS: Selecting this button allows you to draw a sphere with three clicks. When drawing a sphere relative to a reference plane, all three points are on that same plane (Figure 4.1.4.3). However, the points can be anywhere on the surface of the sphere and Point snap can be used to make the newly generated sphere pass through three points in 3D space, as shown in Figure 4.1.4.4.

4 POINTS: Selecting this button allows you to draw a sphere with four clicks, which are four points in 3D space the sphere passes through. When drawing relative to a reference plane, the four points are generally on the same plane. In such a case, the sphere is actually generated from three points. That is, as soon as the fourth point is entered, it is used as a third point and a 3 point sphere is generated. However, point snapping can be used to generate a sphere that passes through four points in 3D space, as illustrated in Figure 4.1.4.5.

ELLIPSE: Selecting this method allows you to interactively draw an elliptical sphere with four clicks (Figure 4.1.4.6).
- The first click defines the center of the sphere.
- The second click sets the end of the sphere’s X axis.
- The third click defines the length of their Y axis.
- The fourth click sets the size of the sphere’s Z axis.

Figure 4.1.4.3: Drawing a sphere with the 3 POINTS method and all points on the same reference plane.

Figure 4.1.4.4: Drawing a sphere by the 3 POINTS method and snapping to predrawn points. (a) The snap points and (b) the 3D sphere.

Figure 4.1.4.5: Drawing a sphere with the 4 POINTS method and Point snap. (a) Snap points and (b) resulting sphere.

Figure 4.1.4.6: Drawing a sphere with the ELLIPSE method.

Modeling • Generating and editing primitives
**Partial**: When this option is on, you can create partial spheres. The length of the circumference may be determined by the angles typed in the **Horizontal** and **Vertical** fields or by using the interactive icons, which work the same as for the cones and cylinders (see previous section). The icon on the left represents a top and that on the right a side view.

Figure 4.1.4.7 illustrates two partial spheres. They both use the same **Start** and **End** parameters, except for **Vertical End**, for which 90° and 45° are set for spheres (a) and (b), respectively. As can be seen, when the angle is less than 90°, the top of the sphere is truncated with a flat plane. Conversely, if a less than 90° angle is entered in the **Vertical Start** field, the bottom of the partial sphere will be truncated.

**Closure**: These three options in the **Sphere Options** dialog determine which type of object will be generated when making a sphere, open or closed. Figure 4.1.4.8 shows the three types of spherical closure.

- **NONE**: When this option is on, a surface object is created. It has the outer shape of the sphere, but is open, as shown in Figure 4.1.4.8(a).

- **PLANE**: Selecting this option (default) creates a closed, solid object, as shown in Figure 4.1.4.8(b). When the sphere is partial (as in the shown example), its open parts are closed with a plane.

- **CENTER**: Selection of this option creates a closed, solid object. When creating a complete sphere, this option produces the same result with PLANE. When creating a partial sphere, as shown in Figure 4.1.4.8(c), the open areas are closed with triangular faces meeting at the center of the sphere.
4.1.5 Drawing tori

Analytic tori, which are ring or donut like shapes, can be generated with the Torus tool.

There are five methods for generating tori. These and other options are selected from the Options tab in the Torus Options dialog (Figure 4.1.5.1), which is invoked directly from the tool. The dialog also contains the Display Resolution tab, which is discussed in section 4.1.1.

PRESSET: When this option is selected, a torus is generated using the parameter values set in the Major Radius, Minor Radius X, and Minor Radius Z fields in the Preset section of the dialog. Their meaning is illustrated in Figure 4.1.5.2. With this option, the torus is generated with a single click of the mouse. The local origin and centroid of the torus are placed where the mouse is clicked (Figure 4.1.5.3).

Figure 4.1.5.1: The Torus Options dialog.

Figure 4.1.5.2: The parameters of the torus.

Figure 4.1.5.3: Drawing a torus using the PRE-SET button.
RADIUS: When this button is selected the torus is drawn with three mouse clicks (Figure 4.1.5.4(a)).
- The first click sets the center of the major radius.
- The second click sets the size of the major radius.
- The third click sets the size of the minor radius.
Note that with this drawing method, the minor X and Z radii are always the same.

DIAMETER: When this option is selected the torus is again drawn with three clicks.
- The first two clicks set the major diameter.
- The third click sets the minor X and Z radii, which, with this drawing method are always the same.
The process is shown in Figure 4.1.5.4(b).

3 POINTS: When this button is selected the torus is generated with four clicks.
- The first three clicks set points on the major circle of the torus, as for the 3 Point Circle.
- The fourth click sets the minor X and Z radii, which, with this method are always the same.
An example is shown in Figure 4.1.5.4(c).

ELLIPSE: When this button is selected the torus is generated with four clicks (Figure 4.1.5.4(d)).
- The first click sets the center of the major radius.
- The second click sets the major radius.
- The third click sets the minor X radius.
- The fourth click sets the minor Z radius.
Note that this is the only method that allows you to create a torus whose cross section is an ellipse. The major circle is always a real circle.

Figure 4.1.5.4: Drawing a torus with the (a) RADIUS, (b) DIAMETER, (c) 3 POINTS, and (d) ELLIPSE method.
**Partial:** When this option is selected, partial tori, relative to both their major and minor radii, can be generated. The parameters in this section of the dialog work as for the spheres (see previous section).

Two examples of partial tori and the parameters used for their generation are shown in Figure 4.1.5.5.

**Closure:** The three options in this section of the **Torus Options** dialog determine whether surface or solid objects will be generated from partial tori, as illustrated in Figure 4.1.5.6. Note that complete tori are always solid objects.

- **NONE:** When this option is selected, partial tori are generated as surface objects and no attempt is made to close their open ends (Figure 4.1.5.6(a)).

- **PLANE:** Selection of this option (default) creates a closed, solid partial torus by generating additional faces that extend between the open ends (Figure 4.1.5.6(b)).

- **CENTER:** This option also creates a closed, solid partial torus. However, this is done differently than the previous method. The partial torus is closed by connecting the open ends with the major circle of the torus (Figure 4.1.5.6(c)), rather than by connecting the open ends directly, as the previous method does.
4.1.6 Drawing paraboloids

Paraboloid

A paraboloid is the surface of revolution of a parabola. The cross sectional curves in the horizontal direction are circles or ellipses, while the cross sectional curves in the vertical direction are parabolas. Paraboloid is the shape used in the reflectors of automobile headlights.

The options that affect this tool are selected from the Options tab of the Paraboloid Options dialog (Figure 4.1.6.1). The Display Resolution tab is discussed in section 4.1.1 of the form·Z User’s Manual. Note that the Options tab is identical to the Cylinder Options dialog except that the icons of the buttons show paraboloids rather than cylinders.

Generating a paraboloid is identical to the processes applied for the generation of cones. The four graphic methods corresponding to iconic buttons (RADIUS, DIAMETER, 3 POINTS, and ELLIPSE) are illustrated in Figure 4.1.6.2.

Figure 4.1.6.1:
The Paraboloid Options dialog.

Figure 4.1.6.2:
Drawing a paraboloid with (a) RADIUS, (b) DIAMETER, (c) 3 POINTS, and (d) ELLIPSE method.
4.1.7 Drawing single hyperboloids

A single hyperboloid is a surface of revolution obtained by rotating a hyperbola about the perpendicular bisector to the line that connects the two foci. The horizontal cross sectional curves of a hyperboloid are circles or ellipses while, the vertical cross sections are hyperbolas.

The options that affect this tool are selected from the Options tab of the Single Hyperboloid Options dialog (Figure 4.1.7.1). The Display Resolution tab is discussed in section 4.1.1. Note that the Options tab is identical to the Cylinder Options dialog except that the icons of the buttons show single hyperboloids rather than cylinders and there is one additional parameter, as follows:

Top Ratio: This parameter is expressed as a percentage of the Waist Radius. The default is 100%.

Generating a single hyperboloid is identical to the processes applied for the generation of cones with the Truncated Cone option selected. The four graphic methods corresponding to iconic buttons (RADIUS, DIAMETER, 3 POINTS, and ELLIPSE) are illustrated in Figure 4.1.7.2.
4.1.8 Drawing double hyperboloids

A double hyperboloid is a surface of revolution obtained by rotating a hyperbola about the line joining its two foci. Its name reflects the fact that this surface normally consists of two branches. However, one of the branches can be dropped, resulting in a single surface. The horizontal cross sections of this surface are circles or ellipses, while the vertical cross sections are hyperbolas.

The options that affect this tool are selected from the Options tab of the Double Hyperboloid Options dialog (Figure 4.1.8.1). The Display Resolution tab is discussed in section 4.1.1. Note that the Options tab is identical to the Cylinder Options dialog except that the icons of the buttons show hyperboloids rather than cylinders.

Generating a double hyperboloid is identical to the processes applied for the generation of cones. Whether both branches of the double hyperboloid surface are generated or just one depends on whether the Half Only option is on or off. Both variations and the four graphic methods corresponding to the iconic buttons (RADIUS, DIAMETER, 3 POINTS, and ELLIPSE) are illustrated in Figure 4.1.8.2.
4.1.9 Drawing hyperbolic paraboloid

A hyperbolic paraboloid is a quadratic doubly ruled surface. The cross sectional curves in one direction are hyperbolas, while in the other direction are parabolas. They are often referred to as saddle surfaces.

The options that affect this tool are selected from the Options tab of the Hyperbolic Paraboloid Options dialog (Figure 4.1.9.1). The Display Resolution tab is discussed in section 4.1.1. Note that the Options tab is identical to the Cube Options dialog except that the icons of the buttons show hyperbolic paraboloids rather than cubes.

Generating a hyperbolic paraboloid is identical to the processes applied for the generation of cubes. The three graphic methods corresponding to iconic buttons (PRESET, DIAGONAL, and AXIAL) are illustrated in Figure 4.1.9.2.

Figure 4.1.9.2: Drawing a Hyperbolic Paraboloid with (a) PRESET, (b) DIAGONAL, and (c) AXIAL method.
4.1.10 Editing analytic primitives and quadratic surfaces

After their initial generation, analytic primitives can be edited to change their shape. This is done using the Edit Controls or Edit Surface tools. Their shape can also be changed using the Query Object tool.

Editing the controls of primitives

With the postpick selection method, clicking on a primitive with the Edit Controls tool sets it in **controls edit mode** and displays its controls. This allows you to edit one object at a time, which is the most common method of editing. However, you can also edit more than one primitive object at the same time, using the prepick selection method. With the Pick tool, click on the objects you wish to edit to preselect them. Then, with the Edit Controls tool, click anywhere in the window. All the preselected primitives are set into edit mode and their controls are displayed.

There are two types of controls, as shown in Figure 4.1.10.1 and 4.1.10.2.

- **Size control**: This is an arrow drawn green. It represents the size parameters of the respective primitives.

- **Angle control**: This is a blue arrow with a 90 degree corner. It represents the start and end angle of a primitive’s closure.

Two methods are available for editing and transforming the controls of a primitive:

- **Click-and-drag**: With this method a single control is changed at a time. You click on a control and, as you keep the mouse button pressed, you drag it to a new position and release it.

- **Click-and-click**: With this method one or more controls can be selected to be transformed at the same time. While you press **shift** you click on the controls you wish to preselect. Then you click on one of the controls again to initiate the move. You move the mouse to where you want it to go and you click again. The motion is applied to all the preselected controls. Note however that, to move more than one control with one operation, they must all be of the same kind and they must be on different primitive objects being edited at the same time. That is, angle and size controls can not be mixed even when they are on different primitives and two size controls can not be changed when on the same primitive.

Figure 4.1.10.1: The controls of primitives.

Figure 4.1.10.2: The controls of quadratic surfaces.
If the lock option of a primitive is on and a size parameter is edited, the locked parameters will change simultaneously. The lock option can be turned on/off in the dialogs invoked when clicking the **Edit...** button in the **Query Object** dialog, as discussed at the end of this section.

**Editing the surface of primitives**

Clicking on a primitive with the Edit Surface tool highlights the object and sets it in **surface edit mode**. No controls of the object are displayed, but a little arrow that represents the normal of the surface to be moved is shown, as illustrated in Figure 4.1.10.3 for a sphere. This editing mode can only be applied using the postpick selection method, only on one primitive at a time, and the **click-and-click** method is used exclusively.

Transformations are applied immediately as soon as the object is picked. That is, clicking on a primitive selects it and an arrow is displayed at the position of the click, which follows the motion of the mouse, until the next click, which completes the surface move operation. Note that this operation only affects the size of the object and can not be used to change the closure angles. The object is resized in such a way that the surface of the object always passes through the point, which is moved by the mouse along the normal direction of the surface. Per type of analytic shape, the following surface move operations are available:

**Cube:**
- Any of its flat faces can be moved along their normal.

**Cylinder:**
- Clicking on its curved surface changes its radius.
- Clicking on its flat ends moves them along the axis of the cylinder.

**Cone:**
- Clicking on the curved surface changes the radius of its base, which also changes the angle of the cone.
- Clicking on the curved surface while pressing **control** (Macintosh) or **ctrl+alt** (Windows) changes the radius throughout the cone, which effectively transforms it into a truncated cone, and vice versa. This is illustrated in Figure 4.1.10.4.

- Clicking on its flat base moves it along its axis.

**Sphere:**
- Its radius changes, affecting overall size.

**Torus:**
- Its minor radius (of profile) changes.
Editing the parameters of primitives

While the Edit Controls and Edit Surface tools allow you to edit and change the shape of an analytic primitive graphically, its parameters can be edited and changed numerically using the Query tool.

With topological level set to Object, clicking on an analytic primitive with the Query tool invokes the **Query Object** dialog, which contains information about it, including what type of primitive it is. It also contains an **Edit...** button in its upper right area. Clicking on this button invokes an **Edit** dialog for the respective primitive. This dialog is very similar to the **Options** dialog invoked from the icon of a primitive, except that it also includes a preview area and positional parameters of the primitive. The **Torus Edit** dialog is shown in Figures 4.1.10.5.

**Preset**: These fields display the settings that currently exist in the primitive being edited. When these parameters are changed, they are applied to the primitive as soon as you click **OK** to exit the **Query** dialog. When editing a size parameter and the lock option (■) for that primitive is on, all locked radii are changed at the same time. By default, this option is on if the primitive is created with equal radii. If the primitive is created as an elliptical shape, this option is defaulted to off.

For each type of analytic primitives the lock option applies as follows:

- Cube: **Width**, **Depth**, **Height**.
- Cone, cylinder: **Radius X** and **Y**.
- Sphere: **Radius X**, **Y**, and **Z**.
- Torus: **Minor Radius X** and **Z**.
4.1.11 Spherical objects

Eight spherical objects can be generated directly as primitives, using a tool on the first row of the right column of the tool bar. They are the five Platonic solids, the soccer ball, revolved spheres, and geodesic spheres. Intuitively speaking, the lower end Platonic solids do not appear to have a round shape. However, because all their points are at the same distance from their center, and their generation process is similar to that of the spheres, they belong in the same category as the spheres. Revolved spheres can also be generated as objects of revolution by revolving a semicircle about an axis. The geodesic spheres consist of equal triangles, and can be generated at four levels of resolution.

All spherical objects are parametric facetted objects. They are stored with their parameters, which can be changed later. You can also change the type of object. That is, a geodesic sphere can be switched to a revolved sphere, a soccer ball can become an icosahedron, etc.

Spherical Object

This tool is used to generate eight variations of spherical objects. The type of spherical object to be generated, its parameters, if any, and the method of generation are selected from the Spherical Object Options dialog (Figure 4.1.11.1), which can be invoked directly from the Spherical Object tool. In addition to options, the Spherical Object Options dialog contains a preview box in which the type of object currently selected from the Shape pop up menu is displayed. Clicking in the preview box itself moves the selection to the next type, and allows you to parade through all the types.

All spherical objects are generated using one of six methods, which is selected from the iconic buttons on the top of the dialog. These are the same drawing methods used for the generation of the analytic spheres (see subsection 4.1.4).

Figure 4.1.11.1: (a) The Spherical Object Options dialog and (b) the Shape pop up menu.
**Preset:** The values in the **Radius X**, **Y**, and **Z** fields in this box of options are applied when the PRESET method (first icon on the left) is selected and used for the generation of a spherical object. They are ignored by all the other methods, which set the size of a spherical object interactively.

**Shape:** The types of spherical object to be generated are selected from this pop up menu. The types available include the five Platonic solids, a soccer ball, a revolved sphere, and a geodesic sphere. The Platonic solids (Figure 4.1.11.2) and the geodesic sphere (Figure 4.1.11.3) are bounded by faces that have the same shape and equal size.

The **Tetrahedron** is bounded by four equal triangles.

The **Hexahedron** is bounded by six equal squares, and is more commonly known as the cube.

The **Octahedron** is bounded by eight equal triangles.

The **Dodecahedron** is bounded by twelve equal pentagons.

The **Icosahedron** is bounded by twenty equal triangles.

The **Soccer Ball** is bounded by twenty-four faces, eight of which are equal pentagons, and sixteen are equal hexagons (Figure 4.1.11.2). The soccer ball is not a simple Platonic solid. It can be derived by intersecting a dodecahedron with an icosahedron.

When any of these types of spherical objects is selected, there are no additional options. However, more options show up when one of the spheres is selected, as follows:
**Revolved Sphere**: When this item is selected from the Shapes pop up menu, a sphere is derived by revolving a semi-circle about an axis (Figure 4.1.11.3). Similar objects can be generated using the Object of Revolution tool. The following options, which only appear when this item is selected, determine the resolution of the revolved sphere.

**Length Resolution**: This value specifies the number of segments with which the semi-circular source profile will be generated. The number entered in the field is the number of segments used in the complete circle, therefore half of it defines the semi-circle. This number must be even, and no less than 4.

**Depth Resolution**: This value determines how many times the semi-circle will be revolved around the axis of revolution. This number can not be less than 3.

**Geodesic Sphere**: When this item is selected from the pop up menu, a sphere that is bounded by triangles is generated. These triangles can be subdivided 2, 3, 4, 5, etc. times. The number of triangles per original triangle a certain number of subdivisions produces is given by the square of this number. For example, a subdivision of 5 produces $5 \times 5 = 25$ new faces for each of the original triangles. The total number of faces per **# Of Subdivisions** is given by:

$$\text{# of faces} = (\text{# of subdivisions})^2 \times 20.$$

**# Of Subdivisions**: When this method is used, the value entered in its field specifies the number of subdivisions to be applied to the original triangles of an icosahedron, as shown in Figure 4.1.11.4.

**Edge Size**: When this method is selected, the value entered in its field represents the edge size (the length of the sides of the triangles) desired for the sphere. From the edge size, the level of resolution is calculated. Note that the same edge size will result in different level densities for different sizes of spheres.

*Figure 4.1.11.3*: A revolved sphere.

*Figure 4.1.11.4*: Geodesic spheres generated with **# Of Subdivisions** set to (a) 3, (b) 5, (c) 7, and (d) 10.
Editing spherical objects

Spherical objects can be edited using the Query tool. That is, with the Query tool active and topological level set to Object, click on a spherical object. This invokes the **Query Object** dialog, which has an active **Edit...** button at its upper right area. Clicking on this button invokes the **Spherical Object Edit** dialog shown in Figure 4.1.11.5. As with the primitive objects, this dialog is very similar to the dialog invoked from the Spherical Objects tool, but has a different type of preview and positional parameters. This dialog can be used to change the parameters as well as the type of the spherical object.

**Dimensions**: These fields display the three size parameters of the spherical object, all of which are locked to the same value, except when the sphere was generated with the ELLIPSE method. However, this can change by deselecting the lock option. Any of these fields can then be changed independently to change the respective dimension of the spherical object.

**Shape**: The object type displayed in this pop up menu field is the current spherical object type. Selecting another item changes the object to another type.

*Figure 4.1.11.5: The Spherical Object Edit dialog.*
4.1.12 Metaballs

The metaballs belong to a parametric category of objects called metaformz. From these objects, only the metaballs can be generated directly as primitives. The other metaformz are generated as derivative objects and are discussed in section 4.16. The metaballs are generated using the second tool on the first row on the right of the modeling tool bar.

The metaformz, including the metaballs, are parametric objects that have the ability to blend smoothly when they overlap. A metaform has a region of influence, which is determined by its radius. This region of influence interacts with other metaformz to create a blended surface known as implicit surface. The workings of the metaformz are discussed in more detail in section 4.16.

There are four types of metaballs that can be created directly as primitives: ball, stretched ball, ellipsoid, and stretched ellipsoid. They are illustrated in Figure 4.1.12.1. Note that the stretched balls and the ellipsoids look similar, but they are parametrically different and behave differently.

Individual metaformz may be unevaluated, which they always are when they are generated, or evaluated. The former are displayed as wire frames that represent their parameters and cannot be rendered. The latter are displayed with their implicit surfaces and can be rendered. The examples in Figure 4.1.12.1 illustrate both unevaluated and evaluated metaformz.
**Metaballs**

This tool is used to create four types of metaballs. It works in a manner similar to the Spherical Object tool. Which type of metaball it generates depends on the option selected from the **Metaball Options** dialog, shown in Figure 4.1.12.2.

**Ball**: This is a sphere with a uniform radius throughout. It requires two mouse clicks. The first is its center. As soon as it is entered, a sphere is rubber banded and follows the motion of the mouse. While rubber banding you can both increase/decrease the radius and rotate the position of the sphere. The second click generates the sphere. An example is shown in Figure 4.1.12.1(a).

**Stretched Ball**: This is a sphere that has independently specified horizontal, vertical, and perpendicular radii. When drawing relative to the XY reference plane, horizontal is X, vertical is Y, and perpendicular is Z. For other reference planes these directions are analogous. The stretched ball requires four clicks. The first two are as for the ball. After the second click, the horizontal dimension freezes, rubber banding continues and you can increase or decrease the vertical and perpendicular dimensions. After the third click, the horizontal and vertical dimensions freeze, rubber banding continues and allows you to set the perpendicular dimension. After the fourth click the stretched ball is generated. An example is shown in Figure 4.1.12.1(b).

**Ellipsoid**: This is a spherical shape that looks like a stretched ball, but blends differently. It has two centers and can be generated through two different methods.

**By Ellipse Points**: When this option is on, the ellipsoid is generated with three clicks, by first setting the size of its axis and then the size of its radius. The first click enters one end of its axis. After the first click the axis line is rubber banded and you can both set its size and rotate its position. Note that, at this point the radius of the ellipsoid is 0 and the ellipsoid itself coincides with its axis. After the second click, as soon as you move the mouse a little, an ellipsoid is rubber banded. As you move the mouse you are setting the size of the radius, which is used at both ends of the axis to construct the ellipsoid. The third click completes the generation of the ellipsoid.
By Extents: When this option is on, the ellipsoid is again generated with three mouse clicks, but the sequence in which its parameters are entered is different. After the first click, an ellipsoid is rubber banded. At this point the ellipsoid has a constant radius, which is determined by the value you enter in the Default Radius field. After the second click, rubber banding continues and you adjust the size of the radius. After the third click, the ellipsoid is generated.

Stretched Ellipsoid: This is an ellipsoid with different perpendicular and vertical dimensions. That is, when working in XY, Z is different from Y. It is generated with four clicks. For both By Ellipse Points and By Extents methods, the first three clicks are as for the ellipsoid. After the third click, you can set the size of the perpendicular dimension.

Examples of an ellipsoid and a stretched ellipsoid are shown in Figure 4.1.12.1(c) and (d).

At the lower end of the Metaball Type dialog is the Weight slider which has a major effect on how metaformz interact with each other when they are evaluated.

Weight: This slider bar controls the intensity with which a metaform is blended and whether it will have an additive or subtractive effect. A value for this parameter can also be typed in its numeric field. Positive values have an additive and negative values have a subtractive effect, as illustrated in Figure 4.1.12.3. These matters are discussed in more detail in section 4.16.

As already mentioned, when the metaballs are first generated using the Metaball tool, they are in an unevaluated form. To evaluate them the Metaformz Evaluate tool is used, which is discussed in section 4.16. Unevaluated and evaluated metaballs are shown in Figure 4.1.12.3(1) and (2), respectively.

Both evaluated and unevaluated metaballs can be edited to change their parameters. This is done with the Derive/Edit Metaformz tool, discussed in section 4.16.
4.1.13 Stars

Different star like solids, with a varying number of rays, can be generated. The geometry of a star object is based on the shape of an object that can be created using the Spherical Object tool, discussed in section 4.1.11. These objects are symmetrical around a center with a regular arrangement of 3, 4, 5, or 6 sided faces. A star uses such an object as the base shape. Each face of the base object is subdivided by placing a point in its center and connecting it to the face’s corners. The point is then moved outward in a direction perpendicular to the face. This creates a ray. The process is illustrated in Figure 4.1.13.1.

This tool is used to generate eight variations of a star, which is a parametric object, whose parameters can be edited through the Query Object dialog. The type of star and its parameters are selected from the Star Options dialog (Figure 4.1.13.2), which is invoked directly from the tool.

**Base Type**: The base shape from which the star is generated can be selected from this menu. They are: **Tetrahedron** (4 triangles), **Hexahedron** (6 squares), **Octahedron** (8 triangles), **Dodecahedron** (12 pentagons), **Icosahedron** (20 triangles), **Soccer Ball** (8 pentagons and 16 hexagons), **Geodesic Sphere 1** (80 triangles), and **Geodesic Sphere 2** (320 triangles). Stars created with each base type are shown in Figure 4.1.13.3.
**Dynamic:** When this option is selected, the size of the star is determined interactively. The first click in the project window places the star’s center. After the first click the radius of the star is fit between the first point and the current location of the mouse. This gives an interactive preview of the size of the star. When the mouse is clicked for the second time, the radius of the star is fixed and the operation is completed.

**Preset:** When this option is selected, the construction of the star requires only one click, which determines the center of the star. The other parameters are taken from the following two options.

**Radius:** The value entered in this field specifies the overall size of the star. How much of this is allocated to the core of the object and how much to the rays is controlled by the following parameter.

**Ray Ratio:** This slide rule determines how long the rays are relative to the star’s core. If the ray ratio is set to 100%, the core is large and the rays extent only a small amount. If the ray ratio is 0%, the core is small and the rays extent far. Examples of the same star with ray ratios of 0%, 25%, 50%, 75%, and 100% are shown in Figure 4.1.13.4.

![Stars with different ray ratios](image)

*Figure 4.1.13.4:* Stars with different ray ratios: (a) 0%, (b) 25%, (c) 50%, (d) 75%, and (e) 100%.
4.2 Direct object generation

In addition to the primitives that are generated by the tools on the first rows of the tool bar, the modeling environment of form•Z offers two more methods by which objects can be generated: they are referred to as the direct and derivative object generation methods. With the direct method, an object is generated immediately when a shape is drawn by the user. The derivative object generation method creates a new object from an existing object or part of an existing object, rather than from a drawn shape. This section discusses the direct object generation method. The generation of derivative objects is discussed in section 4.5.

To generate an object by the direct method, you first select one of the Object Type (or Direct Generation) modifiers on the left second row of the tool bar. You then draw a shape, called the source shape, with one of the drawing tools. This source shape is typically two dimensional and is drawn on the current reference plane. The shape may be open or closed. All shapes may be created through graphic or numeric input.

The drawing tools of form•Z are structured in two groups. The first group consists of the regular polygons, circles and ellipses, and occupies the left third row of the modeling tool bar. The second group consists of the single point, single segment, vector line, splines, stream line, and arcs. They occupy the right third row of the tool bar. The shapes in the first group are always closed and concentric. Except for the first two, the tools in the second group produce shapes that may be open or closed. The first two produce single point and single segment shapes, which are always open.

When generating 2D objects, some of the drawing tools default to and generate plain objects and some generate parametric. Parametric objects are generated when we draw circles, ellipses, splines, and arcs. Plain objects are generated by all the other drawing tools (single point, single segment, vector line, rectangles, polygons, and stream lines). It is also possible to draw composite lines using different drawing tools. If such a line includes a portion that was drawn with a Spline or Arc tool, then the whole line is parametric. Parametric and plain objects can be edited to change their shape. When generating 3D objects, such as extrusions and convergences, in addition to that the drawn source shape may be parametric and editable, the object is also parametric relative to its height, which can be edited and changed. Parametric objects can also be changed to plain using the Convert tool (see 4.21.9).

Dialogs can be invoked from most of the Object Type modifiers that contain options affecting the generation of the objects. Dialogs can also be invoked from most of the drawing tools, most of which include a pair of options that allows the generation of either faceted or smooth types of models. These options are not available for the Point, Segment, and Vector Line tools, which always generate faceted objects.
4.2.1 Types of objects

When one of the direct object generation modes is active, the source shape you draw with one of the drawing tools is used to generate a surface or a solid object. Two types of surface objects and three types of solid objects may be created through the direct object generation method.

2D Surface/Wire

These objects are exactly the shape you draw. They can be open or closed, they are not solids, and they are typically planar, except when object snapping is used and free form shapes are drawn with their points snapped to parts of other objects.

The surface objects are essentially 2D shapes, typically created on the flat surface of the reference plane (Figure 4.2.1.1). However, internally they exist in 3D space, can be manipulated in 3D space, and can be transformed to true 3D objects.

The direction of the boundary line of closed surface objects is significant. One of their sides is positive (clockwise boundary) and the other is negative (counterclockwise boundary). All closed surface objects are initially generated with their positive side on the positive side of the reference plane on which they are drawn. When you draw a negative closed shape, the system automatically reverses it before generating the surface object.

Note that the side from which a closed shape is viewed may determine if it is visible. For all display types, except for wire frame, the object is not visible when viewed from its “back side” (Figure 4.2.1.2) and Show Surfaces As Double Sided in the Display menu is off. However, this item is on by default, which makes the closed shapes visible regardless of their direction.

Figure 4.2.1.1: 2D surface objects.

Figure 4.2.1.2: Two hexagons drawn on (a) the XY and (b) the ZX planes. (c) When viewed in hidden line from -Y and Show Surfaces As Double Sided is off, the second hexagon is not visible.
As already mentioned, surface objects are typically drawn on a reference plane and they are planar shapes. However, they can also be drawn as non-planar 3D shapes, which are called wires. How a non-planar wire can be drawn using point snapping and a cuboid to snap to is illustrated in Figure 4.2.1.3. Following is how it can be done:

- Draw a cuboid, roughly as shown in Figure 4.2.1.3(a).
- Turn on Snap to Point and select the 2D Surface/Wire modifier.
- Select the Draw Vector Line tool and with it click at points 1, 2, 3, 4, and 5. Then triple click at point 6, to close the shape. The result is the non-planar wire shape shown in Figure 4.2.1.3(b).

If you repeat the process and this time draw with the B-Spline, Cubic tool, the result will be as in 4.2.1.3(c). You can actually use any of the spline tools to draw a wire object.

Figure 4.2.1.3: Drawing a wire object by snapping to the points of a cuboid: (a) the drawing process, (b) drawing a vector line, and (c) drawing a spline.

Wire objects can not be rendered or, in other words, when they are rendered they behave as the open vector lines or open splines do. However, closed wire shapes can be converted to meshed surfaces, when they satisfy certain conditions. Once converted, they can be rendered. This is illustrated in Figure 4.2.1.4 and is discussed in more detail in section 4.21.10.

Figure 4.2.1.4: (a) A wire object (b) converted to a surface and (c) rendered.
2D Enclosure

To generate these objects form\-Z constructs two parallel lines for each of the segments you draw, and perpendicular lines at the endpoints of open shapes. It then calculates and “cleans” the points of intersection between the parallel lines, creating objects that are always closed. If the source shape is open, the corresponding double line object consists of a single closed outline. If the source shape is closed, the corresponding double line object consists of two outlines: an outer outline which is positive, and an inner outline which is negative (hole), when viewed from the positive side of the plane they are drawn on. The two outlines are structured together as a face (Figure 4.2.1.5). These double line surface objects are called 2D enclosures, which have a 3D counterpart, discussed later in this section.

For the generation of double line objects, the shape you draw functions as an axial shape. How the parallel lines are positioned relative to the axial shape is set by the Justification option selected in the 2D Enclosure Options dialog, shown in Figure 4.2.1.6. This is invoked directly from the 2D Enclosures tool.

**Left, Center, Right**: Selecting one of this options justifies the parallel lines accordingly. The default is Center. The Justification options are illustrated in Figure 4.2.1.7.

**Wall Width**: This value determines how far from the axial shape the parallel lines will be generated. When justification is Center, half of this value is used for placing a line on each side of the axial shape. The default value is 1' - 0" (or 0.25 m in metric).

Note that, while options are typically set in the dialog before the drawing process starts, it is also possible to change these options during the drawing process.

---

**Figure 4.2.1.5**: Construction of a 2D enclosure: (a) a top view (XY projection) and (b) a 3D view.

**Figure 4.2.1.6**: The 2D Enclosure Options dialog.

**Figure 4.2.1.7**: Enclosure Justification options: (a) Left, (b) Center (default), and (c) Right.
3D Extrusion

These objects are generated by sweeping the source shape in a direction perpendicular to the current reference plane, or the plane of the source shape, and to a distance equal to the current height value. Examples are shown in Figure 4.2.1.8. There are two methods by which the height can be applied: the interactive and the preset method. They are both selected from the Heights menu (see section 3.4).

The first item in the Heights menu (Graphic/Keyed) is the interactive method. When selected, as soon as you draw the source shape a solid object is rubber banded and its top face follows the motion of the mouse. That motion is locked to a direction perpendicular to the currently active reference plane, or to the plane of the shape you drew. The next click of the mouse establishes the height and completes the generation of the 3D object. Instead of clicking the mouse to determine the object’s height graphically, a number can be entered into the Prompts palette through the keyboard, next to the “Height:” prompt. Pressing the return/enter key completes the numeric input and the generation of the object.

To define an object’s height using the preset method, select any of the numeric items in the Heights menu. As discussed in section 3.4, the Heights menu initially appears with a number of preset height values, which can be changed or added to.

When using the preset height method an extruded object is rubber banded as soon as drawing proceeds. The shape of this object continually changes until the drawing is completed. It is possible to switch a Heights value while drawing. It is also possible to switch between a fixed height and the interactive height method (Graphic/Keyed) while drawing. Also, the active surface style can be changed while drawing which will update the color of the object being drawn.

Generally, an extruded solid consists of a bottom (base) and a top face, which can be of any shape, and a number of side faces. These are flat, when a faceted type is generated, or can be curved, when a smooth type is created. If an extrusion is generated from a closed shape, it is always a well formed solid (Figure 4.2.1.8). When it is generated from an open source shape, it is a surface object.

The direction of the extrusion is set in the Extrusion/Convergence Options dialog (Figure 4.2.1.9), which is invoked directly from the 3D Extrusion tool.

**Perpendicular To Reference Plane:** When this options is selected, which is the default, the base of an extruded solid is positioned on the reference plane where the source shape is drawn, and the direction of the extrusion is perpendicular to the reference plane.

**Figure 4.2.1.8:** 3D extruded objects.

**Figure 4.2.1.9:** The Extrusion/Convergence Options dialog.
When a numeric height is selected in the **Heights** menu, and this height is positive, the extrusion occurs on the positive side of the reference plane. When the height is negative, the object is extruded on the negative side of the reference plane. For example, if the source shape is drawn on the XY reference plane and a positive height is used, the extruded object has a height of some positive Z value. If a negative height is used, the height of the extruded object is at some negative Z position. In both cases, the base of the object is on the reference plane at Z = 0' 0" (Figure 4.2.1.10).

It is also possible to draw the base of an object away from the reference plane by using object snapping (see subsection 2.2.6). When this is done and this option is selected, the base shape will still be extruded in a direction perpendicular to the reference plane.

**Perpendicular To Surface**: When this option is selected, if the base shape is drawn on the reference plane or at an orientation parallel to the reference plane, this option has the same effect. However, when using one of the Object Snap options (Window tool palette), shapes can be drawn in positions and orientations different from the reference plane. In these cases, selection of this option produces extrusions perpendicular to the surface of the shape you drew. Closed polygonal shapes and circles are always drawn on flat (planar) surfaces determined by the input points (see Figures 4.2.1.11 and 4.2.1.12). Open vector lines may be drawn in 3D space and all their points may not lie on a single plane. In such a case, the first three points of the line are used to determine the surface perpendicular to which the extrusions will be constructed.

**Warning**: The **Perpendicular To Surface** option is **direction sensitive** (as it should be). If you are using one of the object snap options, and you draw a shape clockwise, and then the same shape counterclockwise, they will be extruded in opposite directions when using the same height.
The **converged objects** or **extrusions to a point** are generated as are the extruded solids, except that the extruded shape is converged into a single **apex**, which replaces the top face. The side faces of a facetted converged solid have triangular shapes. The apex is positioned on a line that runs through the centroid of the source shape, and is perpendicular to the reference plane or to the surface of the source, depending on the option selected in the **Extrusion/Convergence Options** dialog (Figure 4.2.1.13), which can be invoked directly from the 3D Converged tool. The apex is on the positive side of the reference plane when a positive height is used, and on the negative side for negative heights. Converged solids work well with the regular polygons, which produce pyramids. They may not produce a meaningful form when an irregular source shape is used. Examples of converged solids are shown in Figure 4.2.1.14.

The heights of the converged objects are selected as for the extruded objects and either an interactive or a preset method can be used. These objects can also be generated as derivative objects (see subsection 4.5.3).
3D Enclosure

These objects are created by applying a special type of an extrusion to a double line surface object that is derived first from the source shape you draw (Figure 4.2.1.15). This intermediary step is executed by the system, and is not visible to the user. The semantics of the positive and negative heights are identical to the extruded solids.

From a practical point of view, the 2D and 3D enclosures facilitate the modeling of architectural floor plans and buildings. However, they are also useful in a variety of other areas.

Double clicking on the 3D Enclosures tool invokes the 3D Enclosure Options dialog shown in Figure 4.2.1.16. It contains the parameters that affect the generation of 3D enclosures. The 3D enclosures may have their top and bottom “slabs” open or closed, which results in a total of four possible combinations or 3D enclosure types. The desired type and the thickness of the slabs are set in the 3D Enclosure Options dialog.

**Justification, Enclosure Wall Width:** As for the 2D enclosures.

**Enclosure Top Thickness:** The value entered in this field determines how thick the top slab (ceiling) will be, when the top of the enclosure is closed. Default value is 1'-0" (or 0.25 m in metric).

**Enclosure Base Thickness:** The value entered in this field determines how thick the bottom slab (floor) will be, when the bottom of the enclosure is closed. Default value is 1'-0" (or 0.25 m in metric).

Figure 4.2.1.15: Generation of 3D enclosures:
(a) A 2D shape is drawn first.
(b) A 2D double line shape is derived.
(c) A 3D enclosure is derived.

Figure 4.2.1.16: The 3D Enclosure Options dialog.
**Top, Base: Open, Closed:** Selection of these options determines the type of the 3D enclosure. The four types are shown in Figure 4.2.1.17. The default type is both top and bottom open. When the bottom and top are both closed, the 3D enclosure contains a space that is completely inside it.

**Perpendicular To Reference Plane, Perpendicular To Surface:** As for the 3D Extrusions.

*Figure 4.2.1.17:* Types of the 3D enclosures shown in wire frame and in hidden line plots:
(a) bottom and top open,
(b) bottom closed, top open,
(c) bottom open, top closed,
and (d) bottom and top closed.
4.2.2 Drawing polygons

A polygon in form•Z is always a symmetric closed shape that you draw with one of the tools in the left third row of the modeling tool bar. A special type of a polygon is the rectangle that you can draw parallel to the orthogonal axes or in a rotated position.

All the polygonal shapes are created by two or three input points, each entered by a distinct click of the mouse. In most cases, the shape you draw is rubber banded after the first click. Note that, unlike other applications where the mouse is dragged between the first and the second point, in form•Z each point is entered through an independent click.

**Rectangle**

It draws a rectangle from two input points that represent two diagonally opposite corners of the rectangular shape. The shape you draw is rubber banded after the first click of the mouse. Any pair of opposite corners may be used, as shown in Figure 4.2.2.1. At generation time, the sides of the rectangle are always parallel to the X, Y axes of the current reference plane. The horizontal and vertical dimensions of the shape may be equal, forming a **square**, or they may differ.

**Rectangle, 3 Point**

It draws a rectangle in a rotated position from three input points. The first and the second points determine the orientation or rotation angle of the rectangle. A line is rubber banded after you enter the first point. After the second point, a rectangle is rubber banded. The third point determines the final size of the rectangle and completes its generation. The process is illustrated in Figure 4.2.2.2.

**Figure 4.2.2.1:** Drawing rectangles. Any pair of opposite points can be used.

**Figure 4.2.2.2:** Drawing a 3 point rectangle: Clicking on (1) a line is rubber banded, clicking on (2) a rectangle is rubber banded, and clicking on (3) completes the generation of the rectangle.
This tool draws polygons from two or more points. The shape drawn is rubber banded after you enter the first point. By default, a hexagonal shape is generated and the input points represent the center and the radius of the circle in which the polygon is inscribed. However, polygons with any number of sides can be generated and the input points can be interpreted in a variety of different ways. These options are selected from the Polygon Options dialog, shown in Figure 4.2.2.3, which is invoked directly from the Polygon tool. This dialog contains two tabs, Edges and Patterns, as well as options outside the tabs.
Options outside the tabs

**Polygon Drawing**: This group of options determines how the two input points will be interpreted and whether a point or a segment will be positioned where the second point is entered. The available options are illustrated in Figure 4.2.2.4.

- **Center & Radius**: This is the default option. The first point defines the center of the polygon. The second determines the size of the radius of the circle within which the polygon is inscribed, as well as the orientation of the polygon (Figure 4.2.2.4(a)).

- **Diameter**: The first point sets the position of a point on the circumference of the circle within which the polygon is inscribed. The second point sets the length and the orientation of the diameter of the circle (Figure 4.2.2.4(b)).

- **X, Y Dimensions**: This option is similar to the previous, except that the position of the second point relative to the first point is interpreted independently for the horizontal and the vertical direction. This input option results in polygons that are symmetric about the line between the input points, but may be stretched (Figure 4.2.2.4(c)).

- **Construct Through Point**: This is the default option. It inscribes a polygon by positioning a point (vertex) at the position of the second input point.

- **Construct Through Segment**: It inscribes a polygon by positioning a segment (edge) at the position of the second input point.

The two options at the lower end of the **Polygon Options** dialog have a global effect. That is, they can be turned on or off from this dialog, as well as from dialogs of other drawing tools. Their status affects all drawing tools.

**Allow Intersecting Lines**: When this option is on, shapes can be drawn that intersect themselves. When off, such shapes are disallowed and, when drawn, an error message is posted.

**Allow Colinear Points**: When this option is on, any colinear points that may have been drawn with a shape are preserved. When off, colinear points are filtered out.
The Edges tab

Simple polygons are generated when **Pattern** is off in the **Pattern** tab (see below). The generation of polygons is controlled by the options in this tab, which specify the number of sides in a polygon by one of two methods, as follows:

**By # Of Segments**: When this option is on, polygons are generated with the specified number of sides, regardless of their size. There are two methods for determining the number of sides (and points) in a polygon. For a “quick” visual selection of the most frequently used polygons, six icons are provided for selecting **triangles** (3 sides), **diamonds** (4 sides), **pentagons** (5 sides), **hexagons** (6 sides), **octagons** (8 sides), and **decagons** (10 sides). Any number greater than two can also be entered in the numeric field to determine the number of sides in a polygon. Whenever an icon is selected, the value in the numeric field is updated and always reflects the currently selected number of polygon sides. The default is 6.

**By Size Of Segments**: When this option is selected, the number of sides the polygon will have is not determined by an explicit \( n \) value, but by the size of its segments, which is controlled by the value entered in its numeric field. When this option is selected, the **By # Of Segments** options are dimmed and are ignored.

Note that the circles have a similar option, however, there is a difference between the two. For the polygons, the size of the sides is always a whole number, which is exactly what is entered in its field. As the size of the radius increases, the number of sides and also their size remain constant until another side of the same size can fit. With the circles, the number of sides remains the same, but the size increases as the radius increases. Polygons generated using this option are illustrated in Figure 4.2.2.5.

![Figure 4.2.2.5: Polygons generated using the Polygon By Size Of Segments option.](image)
The Pattern tab

Pattern: This option offers the ability to draw polygon based variations of stars, daisies, and gears, called patterned polygons. When this option is selected, the sub-options it contains become active and display nine icons which represent the different types of shapes that can be generated (Figure 4.2.2.6). This option can be used with either the By # or the By Size drawing options.

When using the Polygon By # Of Segments option, the variations in the middle and right columns require an even number. If an odd number is entered, a warning is posted and the number is adjusted to the next even number. This is because these patterns use pairs of symmetric shapes to generate the overall shape.

Patterned polygons are created by inputting three points. The first is at its center and the second on its perimeter. The third is also on the perimeter and can be either farther or closer to the center than the second point, each case producing a different shape. After the first click, a polygon is rubber banded. After the second click, the exact shape that will be created is rubber banded. This allows you to accurately inspect the shape that you will be generating and how it is affected by the available options. Two sets of patterned polygons, arranged in one-to-one correspondence with the respective icons, are shown in Figure 4.2.2.7. The set on the left (a) was created by drawing the third point outwards; the set on the right (b) was created by drawing the third point inwards. In all cases, # Of Segments 12 was used. For all other options the defaults were used.

Figure 4.2.2.6: The Polygon Pattern options.

Figure 4.2.2.7: Shapes created with the Pattern option on and by drawing the third point (a) outwards and (b) inwards.
A number of options are available for generating variations of patterned polygons. To describe them we first need to establish a few conventions and terms.

Looking at the blown up icons in Figure 4.2.2.8, they all have a teal (lighter) horizontal line. The parts of the pattern that are above and under the teal line will be referred to as **upper** and **lower**. Note that the left and right patterns only have upper parts. The pattern in the middle has both an upper and a lower part. This is true for all the patterns in the middle column. In these cases the vertical lines of the pattern have a point where they intersect the teal line and they thus consist of two segments, each of which may behave differently, depending on the selected options.

The term **inside** means towards the center of the polygon, while **outside** means away from the center. These terms are relative to the direction we draw and not relative to the pattern itself. That is, if we draw a patterned polygon by placing the third input point beyond the second, the upper part of the pattern is outside. If we draw by placing the third point closer to the center than the second point is, the upper part of the pattern becomes inside.

**Upper First, Lower First**: When drawing a patterned polygon, we can start with a point that lies on the upper or lower end of the pattern. This is controlled by these mutually exclusive options, as illustrated in Figure 4.2.2.9. The patterned polygons on row (a) were generated with the **Upper First** option on. Those on row (b) were generated with **Lower First** on. They were all drawn by placing their third point outwards and they all have **# Of Segments = 12**.

When generating a patterned polygon, the vertical segments of the pattern may remain perpendicular to the horizontal lines or they may be adjusted to pass through the center of the polygon. Such adjustments are useful for avoiding self-intersecting shapes.
**Adjust Inside To Angle, Adjust Outside To Angle:** Selecting one or both of these options causes the vertical segments that are in the respective part of the pattern to be adjusted to angles that converge to the center of the polygon.

These options are illustrated in Figure 4.2.2.10. The patterned polygons on rows (a) and (b) have been drawn by placing the third point outwards. The pattern on (c) is the same with (b) except that its third point was placed inwards. None of the Adjust options is used for column (1), Adjust Inside is used for (2), Adjust Outside is used for (3), and both Adjust Inside and Outside are used for (4).

**# Of Segments:** This option affects the patterns on the last row only. It determines the density (resolution) of the points that the pattern will be generated with. The number entered in its field refers to the resolution of each section of the pattern (and not to the complete pattern). This option is illustrated in Figure 4.2.2.11.

*Figure 4.2.2.10:* Patterened polygons created with different combinations of the Adjust options.

*Figure 4.2.2.11:* Patterned polygons created with # Of Segments (a) 2, (b) 6, and (c) 12.
4.2.3 Drawing circles and ellipses

There are three tools for drawing circles and two tools for drawing ellipses. Their difference is in the way the input points are interpreted. From all of them the Circle/Ellipse Options dialog, shown in Figure 4.2.3.1, can be invoked. It contains two tabs with options for setting the resolution of shapes and for specifying the model type. All of these options are also found in the dialogs of a number of other tools.

The Options tab contains a single group of options:

Model Type: The two mutually exclusive options in this group, Facetted and Smooth, specify the type of model that will be generated for the circle that is drawn. A facetted circle consists of a number of straight segments while a smooth circle consists of a single segment, which is an arc.

The Display Resolution tab contains two methods for specifying the resolution of the circle being drawn: a Simple and a Detailed method. These are equivalent to the options in the Point Density group of the Display Resolution tab of the Options dialogs for the primitives and are as discussed in section 4.1.1.

Note that the options in the Display Resolution tab affect both the facetted and the smooth circles, but in different ways. For the smooth circles they affect the resolution of the shapes displayed on the screen but have no affect on how the smooth circles are stored internally, where they always consist of a single circular segment. In contrast, for the facetted circles, the resolution options affect the number of segments with which a circle is stored internally, which is also the number of segments with which these circles are displayed. Consequently, for the facetted circles, these options have memory implications.

Internally, a facetted circle is represented as a regular polygon and the number of its sides can actually be set to a value small enough that the circle tools are essentially used to generate polygons. The reverse is also true and facetted circles can also be generated by setting a high value for the number of sides in a polygon. How many sides are appropriate for a circle depends on the specific circumstances of the modeling task at hand. In general, to appear sufficiently smooth, small circles require fewer sides than larger circles. Note that objects with more sides use more memory and slow down operations. In general, it is recommended that the number of sides in a facetted circle be kept to the minimum required by the task at hand. This is particularly important when circles are used for the generation of a large number of 3D solids.

Figure 4.2.3.1: (a) The Circle/Ellipse Options dialog with the Options tab open and (b) the Display Resolution tab.
Circle, Center and Radius

This tool uses two points to draw a circle. The first point defines the center, and the second determines the length and the position of the radius of the circle. The shape of the circle is rubber banded after the first point is entered (Figure 4.2.3.2(a)).

Circle, Diameter

This tool also uses two points to define a circle. The first defines the position of a point on the circumference of the circle. The second controls the size and position of a diameter of the circle. The shape of the circle is rubber banded after the first input point (Figure 4.2.3.2(b)).

Circle, 3 Point

This tool requires three input points, all of which are points on the circumference of the circle. The shape of the circle is rubber banded after the second point (Figure 4.2.3.2(c)).

Ellipse, Major and Minor Radius

This tool draws an ellipse from three input points. The first and the second points establish the position and the size of one radius of the ellipse. The first point is a point on the circumference and the second is the center of the ellipse. The third point determines the size of the other radius. The shape of the ellipse is rubber banded after the second point (Figure 4.2.3.3(a)).

Ellipse, Diameter and Radius

This tool also generates an ellipse from three input points. The first two points are points on the circumference, and establish the size and position of one diameter of the ellipse. The third input point controls the size of the second radius, which is the perpendicular bisector of the first diameter line. The shape of the ellipse is rubber banded after the input of the second point (Figure 4.2.3.3(b)).
4.2.4 Drawing points, segments, and vector lines

All polygons, circles, and ellipses discussed in the previous section are exclusively closed and symmetric shapes. The tools discussed in this section can be used to draw single points or segments, or any irregular open shape consisting of straight lines. More tools that draw curves and arcs are covered in subsequent sections. These tools can also be used to draw closed shapes and can be used independently to draw one type of shape at a time, or they can be used in sequence to draw a shape of a mixed type.

Point

With this tool active, clicking on the graphics window creates an object from a single-point. This can be literally a single point object when the Object Type modifier is set to 2D Surface Object. Or, it can be a true object derived from the single point when one of the other modifiers is used. The effects of these modifiers are illustrated in Figure 4.2.4.1.

Note that while single point objects are not “true” objects, they are useful as reference entities to which we can snap to draw lines or solid objects. This type of object is also useful for importing special types of entities, such as point clouds, that result from satellite data.

Segment

With this tool active, clicking twice on the graphics window creates an object from a single segment. The types of objects generated for each of the Object Type modifiers are illustrated in Figure 4.2.4.2.

Note that single segment objects can also be drawn with the Vector Line tool (see below), by double clicking at the second point entered to terminate the drawing. The Segment tool makes this drawing process more direct.
Vector Line

With this tool active, clicking on the graphics window a number of times to enter a sequence of points, generates a vector line. A rubber banded line segment appears after the first point and is anchored on that point. Its second point follows the motion of the cursor until the next mouse click. After each click, a new segment is rubber banded and has its first point anchored to the previous point. The drawing sequence is terminated by a double click or a triple click. When numeric input is entered through the Prompts palette, the input is terminated by typing an “e” or a “c” in the Prompts palette, assuming these default key commands have not been reassigned.

The double click (and the “e” character) produces an open shape, the last point of which is at the position where the mouse was double clicked. The triple click (and the “c” character) produces a closed shape, by connecting the last point entered to the first point. A closed shape is also generated when the position of the cursor coincides with the first point and the mouse is double clicked (Figure 4.2.4.3). Note that the latter method for closing shapes works well only when the Grid Snap switch is on, since it is not easy to position the cursor sufficiently close to the initial point, when snapping is not in use.

The Vector Line Options dialog (Figure 4.2.4.4) can be invoked from the Vector Line tool. It contains the two typical tabs: Options and Display Resolution. While a vector line is by definition a sequence of straight segments and can never be smooth, neither can its resolution be changed, these options are available in its dialog. This is because it is possible to mix vector lines with splines and arcs, when shapes are drawn, as discussed in section 4.2.7. Thus selecting Smooth or changing the resolution will affect the splines and the arcs that may have been included in a continuous line, but not the vector line parts. In contrast, the Allow Intersecting Lines and Allow Collinear Points options, found in the Options tab, affect the vector lines directly. Their effect is discussed in subsection 4.2.2.
**Make Closed:** When this option is selected, the shape that is drawn will be closed. That is, as the shape is drawn, a connection to the starting point from the mouse location is included to make the shape closed. For vector lines, stream lines, and arcs, the connection is a line. For splines, additional control points are added to connect to the starting point. This is the same result that is achieved when triple clicking to close an open shape, except that, with this option on, the drawn shape is rubber banded closed right after the second point is entered.
4.2.5 Drawing curves

`form•Z` includes five tools for drawing splines or curves. These tools will always generate parametric curves, which can later be edited or changed. The Spline/Arc Options dialog, shown in Figure 4.2.5.1, is invoked from all the spline and arc drawing tools and is identical to that invoked from the Vector Line tool. It also contains two tabs: Options and Display Resolution, whose options are as discussed in the previous section.

Figure 4.2.5.1:
The Spline/Arc Options dialog.
Spline, Quadratic Bezier

This tool is used to draw 2nd degree Bezier splines, as shown in Figure 4.2.5.2.

• The first point drawn is the first point of the curve as well as the first point of the first control segment (point 1 in Figure 4.2.5.2(a)).

• The second point is the second point of the first control segment (point 2), which is not a point on the curve itself.

• All subsequent points drawn are points on the curve. They are actually the points at the middle of the control segments, where the curve is tangent to the control line (points 3 through 10).

• The last point is the last point on the curve (point 11), assuming an open curve is drawn by double clicking on the last point. Had we triple clicked, the curve would have been closed.

As you draw, the program internally calculates and stores the control segments it needs to draw a Quadratic Bezier curve through the points you click on. Note that such curves can also be generated as controlled curves (c-curves), but the drawing process is different. For the c-curves we explicitly draw the control points. When we use this tool, we draw points that are actually on the curve. However, the end result can be identical. Figure 4.2.5.3 illustrates the derivation of the same curve by first explicitly drawing the control line (Figure 4.2.5.3(a)), and then using the C-Curve to derive the curve (Figure 4.2.5.3(b) and (c)).

Figure 4.2.5.2: Drawing a curve with the Spline, Quadratic Bezier tool:
(a) drawing sequence and (b) the curve.

Figure 4.2.5.3: Generating a Quadratic Bezier with the C-Curve tool:
(a) the control line, (b) curve in edit, and (c) the final curve.
Spline, Cubic Bezier

This tool is used to draw 3rd degree Bezier splines directly. These types of curves are commonly available in popular illustration programs and the method of drawing them in form•Z is similar to these programs. It is illustrated in Figure 4.2.5.4 and is as follows:

- The first click is on the first point on the curve.
- All subsequent points are entered as pairs of points; the first of the pair is a point of tangency on the curve and the second is the end point of the tangent segment. For example, points 2 and 3. Point 2 establishes a point of tangency on the curve. After clicking on 2 a control segment is rubber banded and follows the motion of the mouse while it extends on both sides of point 2. Clicking on 3 freezes the size and direction of the control segment.
- A double click at the second point of a pair completes the drawing of an open curve (point 11 in Figure 4.2.5.4(a)). A triple click closes the curve.

As with the quadratic Bezier, the cubic Bezier can also be generated using the C-Curve tool, as illustrated in Figure 4.2.5.5. A control line is drawn first (Figure 4.2.5.5(a)) and then the C-Curve tool is used to generate a curve from it. The result is as shown in Figure 4.2.5.5(c). Note that, to be able to generate the same curve with that in Figure 4.2.5.4, the control line’s first point is a double point. Such a point can be generated by first drawing an extra segment at the beginning of the line and then moving the first point on top of the second. This can also be done while in edit mode with the cubic Bezier c-curve (see subsection 4.13.9).
B-Spline, Cubic

This tool is used to draw a 3rd degree B-Spline, as shown in Figure 4.2.5.6. The drawing process is as follows:

- The first point is the first point of the curve, if an open curve is drawn.
- All subsequent points are points on the curve, including the last point, which is the last point on the curve, if the drawing is terminated with a double click to produce an open curve. If the drawing is terminated with a triple click, the curve is closed.

Note that, a major characteristic of this drawing method is that all the points you draw are on the curve. As with the other directly drawn curves, this curve can also be generated as a c-curve, by drawing its control line first and then using the C-Curve tool to generate the final curve. This is illustrated in Figure 4.2.5.7.

Figure 4.2.5.6: Drawing a Cubic B-Spline:
(a) the drawing sequence and control line;
(b) the curve.

Figure 4.2.5.7: Drawing a Cubic B-Spline with the C-Curve tool:
(a) the control lines,
(b) the generation, and (c) the curve.
Sketch Spline, Quadratic Bezier

This tool generates a 2nd degree Bezier spline that can be drawn in two different ways, which can also be mixed:

- When this tool is applied in a click-and-click manner, it generates a Bezier spline by using the line drawn as the control line for the 2nd degree spline. This is illustrated in Figure 4.2.5.8. Note that this curve is the same as that in Figure 4.2.5.2, which was drawn with the Spline, Quadratic Bezier tool. They both use the same control line, which is drawn differently in each case. Here the spline is drawn by drawing the points of the control vector. For the previous spline the line is drawn by drawing the midpoints of the control segments, after the first two points.

- When this tool is applied in a click-and-drag manner, a 2nd degree Bezier spline is generated again, but it is much denser, which gives it a “sketch” quality that resembles free hand drawn lines, as shown in Figure 4.2.5.9.

The two methods of drawing with this tool can be mixed, allowing for even more flexibility in sketching lines.
Stream Line

This is a variation of the vector line drawing procedure. When you drag the mouse as you move it (its button is held down), a “stream” of connected small line segments is generated, following the movement of the cursor. It returns to the regular vector line drawing method when you release the mouse button. The drawing process is terminated by a double click or a triple click, as for the other line drawing methods.

The stream drawing procedure generates a segment each time the mouse moves a preset distance. It pauses when the mouse remains idle. The distance that needs to be covered in order to generate a segment is controlled by the Stream Distance parameter entered in the Streamed Input Options dialog, shown in Figure 4.2.5.10. The dialog can be invoked directly from the Stream Line tool and is identical to the dialog invoked from the spline drawing tools, except for the additional Stream Distance parameter. The default value for this parameter is 2' - 0" (or 0.50 m in metric). The stream line drawing process is illustrated in Figure 4.2.5.11.

The stream line drawing procedure is inherently mouse driven. While it may be simulated through numeric input, it works best when the graphic input method is used. Also note that the stream drawing procedure works better when the snapping switches are off. When used with a snapping switch on, it produces jagged stream lines. While such lines may be desirable under certain circumstances, in general, they contradict the intended functionality of the stream lines, which is to produce smooth and free flowing vector lines.
4.2.6 Drawing arcs

form•Z includes six tools for drawing arcs, which are always generated as parametric objects. They can subsequently be edited to change their parameters and they can be transformed to plain objects using the Convert tool. Whether an arc will be a Facetted or Smooth type of model depends on the respective option selected from the Options tab of the Spline/Arc Options dialog invoked from all the Arc drawing tools. Note that this is the same dialog invoked from all the Spline tools and is shown in Figure 4.2.5.1.

While all the Arc tools generate arcs from three input points, they differ by the order in which the points are entered. For all types, the third point can be entered with a single, double, or triple-click.

The single click completes the generation of one arc and starts another in a manner such that the third point of the previous arc is also the first point of the next arc. Any number of arcs can thus be drawn continuously. A double click completes the arc and terminates the drawing. A triple click closes the arc shape with a straight line that connects its two open ends. These variations are shown in Figure 4.2.6.1. As with other drawing procedures, when entering the points of an arc through numeric input, typing “e” ends the drawing and produces an open arc, and typing “c” closes the arc shape.

Figure 4.2.6.1: Drawing arcs and entering (a) single and then double at the end, (b) double, and (c) triple click at third point.
These tools can be used to draw clockwise and counterclockwise arcs by first clicking at the *initial* point, then the *center*, and then the *end* point of the arc. After the first click a circle is rubber banded and its size is adjusted as the mouse moves. After the second click, which sets the center, an open arc is rubber banded. The third single, double, or triple-click sets the arc size. Examples are shown in Figure 4.2.6.2 (a) and (b).

These tools can also be used with a modifier key to draw an arc by entering its *center* first, its *initial* point second, and its *end* point last. This can be done by pressing *control* (Macintosh) or *ctrl+alt* (Windows) at the time you click for the first point. However, this modification is available only when you start drawing with these tools. It is not available when the arc is drawn in sequence, after another arc or another shape.

Contrary to the previous four tools that always use the center to draw arcs, these tools can be used to draw arcs by inputting points on their perimeter only. The two tools are different by the order in which the perimeter points are entered. For both the *initial* point of the arc is entered first. For the Endpoint-Last Arc, the *middle* point is entered second and the *end* point last. For the Midpoint-Last Arc, the *end* point is entered second and the *middle* point last. After the first click a straight line is rubber banded and its size is adjusted as the mouse moves. After the second click, an open arc is rubber banded. The third click, double-click, or triple-click sets the size of the arc. 3-Point arcs are illustrated in Figure 4.2.6.2 (e) and (f).
4.2.7 Drawing continuous lines

The line drawing procedures can be freely mixed to produce a connected composite line consisting of a mixture of straight line segments, splines, streams, and arcs. This is done by switching line drawing tools as a drawing is in progress. Such shapes will be referred to as **composite curves**. A continuous drawing process is illustrated in Figure 4.2.7.1. It is as follows:

- Select the Vector Line tool ( ), click on 1, 2, and 3.
- While at 3, select the Arc, Clockwise, Endpoint-Last tool ( ), click on 4 and 5.
- At 5, select the Arc, Counterclockwise Endpoint-Last tool ( ), click on 6 and 7.
- At 7, select Vector Line ( ), click on 8, and double click on 9.

Additional examples of continuous line drawings are shown in Figure 4.2.7.2.

The segments of a line drawing, whether open or closed, may or may not be allowed to intersect or cross each other (Figure 4.2.7.3). This depends on whether the **Allow Intersecting Lines** option is selected in the dialogs invoked by the respective tools. By default this option is on, which allows self-intersecting shapes. If the option is off and such shapes are encountered, the program beeps, issues a message, and cancels the drawing. The program may also filter out coincident and colinear points, if **Allow Colinear Points** is on.

*Figure 4.2.7.1: Drawing continuous vector lines.*

*Figure 4.2.7.2: Examples of continuous line drawings.*

*Figure 4.2.7.3: Examples of disallowed line drawings if the **Allow Intersecting Lines** option is off.*
4.2.8 Undoing a step at a time when drawing lines

When using one of the line drawing tools, you can cancel the most recently drawn point using the **one-step undo** operation, which is executed by selecting the **Undo** item (Edit menu) or by pressing its key equivalent **command+Z** (Macintosh) or **ctrl+Z** (Windows). The latter is expected to be used most frequently. When using **Undo**, the rubber banded line follows the motion of the mouse when you go to select it (as when you change tools during drawing). However, it falls back in place after **Undo** has been selected.

When using the **Undo** item, an indication of what will be undone is included with the item. The list of entries in the Prompts palette is not updated when you undo, but rather a complete record of previous entries that were subsequently undone is maintained.

When you draw with the Vector Line tool, each one-step undo cancels the most recently entered point and the anchor of the rubber banded line is moved to the previous point. You can execute any number of one-step-undos sequentially to cancel more than one point. You can not cancel the first point entered, except by cancelling the complete drawing process; by pressing the **esc** key or selecting another tool (Figure 4.2.8.1).

When drawing with one of the Spline tools, each execution of the one-step undo takes you to the previous input point and the curve is regenerated without the cancelled point. Undoing smooth line segments is illustrated in Figure 4.2.8.2.

When you draw with Stream Line, each one-step undo takes you back to the point where the mouse button was pressed. That is, the stream of line segments between the point where you started dragging and the next mouse click is cancelled as a single step.

When you draw with any of the Arc tools, each execution of the one-step undo cancels the complete arc. This is true whether you are about to enter the center or end point of the arc (see Figure 4.2.8.3).

When executing one-step undos, the current drawing tool remains active and does not switch to the tool that was used to draw the portion of the line to which the cancellation has taken you. If another drawing tool is desired, it can be selected in the usual manner.
4.2.9 Editing shapes

Recall that directly generated and derivative objects can be edited with the Edit Controls tool to adjust their shapes (see subsection 4.3.7). When clicking on an editable object with the Edit Controls tool, the object enters into edit mode and its control handles are displayed so that they can be moved. You move a control point by clicking on it and dragging it, then clicking again to place it at a new position. These operations are also discussed in section 4.3.7. Each type of object has different controls, as follows:

Editing single point, single segment objects, vector lines, stream lines, and polygons

These are not parametric objects, but can still be edited to move their points, which effectively changes their shape.

- When Edit Controls is used with 2D surface objects, in edit mode, their points are highlighted with handles and can be moved by dragging them the usual way.

- When used with 2D or 3D enclosures, in addition to above handles, an arrow is displayed and can be dragged to change their wall width and alignment.
  - To change the width, click on the arrow, drag it, and click again to position it.
  - To change the alignment, click on the width arrow while pressing the control key (Macintosh) or the Ctrl + Alt keys (Windows). This toggles the alignment from left to center, from center to right, or from right to left.

- When used with 3D extrusions, 3D convergences, and 3D enclosures, in addition to above controls, an arrow representing the height of the object is displayed. It can be dragged to adjust the height. For the 3D convergences, the base point, where the apex (tip) is set is also displayed and can be dragged. The following operations can also be applied:
  - To change the direction of the extrusion, while dragging the height arrow, press the option key (Macintosh) or the Ctrl + Shift keys (Windows). The motion is parallel to the reference plane.
  - To reset the extrusion to a direction perpendicular to the source plane, click on the height arrow while pressing the control key (Macintosh) or the Ctrl + Alt keys (Windows).
  - To reset the position of the base point of a 3D convergence to the centroid of the base shape (if it was previously moved), click on the base point while pressing the control key (Macintosh) or the Ctrl + Alt keys (Windows).
**Editing splines and arcs**

Clicking on a spline, arc, circle, or ellipse with the Edit Controls tool invokes the *Edit Controls Options* palette (Figure 4.2.9.2), places the entity you clicked on in edit mode, and displays its controls, which vary according to the type of the entity. The *Edit Controls Options* palette consists of three parts, which are discussed in more detail in section 4.3.7. The palette includes icons for selecting the editing operation to be applied and the Move icon is on by default allowing you to start moving control points as soon as you enter the edit mode. The controls displayed and that can be moved per each type of entity are as follows:

**Quadratic Bezier spline**: The points that were drawn to generate the shape are displayed as control points. They can be freely moved to reshape the curve, as shown in Figure 4.2.9.3. As the control points are moved the curve is regenerated.

**Cubic Bezier Spline**: For this curve, two types of control points are displayed, as shown in Figure 4.2.9.4:

- **Knots or pivot** points: These are the points at which the curve touches the control lines. They are displayed in a blue color. Moving a knot causes the respective portion of the curve to move, as it passes through the new position of the knot.

- **Control** points: These are the points of the control line; that is, points at the ends of control segments, but not points on the spline. They are displayed in green. Moving a control point changes the slope of the curve at that area.

**Cubic B-Spline**: As for the Quadratic Bezier, the points that were drawn to generate the spline are displayed as control points. When they are moved the spline is regenerated. An example is shown in Figure 4.2.9.5.

**Sketch spline**: The points that were drawn to generate the sketch spline are shown as control points (Figure 4.2.9.6). These points do not lay on the curve itself. Moving a control point changes the shape of the curve in the area of the control point.
**Arens**: Each arc is represented by three control points and three arrows. There are two point controls at the ends of the arc and a size control at the position of its center. There are two arrows at its ends and one more at the middle of the arc.

- Moving the start or end control point changes the length of the arc cord (Figure 4.2.9.7(a)).
- Moving the center, which is restricted to the direction of the bisector of the arc angle, changes the size of the radius and the angle of the arc, but leaves the start and end points unchanged (Figure 4.2.9.7(b)).
- Clicking on and moving the size arrow changes the radius of the arc (Figure 4.2.9.7(c)).
- Clicking on and moving one of the arrows at the ends of the arc changes the start or end angle of the arc (Figure 4.2.9.7(d)).

For all types of curves, an **insert point** operation is available when in edit mode. To insert a point, click on the curve, after selecting the Insert Point (getter) icon from the **Edit Controls Options** palette. Similarly, pivot and control points can be deleted from cubic Bezier, NURBS, and tangent splines, by selecting the Delete (getter) icon from the palette. These operations are discussed in more detail in section 4.3.7.

If a parametric shape is drawn using a combination of the Vector Line, Arc, or any of the Spline tools, it can also be subsequently edited using the Edit Controls tool. Each section of such shapes will display different controls and will behave differently when edited. An example is shown in Figure 4.2.9.8.

When splines, arcs, or composite shapes are used as source shapes to generate 2D enclosures, the enclosure parameters (wall width and alignment) are included and are displayed as for the enclosures from vector lines (Figure 4.2.9.1(a)). Similarly, when these source shapes are used to generate 3D extrusions, convergences, or enclosures, the height parameter is included and is displayed in edit mode, as for the 3D objects generated from vector lines (Figure 4.2.9.1(b), (c), and (d)).
All the parametric curves (splines, arcs, and composite curves) and 3D objects derived from them can also be edited using the Query tool. Clicking on such an object with the Query tool invokes a Query dialog that displays information about the object. When the object is parametric, an Edit... button also appears active in the Query dialog. Clicking on it invokes an Edit dialog for the respective type of object. For example, the Composite Curve Edit dialog is shown in Figure 4.2.9.9. The Extrusion/Convergence Edit and 3D Enclosure Edit dialogs are shown in Figure 4.2.9.10(a) and (b). The Edit dialogs allow you to change the parameters of the respective object. These procedures are discussed in more detail in section 4.22.2.

**Figure 4.2.9.9:** The Composite Curve Edit dialog.

**Figure 4.2.9.10:** The (a) Extrusion/Convergence Edit and (b) 3D Enclosure Edit dialogs.
4.3 Picking and editing

All the operators (except for the drawing tools) require you to pick or select some entity so that they can act on it. That entity may be a part of an object, a complete object, or a group of objects, depending on the task at hand. What entity you select is determined by the currently active topological level that you set in the fourth left row of the tool bar. The following topological level modifiers are discussed in this section:

- Point
- Segment
- Outline
- Face
- Object
- Group
- Hole/Volume
- Automatic

You can pick with the Pick tool that is dedicated to picking, or with tools that execute operations. The latter tools are discussed in respective sections. The Pick tool, as well as four editing tools (Edit Controls, Transform Object Axes, Edit Surface, and Edit in Parameter Space) are discussed in this section.

The act of picking involves the identification of an entity by clicking the mouse on, or sufficiently close to it. Some entities may require more than one click in order to be uniquely identified. Also available are two variations of an area pick method by which all the entities that are within or cross the boundaries of a rectangular frame or a free form lasso shape are selected (see subsection 4.3.3). For coincident entities (entities that are on top of each other), the pick parade method can be used to select them sequentially, one at a time (see 4.3.3). Selected entities are highlighted with color. The default highlight color is red. It can be changed in the Project Colors dialog, which can be invoked from the Options menu (see subsection 3.7.1).
4.3.1 The topological levels

Picking works in conjunction with the Topological Level modifiers. The active topological level determines the type of entity that will be picked (point, segment, outline, face, object, group of objects, or hole/volume). The topological levels affect both picking with the Pick tool (typically associated with the prepick selection method, see subsection 4.3.4) and picking with any other tool, when executing an operation.

**Point**

This sets the current topological level to Point. A point is picked by clicking the mouse on, or sufficiently close to it. The picked point is highlighted with a cross mark (Figure 4.3.1.1).

**Segment**

This sets the current topological level to Segment. A segment is picked by clicking anywhere on it or close to it (Figure 4.3.1.2).

**Outline**

This sets the current topological level to Outline. Either two mouse clicks on unique segments, or a single click inside the boundary of the outline can be used to pick an outline, depending on the option selected in the Pick Options dialog (see next section). When picked, the entire outline is highlighted, as shown in Figure 4.3.1.3.

**Face**

This sets the topological level to Face. Two clicks on unique segments or one click inside the boundary of the face can be used to pick a face, depending on the option set in the Pick Options dialog (see next section). Picked faces are shown in Figure 4.3.1.4.

---

**Figure 4.3.1.1:** Picking a point: click on point p while the topological level is at Point.

**Figure 4.3.1.2:** Picking a segment: click on point p while the topological level is at Segment.

**Figure 4.3.1.3:** Picking an outline using the (a) two point method: click on 1, then 2; (b) one point method: click on 3, while the topological level is at Outline.

**Figure 4.3.1.4:** Picking a face using the (a) two point method: click on 1, then 2; (b) one point method: click on 3, while the topological level is at Face.
**Object**

This sets the current topological level to Object. An object is picked by clicking on one of its points, one of its segments, its origin, or its centroid, when the latter two are displayed. When picked, the entire object is highlighted (Figure 4.3.1.5). Objects can also be picked from the Objects palette by clicking in the selection column in front of their names.

When the topological level is set to object, pressing the option (Macintosh) or ctrl+shift (Windows) keys while clicking to select the object, selects all objects in a clone family if the selected object is a member of such a family. Cloned objects are discussed in section 4.23.5.

**Group**

This sets the current topological level to Group. A group is picked by selecting one of its objects. When a group is picked, all its objects are highlighted (Figure 4.3.1.6). Groups can also be picked from the Objects palette by clicking in the selection column in front of their names.

**Hole/Volume**

This sets the current topological level to Hole/Volume. A hole or a volume consists of a number of adjacent outlines that form a closed group. At least one of these outlines must be a hole to a face (negative outline). The hole/volume is picked through one of its negative outlines, with one click of the mouse. Depending on which pick method is on in the Pick Options dialog, the mouse is clicked on a segment or inside the boundary of the negative outline.

The recognition of a group of outlines as a hole/volume depends solely on its topological structure, and not on how it was generated. A hole typically penetrates a solid all the way through, and has at least two outlines that are holes of faces. A volume typically contains a single outline that is a hole of a face. A volume can be negative (cavity) or positive (bump). Examples of proper and improper holes and volumes and where they should be selected are shown in Figure 4.3.1.7.

(a) Block with a hole through it; to pick it by the edge method, click on 1.
(b) Block with a positive volume; to pick it by the edge method, click on 2.
(c) Block with a negative volume (cavity); to pick it using the inside boundary method, click on 3.
(d) The notches in these blocks are neither holes nor volumes and can not be picked as such.
Automatic

This modifier does not set a specific topological level, but rather determines the intended topological level by analyzing where the mouse is clicked. The position of the mouse relative to the object parts determines what topological level is intended to be selected.

- If the mouse is close to a point, segment, or on a face, the object is selected (ﷺ).
- Pressing the `option` key (Macintosh) or `ctrl+shift` (Windows) while the mouse is close to a point, that point is selected (ﷺ).
- Pressing the `option` key (Macintosh) or `ctrl+shift` (Windows) while the mouse is close to a segment (without being too close to an end point) that segment is selected (ﷺ).
- Pressing the `option` key (Macintosh) or `ctrl+shift` (Windows) while the mouse is on the surface of a face (without being too close to a segment or a point) that face is selected (ﷺ).
- Pressing the `control` key (Macintosh) or `ctrl+alt` (Windows) while clicking on any part of an object selects the entire group, whenever the object is grouped with other objects (ﷺ).
- If an object’s centroid or origin are displayed, and the mouse is closer to these than a point, segment, or face, the entire object is picked (ﷺ).

The Automatic topological level can be used with both the Pick tool or with any other tool. When used, the cursor changes to a different shape representing the topological level it detects. This serves as a preview to indicate what level will be picked when the mouse is clicked. These cursor shapes are shown in Figures 4.3.1.8 and 4.3.2.2. The same shapes are used with the Pick tool when the **Show Pick Cursor** option is selected in the **Pick Options** dialog (see next section).
4.3.2 The Pick tool and the Pick Options dialog

Pick

When this tool is selected and the mouse is clicked on an entity, the system processes the one or two input points that were entered and identifies the entity intended for selection. The type of entity it searches for is determined by the active topological level. If successful, it highlights the entity and adds it to a list of selected entities. If it fails, it cancels the pick operation, effectively clearing any entities that may have already been selected. When cancelling the pick, it may also beep, if the respective option in the Pick Options dialog is on (see below).

Whenever previously selected entities are accidentally deselected by failing to click the mouse close enough to a pickable entity, the Select Previous item in the Edit menu can be used to reselect the previously selected entities.

In form•Z you can select more than one entity at a time without being required to press the shift key, as other applications do. However, the option Use Shift Key For Multiple Pick is also available in the Pick Options dialog (see below), should this be a habit of yours.

While failing to pick an entity may be due to a user error, clicking away from a pickable entity is also a proper method for purposely clearing all picks. This is equivalent with selecting the Deselect item in the Edit menu. Individual entities can be deselected by picking them again, while they are picked.

Only active unghosted objects on unlocked layers (and their parts) can be selected using the Pick tool. The only ways to pick ghosted objects is by selecting them globally through the use of the Select All Ghosted item or by using the Select By... item, both in the Edit menu.

As already mentioned, a number of options that affect picking are selected from the Pick Options dialog (Figure 4.3.2.1), which can be invoked directly from all the topological level tools and the Pick tool.

Figure 4.3.2.1: The Pick Options dialog.
**Pick Outlines/Faces/Holes By Clicking:** The two check boxes in this pair of options offer choices each of which may be on/off independently or both can be on at the same time. They determine the picking method to be used for three of the topological levels, as follows:

**On Edges:** When this option is on (which is the default), faces and outlines are selected by clicking on a unique pair of their segments. Holes/volumes are picked by clicking on a single segment of an outline that is a hole of a face.

**Inside Boundaries:** When this option is on, one click inside the boundary of the respective entity suffices for selecting outlines, faces, and holes/volumes. While this method is convenient, it can directly select only front facing entities.

**Pick Objects/Groups By Clicking:** These two check box based options work as the previous for selecting complete objects, as follows:

**On Edges:** When this option is on (default), objects are selected by clicking on a segment.

**Inside Face Boundaries:** When this option is on, one click inside the boundary of any face selects the object. Note that, as for faces, outlines, and holes, both of these options can be on at the same time.

**Beep When Deselecting:** Clicking the mouse away from any selectable entities deselects all the selections. When this happens and this option is on (default), a warning beep also sounds. If you turn this option off, the beep will not sound.

**Click And Drag:** When this option is selected, the Pick tool can be used to execute geometric transformations directly. This is discussed in detail in section 4.3.5.

**Return To Pick Tool:** When this option is selected, form-Z will automatically activate the Pick tool, as soon as an operation, which used another tool for its execution, is completed. If this option is off, you have to activate the Pick tool (by clicking on it) every time you need it.

**Control+Click For Last Tool** (Macintosh) or **Ctrl+Alt+Click For Last Tool** (Windows): This option is only available when **Return To Pick Tool** is on. When selected while the Pick tool is active, pressing `control` (Macintosh) or `ctrl+alt` (Windows) at the time you click on the window selects and executes the most recently used tool (other than the Pick tool).

**Use Shift Key For Multiple Pick:** When this option is selected, to pick more than one entity, you need to press the shift key when you click to select entities beyond the first one. When this option is off, you select multiple entities by simply clicking on them.

**Auto Select New Objects:** When this option is on, all new objects will become selected and highlighted as soon as they are created.
**Pick Facets Of Controlled Objects:** This option determines if the components of the facetted representation of parametric objects (points, segments, outlines, faces, and holes) can be selected. When this option is off, they cannot be selected. When it is on, they can.

Recall that the facetted representation of all controlled objects is generated from the control parameters at creation time and whenever a control parameter is changed. If a change is made in the facetted representation, such as moving a point or changing the color of a face, the change will be lost the next time the facetted representation is regenerated from the control parameters. It is recommended that, when you want to edit the facetted representation, you drop the controls from the object using the Convert tool (see section 4.21.9).

**Show Pick Cursor:** When this option is on, the cursor changes to a different icon when placed on an entity that can be picked by the current topological level. These icons are shown in Figure 4.3.2.2. This serves as a preview that a click will pick the desired entity. This feature is always used for the Automatic topological level (see previous section).

**Use Nudge Keys:** When this option is on, you can use the arrow keys to move, rotate, or scale the selected object. These operations are discussed in detail in section 4.3.6.

**Area Pick:** These two options determine whether a rectangular **Frame** (default) or a **Lasso** shape is drawn for area picking. This method is discussed in the next section.

**Pick Crossing:** This option of the **Area Pick** offers the ability to select objects that intersect the shape (**Frame** or **Lasso**) drawn by the area pick. When it is on, all entities that are either inside or overlap the pick area are selected. This option is off by default.
4.3.3 Area picking and pick parade

While the main picking method is to click on an entity, additional methods are available for selecting in special situations, such as picking many entities with a single operation or picking entities when they may be behind or coincident with other entities.

Area picking

More than one entity at the same topological level can be selected simultaneously by using one of the two available area picking methods in the Area Pick group of options, in the Pick Options dialog:

**Frame:** When this method is selected, which is the default, entities are picked by drawing a rectangle, as follows: Position the mouse away from any pickable entity and press the button. This sets one of the corner points of a rectangular frame. While holding the button down, drag the mouse to another point. As the mouse is moved a rectangular shape is rubber banded. When the proper size of a rectangle has been drawn, release the button to define the other corner of the rectangle. This is illustrated in Figure 4.3.3.1.

**Lasso:** With this method on, position the mouse away from any pickable entity, press the button, and while holding it down, move the mouse to trace a path around the entities you wish to select. When the area you need has been outlined, release the mouse. For best results, return close to the first point before releasing the mouse. The system needs a closed shape to be able to identify the entities that are within it and it will automatically close the lasso shape you draw by connecting its two endpoints. The pick by lasso process is illustrated in Figure 4.3.3.2.

**Pick Crossing:** When this option is off, the shape used for picking (rectangle or lasso) is drawn around the entities that need to be picked, and for an entity to be picked it must be completely contained in the frame. When this option is on, also entities that cross the boundaries of the frame will be picked. This option is off by default.

Area picking is only available when using the Pick tool for prepicking. It is not available when entities are picked by other tools using the postpick method. When selecting a group of entities using the area picking method, all previously selected entities will be deselected. To select more entities as well as maintain previously selected entities, press the *shift* key.
Pick parade

In some instances, multiple entities may overlap or lay directly on top of each other on the screen. In these cases the system is unable to recognize which entity is intended for selection by the user and will select the first one it finds. To resolve this conflict FormZ provides the **pick parade** facility.

The pick parade sequentially highlights the possible selections for a given input point until the desired entity is selected. The pick parade can be used as follows:

- Press the **shift** key at the time you click on the entity you wish to select.

The first entity found is highlighted in the highlight color, which is **red** by default. For point and segment selections, the face to which the point or segment belongs is also highlighted in **green** and the whole object is highlighted in **black**.

- You continue clicking, while you continue pressing the **shift** key. Where you click after the first click is insignificant. The program remembers and reuses the first click for identifying additional entities.

With each additional click a new entity is highlighted, and so is its face, if picking point or segment, and/or the whole object to which the entity belongs.

- When the desired entity is highlighted, release the shift key and click the mouse one last time. This clears all the other highlights, except for the entity you are selecting, which remains highlighted and awaits the application of an operation, assuming prepick selection mode is being used. The pick parade can be used with both the prepick and postpick methods.

Use of the pick parade for selecting from overlapping segments is illustrated in Figure 4.3.3.3.

When the **Clicking Inside Boundaries** option is selected in the **Pick Options** dialog for picking outlines, faces, and holes, the pick parade will visit both front and back facing entities. This is an exception to the general rule that the one point selection option for outlines, faces, and holes only picks front facing entities.

![Diagram](image.png)

**Figure 4.3.3.3:** Parading to pick a segment:
(a) Four cubes with coincident segments.
(b) First shift-click selects segment of cube **c1**;
(c), (d), (e) subsequent shift-clicks select segments from cubes **c2**, **c3**, and **c4**, respectively.
(f) A last plain click leaves segment of **c4** picked.
4.3.4 Prepicking and postpicking

form•Z offers a choice between a prepick and a postpick method for the selection of the entities involved in an operation. For the prepick method, the entities are picked first and then the operator itself is activated. In the postpick method, the entities are picked after the operator has been activated. The term entity here and throughout this chapter is used to mean a point, segment, outline, face, object, group, or hole/volume. Recall that most of the form•Z tools can be applied to any of these topological levels.

Which of the two picking methods is preferable may depend on the task at hand or the habits of the user. In form•Z the two methods may be freely mixed. In general, the prepick method offers the possibility to select any number of entities and apply the same operation to all of them simultaneously. These entities may actually be at different topological levels. However, certain restrictions may be imposed by the semantics of specific tools. Prepicked entities remain selected and highlighted after the completion of an operation. This feature makes it possible to immediately apply another operation to the same set of prepicked entities. Any number of operators can thus be applied sequentially to the same set of prepicked entities.

The postpick method frequently offers an immediacy that is more responsive to the intuitive notions associated with certain tools and tasks. The postpick method causes an operation to be executed as soon as the required entities have been picked. The number of selections required depends on the specific tool.

Another significant distinction between the prepick and the postpick methods is that the first relies on the use of the Pick tool for selecting the required entities. With the postpick method, picking is embedded within each tool. That is, after the selection of a tool, the mouse is clicked on the entity to which the operation is being applied.

Global picking is also supported by items in the Edit menu: Select All Unghosted selects and highlights all the active objects; Select All Ghosted does likewise for the inactive ghosted objects. They are both applied at the object level, regardless of the currently active topological level. Select By... also allows you to pick items according to certain criteria, which are set in the Selection Criteria: Modeling or Drafting dialogs. These criteria are discussed in detail in section 3.2.4. Global selection of objects through the menu items can only be used for prepicking objects before the execution of an operation. It cannot be used in a postpick manner. The Select All Ghosted item is the only method available for selecting ghosted objects. However, after the ghosted objects have been selected they can be operated as any active object.
4.3.5 Transforming with the Pick tool

The Pick tool can be used to select entities at different topological levels. It can also be used to move, rotate, and resize objects, when the **Click And Drag** option in the **Pick Options** dialog is selected. While objects are simply picked when you quickly click on them, you can geometrically transform them by clicking and dragging. You can also use the Pick tool to transform parts of objects, by selecting a topological level other than Object. Geometric transformations with the Pick tool are executed as follows:

**Move:**

Click and drag the entity. As you drag, the selected entity is rubber banded and it takes its new position as soon as you release the mouse. If the Perpendicular switch is on when moving, the entity moves perpendicular to the reference plane. It moves parallel to the reference plane, otherwise.

Pressing the **shift** key while moving constrains the movement to a direction parallel to the horizontal or vertical axis, whichever is the direction relative to which to movement is the greatest.

**Rotate relative to reference plane:**

Click and drag while pressing the **control** key (Macintosh) or **ctrl+alt** (Windows). The entity is rotated about the axis that is perpendicular to the reference plane and passes through the centroid of the object.

Pressing the **shift** key while rotating, the rotation is executed at 45° increments.

**Rotate relative to screen plane:**

Click and drag while pressing the **command** (Macintosh) or **ctrl** (Windows) key. The entity is rotated about the axis that is perpendicular to the screen plane and passes through the centroid of the object.

Pressing the **shift** key while rotating executes the rotation about the horizontal axis of the screen, when the mouse moves vertically, or about the vertical axis, when the mouse moves horizontally.

**Scale:**

Click and drag while pressing the **option** key (Macintosh) or **ctrl+shift** (Windows). The entity is uniformly scaled relative to the centroid of the object.

Pressing the **shift** key while scaling applies the scaling at 5% increments.
4.3.6 Using the nudge keys

The *up*, *down*, *left*, and *right arrow* keys can be used as *nudge* keys, which is to execute moves, rotations, and scalings, typically by small increments. The nudge keys become active when the **Use Nudge Keys** group of options is selected in the **Pick Options** dialog, as shown in Figure 4.3.6.1.

When the arrow keys are pressed by themselves, a move is executed. When modifier keys are also pressed, the arrow keys can be used to rotate or resize (scale) the selected object or objects. The default key assignments to the nudge operations are listed in the table of Figure 4.3.6.2. Recall that any of these assignments can be customized by the user, using the **Key Shortcuts Manager** dialog, as discussed in section 3.2.7. Typically, the nudge keys execute geometric transformations by small increments. These are set in the following numeric fields:

- **Translate By:** The value entered in this field determines the increment that will be used to move objects with the nudge keys.

- **Rotate By:** The value entered in this field, expressed in degrees, determines the increment that will be used to rotate objects with the nudge keys. Objects are always rotated about an axis that passes through their centroids and is perpendicular to the currently active reference plane.

- **Scale By:** The value entered in this field, which represents a scaling factor, determines the increment that will be used to resize objects with the nudge keys. Objects are always scaled relative to the centroid (center of gravity) of their bounding box. Default value is 0.950, which represents a 95% scaling factor when the left or down arrows are used. It represents a 105% scaling factor when the up and right arrow keys are used.

- **Relative To Reference Plane:** When this option is selected, the nudge transformations are executed relative to the reference plane. If this option is off, they are executed relative to the world coordinates.

- **Apply To Each Selected Entity Individually:** This option affects how entities are transformed, when more than one is selected. When only one entity is picked, it has no effect. When this option is on, each selected entity is rotated or scaled about the individual centroid of the object it is on. When this option is off, all selected entities are rotated or scaled about the centroid of the group of objects. This option has no effect when moves (translations) are applied.
Move:

<table>
<thead>
<tr>
<th>Macintosh</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>left arrow</td>
<td>left arrow</td>
</tr>
<tr>
<td>Moves left along horizontal axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>right arrow</td>
<td>right arrow</td>
</tr>
<tr>
<td>Moves right along horizontal axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>up arrow</td>
<td>up arrow</td>
</tr>
<tr>
<td>Moves up along vertical axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>down arrow</td>
<td>down arrow</td>
</tr>
<tr>
<td>Moves down along vertical axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>shift + up arrow</td>
<td>shift + up arrow</td>
</tr>
<tr>
<td>Moves up along perpendicular axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>shift + down arrow</td>
<td>shift + down arrow</td>
</tr>
<tr>
<td>Moves down along perpendicular axis of reference plane.</td>
<td></td>
</tr>
</tbody>
</table>

Rotate:

<table>
<thead>
<tr>
<th>Macintosh</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>control + left arrow</td>
<td>ctrl + alt + left arrow</td>
</tr>
<tr>
<td>Rotates counterclockwise about perpendicular axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>control + right arrow</td>
<td>ctrl + alt + right arrow</td>
</tr>
<tr>
<td>Rotates clockwise about perpendicular axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>control + up arrow</td>
<td>ctrl + alt + up arrow</td>
</tr>
<tr>
<td>Rotates clockwise about horizontal axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>control + down arrow</td>
<td>ctrl + alt + down arrow</td>
</tr>
<tr>
<td>Rotates counterclockwise about horizontal axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>shift + control + up arrow</td>
<td>shift + ctrl + alt + up arrow</td>
</tr>
<tr>
<td>Rotates counterclockwise about vertical axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>shift + control + down arrow</td>
<td>shift + ctrl + alt + down arrow</td>
</tr>
<tr>
<td>Rotates clockwise about vertical axis of reference plane.</td>
<td></td>
</tr>
</tbody>
</table>

Independent Scale:

<table>
<thead>
<tr>
<th>Macintosh</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>command + left arrow</td>
<td>ctrl + left arrow</td>
</tr>
<tr>
<td>Decreases scale along horizontal axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>command + right arrow</td>
<td>ctrl + right arrow</td>
</tr>
<tr>
<td>Increases scale along horizontal axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>command + up arrow</td>
<td>ctrl + up arrow</td>
</tr>
<tr>
<td>Increases scale along vertical axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>command + down arrow</td>
<td>ctrl + down arrow</td>
</tr>
<tr>
<td>Decreases scale along vertical axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>shift + command + up arrow</td>
<td>shift + ctrl + up arrow</td>
</tr>
<tr>
<td>Increases scale along perpendicular axis of reference plane.</td>
<td></td>
</tr>
<tr>
<td>shift + command + down arrow</td>
<td>shift + ctrl + down arrow</td>
</tr>
<tr>
<td>Decreases scale along perpendicular axis of reference plane.</td>
<td></td>
</tr>
</tbody>
</table>

Uniform Scale:

<table>
<thead>
<tr>
<th>Macintosh</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>option + left arrow</td>
<td>ctrl + shft + left arrow</td>
</tr>
<tr>
<td>Decreases scale uniformly (all directions).</td>
<td></td>
</tr>
<tr>
<td>option + right arrow</td>
<td>ctrl + shft + right arrow</td>
</tr>
<tr>
<td>Increases scale uniformly (all directions).</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.3.6.2: The default nudge key assignments for Macintosh and Windows.
4.3.7 Edit tools

Next to the Pick tool, there are three tools for picking and editing parametric entities after their initial generation, Edit Controls, Edit Surface, and Edit in Parameter Space. As their names represent, the first is used to transform the controls of an object, while the second transforms the parametric surface directly. They both apply to the following types of objects:

- **Analytic primitives**: These are the directly generated primitive objects, namely, cube, cone, cylinder, sphere, and torus. Their editing operations are discussed in subsection 4.1.10.

- **Derivative nurbs** and **nurbs from converted primitives**: The former are the derivative objects that can also be generated as nurbs, which includes objects of revolution, surface/solid helixes, sweeps, and skins. The latter are objects initially generated as analytic primitives and subsequently converted to nurbs. The editing of nurbs objects is discussed in subsection 4.14.2.

- **Nurbs from control lines**: These are surface or solid objects that are generated from a set of open or closed control lines. Their editing operations are discussed in subsection 4.14.4.

- **Patch meshes**: These are objects generated from other common objects or by converting analytic primitives. Their editing operations are discussed in subsection 4.15.4.

- **Composite curves**: These are 2D surface objects drawn with any combination of the vector line, spline, or arc drawing tools. These editing operations are discussed in subsection 4.2.9.

The third editing tool applies specifically to **curves** drawn on **nurbs** surfaces and to **trimmed nurbs**.
Edit Controls

This tool is used to edit the controls of parametric objects. These can be edited when displayed in **Wire Frame** or **Interactive Shaded** mode. They can be edited one at a time, or more than one object of the same type may be edited simultaneously. Both the prepick and postpick selection method can be used with this tool. Postpicking is generally preferable when a single object is edited. Prepicking is necessary when more than one object is edited at the same time.

To edit a single object, with the Edit Controls tool active click on the object. To edit more than one object, use the Pick tool to preselect them. Then, with the Edit Controls tool click anywhere in the graphics window. The selected objects are placed in edit mode and their edit controls become visible. These are graphic representations of a parametric object’s controls, which vary by object type and are discussed in the respective sections. However, manipulation of these controls is similar for all the object types.

When you click on an object with this tool and it is placed in edit mode, the **Edit Controls Options** palette, shown in Figure 4.3.7.1, is also invoked. It consists of up to three parts, but some parts are not available for some types of objects. This palette remains open during the edit process and closes as soon as you exit the edit mode. Options selected from the palette determine the what editing operation is applied.

**Preview Density:** When this option is on, which is the default, the resolution of the facets of the parametric object is changed to a less dense level, when the object is edited. This is intended to facilitate the dynamic editing of such objects.

The density of the facets is set using the slider bar of this option. Moving the slider bar to the left results in a lower facet density and faster redraw speed. When this option is off, the original facet density of the object is used, which is initially set in the **Display Resolution** tab and can be changed with the Query Attribute tool.

**Action:** This section contains icons that represent editing operations. For example, the palette in Figure 4.3.7.1 contains a Move (which is always the default), an Insert Point, and a Delete Point operation. Which operations are available varies according to the type of object being edited.

**Selection:** This group contains the **Point Type** pop up menu, which allows you to change the type of pivot points in Cubic Bezier Splines. For all other types of objects this menu is either dimmed or does not show up.
Transforming controls

Two methods are available for transforming controls: a click-and-drag and a click-click method. Each has advantages and disadvantages, depending on the circumstances of the editing operations:

Click-and-drag method: With this method you click the mouse on the control you wish to transform and, while you keep the mouse button pressed, you drag it to apply the desired transformation. The transformation is completed as soon as you release the mouse.

Click-click method: With this method, one or more controls are selected first and then transformed. This method involves the following separate operations:

- **Select a control:** Place the cursor on the control and click. A picked control is shown highlighted, which is red by default.

- **Select multiple controls:** Click on the controls while holding the shift key down. Or use area pick, which is click and drag to draw a rectangle that contains the controls you wish to select.

- **Transform a picked control:** Click on a selected control and move the mouse in the direction you want to transform it. All the selected controls (on the same or multiple objects) follow the same transformation. This is most commonly a move.

- **Unpick selected controls:** Click the mouse on an area of the screen with no controls. Previous picks are also cleared when you click to select another entity, without pressing shift.

- **Exit the edit mode:** Press esc (escape key) or click on the screen away from the objects and any selectable controls, when more are selected.

The main advantage of the click-and-drag method is its directness. The main advantage of the click-click method is that it allows several controls to be transformed simultaneously. It also allows other operations, such as zooming, panning, and turning snapping on/off to be executed while in the process of transforming a control.

The Edit Controls tool can also be used to graphically edit parameters of individual faces of smooth solid objects. The position of a face relative to the rest of the object can be graphically moved and rotated. Once a face is edited, its intersections with the other faces of the object are calculated and the integrity and solidity of the object is maintained.

Recall that a smooth face can be one of six types: plane, cylinder, cone, sphere, torus, or spline. When editing a smooth face, the controls displayed are the same as those of the corresponding surface type, plus an additional square handle and arrows that allow you to move and rotate the surface, respectively. For example, the controls of a cylindrical surface are the same as the controls of a cylinder, as shown in Figure 4.3.7.2.

![Figure 4.3.7.2: Controls of a cylindrical face.](image-url)
When a smooth face is only a portion of the complete surface, such as only a part of a cylinder, the complete surface is shown during editing. The rotation controls consist of three axis lines with arrows, each of which rotates in a different direction. To edit a smooth face, set the topological level to Face and with the Edit Controls tool active select the face, which is highlighted in red and its controls are displayed, as shown in Figure 4.3.7.2.

The Edit Controls tool can also be applied in a prepick manner, to simultaneously manipulate more than one point. To do so, use the Pick tool and, with topological level at Face, preselect any number of smooth surfaces. Then, with the Edit Controls tool active, click anywhere in the project window to start the edit mode. Continue clicking to select the controls you need to manipulate. After the first selection, press shift as you click on additional controls. Then click-and-drag or click-click to move one control. As it moves the other highlighted controls also move. The prepick mode can only be used to edit similar types of smooth surfaces simultaneously. There are also a few restrictions that affect both the prepick and the postpick modes.

The operation will succeed in all cases, as long as it can be executed and does not disturb the integrity of the object. For example, scaling a cylindrical hole in the middle of a cube will work, even when it results in an intersection with one or more sides of the cube. This is illustrated in Figure 4.3.7.3(a). However, re-editing and scaling the hole down or moving an intersecting hole to eliminate its intersection with one of the sides of the cube will not work. This is simply because the program does not have enough information to be able to restructure the cube. This is illustrated in Figure 4.3.7.3(b).

A hole that was previously made to intersect a side of an object (such as a cube) can be moved freely along the same side but not to another side (Figure 4.3.7.3(c)). Note, however, how an intersecting hole can move all around the perimeter of a cylinder (Figure 4.3.7.3(d)). This is because the curved boundary of a cylinder is a single face.

Unlike all the other face types, the surface of a spline face does not extend infinitely. Consequently, if a spline face is edited and its ends are moved to a position such that the spline surface no longer intersects its adjacent faces, the operation fails. This is illustrated in Figure 4.3.7.4.
Inserting and deleting controls of splines

Using the Edit Controls tool, control points can be inserted to these types of spline: quadratic Bezier, cubic Bezier, cubic B-spline, sketch spline (quadratic B-spline), arc, circle, ellipse, nurbz curve, and tangent curve.

Control points can be deleted from cubic Bezier splines, nurbz, and tangent curves.

To insert a point, from the Edit Controls Options palette select the Insert ( ) icon and click on the curve where the control point will be inserted. The new point is inserted and displayed immediately.

To delete a control point, from the Edit Controls Options palette select the Delete ( ) icon and click on the control point that will be deleted. The spline is restructured without the deleted point and is redisplayed.

Changing the type of a pivot point in cubic Bezier splines

Recall that there are three types of pivot points: smooth, non smooth, and corner points. They can be switched from one type to another using either a graphic method or the Point Type menu in the Edit Controls Options palette.

To grahically change the type of a pivot point, while in edit mode and the Move icon active, press ctrl (Macintosh) or ctrl+alt (Windows) when you click or drag, as follows:

- to change a pivot point to a corner click on the pivot point. The shape of the curve will be changed to a sharpe corner at the pivot point. The corner shape is achieved by moving the adjacent control points on top of the pivot point.

- to change a corner to a smooth point, drag the pivot point. This will move the adjacent control npoints away from the pivot point, creating a smooth transition of the spline at the pivot.

- to change a smooth pivot point to a non-smooth point, drag an adjacent control point away from the line with the pivot point. This will break the tangency of the control points at the pivot.

- to change a non-smooth pivot point to a smooth point, drag an adjacent control point. This will restore the tangent connection of the control points at the pivot.

To change the pivot type using the Point Type menu, while in edit mode, click on a pivot point. This causes the menu to display the item that corresponds to the current type of the selected pivot point. Select the menu item with the type to which you wish to change. The curve is regenerated with the new type and is displayed. However, you may also need to move the pivot point for the change to become visible.
Transform Object Axes

This tool is used to change the location and rotation of an object’s axes relative to the object. Clicking on an object with this tool sets the object in edit mode and displays the controls through which the object axes can be moved and/or rotated. These controls are shown in Figure 4.3.7.5.

Both the prepick and postpick methods can be used with this tool and the axes of objects can be edited one at a time or the axes of a number of objects can be edited simultaneously. This is controlled by selecting either Transform All or Transform Individually in the Transform Object Axes dialog shown in Figure 4.3.7.6.

To transform the axes of a single object, with the Transform Object Axes tool active, the object is picked. To transform the axes of more than one object, the Pick tool is used first to preselect them. Then, with the Transform Object Axes tool click anywhere in the project window. The selected objects are placed in a special edit mode, where the object’s axes are represented as transform controls. This edit mode is the same as for the Transform tool and is described in more detail in section 4.23.2. The only difference is that the scale transformation, which is activated by clicking in the arrow control, is not available. Instead, clicking in the arrow initiates a move along the arrow’s axis.
**Edit Surface**

This tool is used to directly edit the surface of parametric objects, which can again be edited one at a time or more than one of the same type may be edited simultaneously. The Edit Controls Options dialog (Figure 4.3.7.1), discussed earlier, can also be invoked from this tool to adjust the resolution of the surface of an object during editing. Both the prepick and postpick selection method can be used as with the Edit Controls tool.

To edit a single object, with the Edit Surface tool active click on the object. To edit more than one object, use the Pick tool to preselect them. Then with the Edit Surface tool click anywhere in the graphics window. All objects are placed in edit mode and are highlighted, but no controls appear. The highlight color is red by default but can be changed. Editing occurs not by clicking on a control, but by clicking directly on a surface, which causes a little arrow to appear temporarily. This arrow indicates the position and direction of the subsequent transformation, which is applied immediately by dragging the mouse.

This is the only method available for directly transforming the surface of a parametric object. However, the type of transformation and the method by which its direction is calculated may vary. These options are selected from the Surface Edit Options pop up menu for the respective type of object, which works as for the Edit Controls tool. This menu pops up at the cursor location, when clicking in the modeling window, while pressing option (Macintosh) or ctrl+shift (Windows).

Picking and editing with the Edit Surface tool is different from the Edit Controls tool. Since there are no controls to select, the basic operation is to pick a point on the surface of the object and to move it to a new location. As the point is moved, the surface is transformed based on the parametric nature of the object, to follow the new location of the point. The first click is the point on the surface that will be transformed. It is marked with a cross and the entire object is drawn in the highlight color. Once the shape is changed as desired, an additional click ends the transformation. You can apply additional transformations by repeating the process of picking on the surface and moving points to a new location. Click away from the object or press esc to terminate the surface editing.
Edit in Parameter Space

After creating a **curve on nurbz** or **trimmed nurbz** object, the shape of the curves can be edited in two distinctly different ways:

1. The Edit Controls tool can be used to edit the underlying NURBS surface. After clicking on the object with the Edit Controls tool, the control points of the surface are displayed and can be moved in the usual manner. This is discussed in more detail in section 4.13.9 of the *form·Z* User’s Manual.

2. The curves on the nurbz surface can be edited in parameter space using this tool.

Analytic surfaces, including nurbz, can be thought of as 2 dimensional constructs. For example, a sphere has a latitude and longitude direction. A NURBS surface has a length (U) and depth (V) direction. Although the shape of these objects describes a three dimensional form, the two directions define a two dimensional space, called **parameter space**. The latter is always bounded and by convention defines coordinates from the origin (0,0) at the lower left corner to the upper right corner (1,1). Any 2D point in parameter space has an equivalent point in 3D space. The 3D point can be calculated from the 2D point by applying the mapping formula for a particular analytic shape. This is illustrated in Figure 4.3.7.7.

An untrimmed nurbz object uses the entire parameter space for its 3D shape. That is, the four boundary edges of the object in 3D space are represented by the edges of the unit square in parameter space. If the nurbz object is trimmed, the trimming curves are contained inside the parameter space; they delineate holes that are inside the outer boundary of the nurbz object. An example of a trimmed nurbz object in 3D space and its trimming curves in 2D space is shown in Figure 4.3.7.8.
The curves of a curve on nurbz object and the trimming curves of a trimmed nurbz object can be edited in parameter space with the Edit In Parameter Space tool. With this tool active click on a curve on nurbz or trimmed nurbz object. This will invoke the Edit In Parameter Space dialog, shown in Figure 4.3.7.9.

![Figure 4.3.7.9: The Edit In Parameter Space dialog](image)

(a) before and (b) after editing the controls of a parameter space curve.

This dialog shows the object in **World (3D) Space** on the left and the curves of the object in **Parameter Space** on the right. With the arrow tool of the right window active, click on any curve in parameter space. This will activate the editing controls of the curve. These controls are the same as the editing controls for composite curves, arcs, and splines, when using the Edit Controls tool, as illustrated in Figure 4.3.7.9(b). The shape of the parameter space curves can be changed by dragging the controls in the usual manner. The parameter space curve being edited is drawn in the highlight color in the 3D window. After double clicking to end the editing of a curve in the parameter space, the preview on the left is updated. Note that moving a curve outside the parameter space boundaries is not allowed. When editing the curves of a trimmed nurbz object, **form-Z** also checks for intersecting curves, which are not allowed. The editing is restricted to non intersecting curves only.

As already mentioned, the parameter space is defined over a unit range of (0,0) to (1,1). The grid and snapping parameters can be set with options found under the **Parameter Space** window.

**Show Grid:** When this option is on, an orthogonal grid is displayed in the parameter space.

**# Of Grid Lines:** The value entered in this field determines how many grid lines are drawn, in both the U and V directions. Note that the cells of the grid are square always.

**Snap To Grid:** When this option is on, the mouse will snap to the grid points when editing the control points of a curve in the parameter space.
4.4 Insertions

The form of a previously generated object may be transformed through the insertion of points, segments, outlines, faces, volumes, or holes. The type of insertion to be executed is determined by the active insertion mode, which is set by the Insertion modifiers found in the magenta tool palette. The insertions work in conjunction with the drawing tools. The shape to be inserted is drawn using one of the drawing tools. This shape then becomes the source for the execution of an insertion. How the source shape is interpreted depends on the selected insertion mode.

The insertions belong to the group of operators known as topological transformations or topological editing operations. Their opposites are the deletions, which are discussed in section 4.26. When they are complemented by the geometric transformations (discussed in section 4.23), they provide highly effective sculpting capabilities.

The following topological insertions that can be executed by form•Z are discussed in this section:

- Insert Outline
- Insert Face/Volume
- Insert Hole
- Insert Opening

Two more insertions are available (Insert Point and Insert Segments), but are discussed with the line editing operations in section 4.21. Except for the latter insertions, the insertion operations require preselection of the face to which an insertion will be made. However, when the Clicking Inside Boundaries option is on in the Pick Options dialog, the selection of the insertion face may be accomplished with the first point drawn, whenever that point is inside the boundaries of the insertion face. This effectively cancels the requirement for prepicking the insertion face. Except for the point and segment insertions, the insertion face must be planar.

Note that, while the insertions are discussed at this point in this User’s Manual, because we are following the order in which the tools are structured in the default tool bar, they involve concepts that have not yet been covered, if you are reading the material in the order it appears in the manual. Consequently, you may prefer to come back and read this section later.
4.4.1 Inserting outlines

Outlines are inserted by selecting the Insert Outline modifier.

![Insert Outline]

When this modifier is active and a vector line or spline is properly drawn, it is inserted into a face. The line drawn may be open or closed. Any of the drawing tools may be used to draw the shape that will be inserted. Outline insertions are illustrated in Figure 4.4.1.1.

The insertion may be applied to a smooth or facetted object. In the former case the result is always a smooth object. Likewise, when a facetted shape is inserted into a facetted object the result is always a facetted object. When a smooth shape is inserted onto a facetted object the result is determined by the option selected from the **Insertion Options** dialog, shown in Figure 4.4.1.2. This dialog can be invoked directly from all the insertion tools.

**Facetted Options:** When inserting a smooth shape into a facetted object, the facetted object may become smooth or may stay facetted. This is determined by the selection from this pair of options.

**Keep Facetted:** When this option is selected and the input shape is a smooth shape, such as a circle or a spline curve, and the object is facetted, the insertion is performed as a facetted operation. That is, the smooth curve is converted to a facetted shape. The facetted resolution of the curve is determined by the parameters set in the **Resolution** tab. After the operation is complete, the object remains facetted.

**Make Smooth:** When this option is selected, the insertion is executed as a smooth operation. That is, after the operation is complete, a facetted object has been converted to a smooth object, if the input shape was smooth.
To insert an outline into a face, the face must be prepicked, and must be planar. If the shape to be inserted is open, both of its endpoints should lie on distinct segments and/or points of the face. The body of the insertion line should be completely contained within the boundaries of the face in which it is inserted. This is a requirement for both open and closed outlines. Should any of these conditions be violated, the system beeps, issues an error message, and cancels the execution of the insertion.

The insertion of an outline splits the original face into two new faces, which are coplanar. The structure of the new faces depends on whether an open or a closed shape is inserted. The insertion of an open vector line results in a split of the original face. The insertion of a closed outline subdivides the original face to two structurally different faces. One face consists of two outlines, an outer and an inner outline (hole). The other face consists of a single outline, which is reversely coincident to the inner outline of the first face. From a practical point of view, the second face covers the hole created in the first face.

After completion of an outline insertion, the new face that is to the right of the inserted outline remains picked and highlighted. This way you may continue with the execution of another insertion. However, if you wish to have the other face picked, you can use the **Switch Pick** operation to reverse the picks, as illustrated in Figure 4.4.1.3. Double clicking the mouse away from any selectable entity unpicks the currently picked face and picks the other new face. The Switch Pick operation is intended to be used immediately after the completion of an insertion. Any number of Switch Pick operations may be executed in a sequence to toggle between the two faces. The ability to switch picks is lost when a new tool is selected. It should be noted that, while the Switch Pick operation provides a quick method for controlling which face is picked, the general pick/unpick procedures can also be freely employed.

![Figure 4.4.1.3: Using the Switch Pick operation:](image)

(a) Drawing the outline right-to-left, returns the upper of the two new faces highlighted.
(b) A double click of the mouse anywhere away from the object, reverses the highlight.

Note that, when you draw on a preselected face for the purpose of executing an insertion, the system uses a temporary (invisible to you) reference plane that is generated on the surface of the preselected face. Where your first point is clicked when preselecting the face is significant. The segment where you click determines the direction of the X axis of the temporary reference plane. When you draw irregular shapes, the direction of the X axis has little significance. However, when you draw regular shapes, such as rectangles, the direction of the X axis affects the direction in which the rectangle is drawn.
4.4.2 Inserting faces and volumes

Faces or volumes are inserted by selecting the Insert Face/Volume modifier.

When this tool is active and a proper vector line is drawn, the system inserts a face or rather a set of faces in a face. When the sequence of faces is closed, it results in the insertion of a volume.

The faces that are inserted are derived from the source shape you draw, by extruding it in a direction perpendicular to the plane of the face to which the insertions are applied. The distance of the extrusion is determined by the active height value, selected from the Heights menu. The sign of the height is significant. Positive heights create extrusions on the positive side of the face, and negative heights create extrusions on the negative side.

The height of the insertion can also be set interactively through graphic or numeric input. This is done when the Graphic/Keyed item is selected in the Heights menu. When it is, as soon as you finish drawing the insertion shape, the insertion volume is rubber banded and follows the motion of the mouse. The direction of the motion is always perpendicular to the surface of the insertion face. The final height of the insertion volume can be set away from the insertion face (which is a positive height) or towards the inside of the face (which is a negative height). The height can also be typed in the Prompts palette and entered by pressing return.

The source shape may be open or closed and it may consist of a single segment or a connected sequence of segments. When open, both of its endpoints should lie on segments or points of the face to which the insertion will be applied. The body of the source shape should be completely contained within the boundaries of that face.

The insertion of faces derived from an open vector line produces a step. The original face is subdivided into two new faces, and the one to the right of the drawn line is moved in a direction perpendicular to its plane. These two faces are parallel to the original face. New rectangular faces are also generated and inserted between the two faces. The latter are perpendicular to the pair of parallel faces. Face insertions are shown in Figure 4.4.2.1.

Figure 4.4.2.1: Inserting faces by drawing an open vector line with (a) a positive and (b) a negative height value.
When the sequence of faces to be inserted is derived from a closed source shape, the shape may be completely contained within the boundaries of the face in which it will be inserted, or it may be crossing its boundaries. The shape must not be completely outside the boundaries of the face. When the source shape is completely within the boundaries, a true face insertion is executed. When it crosses the boundaries, a Boolean union is automatically executed. A true insertion is analogous to the insertion of an open sequence of faces. A step is again produced, but this step is now free standing, somewhere within the original face.

Depending on the sign of the height used, the step may be outwards or inwards, relative to the plane of the face in which it is inserted (Figure 4.4.2.2). Semantically speaking, the insertion of a closed sequence of faces is equivalent to the insertion (or extraction) of a volume.

When inserting a face to a smooth object, the extrusion can be either **Perpendicular To Object Surface** or **Perpendicular To View Plane**. One of these options is selected from the **Insertion Options** dialog, shown in Figure 4.4.2.3, that can be invoked from any of the Insertion modifiers.

Note that in both cases, a shape is drawn on the surface where the insertion will occur. When this surface is not flat, the shape is drawn on a plane that is tangent to the curved surface at the point of the first mouse click. After the shape is drawn, it is projected onto the curved surface and is extruded according to the selected option. When **Perpendicular To Object Surface** is on, the extrusion is along the normal at the first click point. If **Perpendicular To View Plane** is on, the extrusion is along the view line, which is perpendicular to the surface of the screen. Note that, in the latter case, when **Graphic/Keyed** (or dynamic input) is used for Height entry, we can only draw dynamically if we use a perspective view or multiple windows. If we use an axonometric view and a single window the height will have to be typed in the Prompts palette. Examples are shown in Figure 4.4.2.4.

![Figure 4.4.2.2: Inserting faces by drawing a closed shape and using: (a) a positive and (b) a negative height.](image)

![Figure 4.4.2.3: Lower part of the Insertion Options dialog.](image)

![Figure 4.4.2.4: (a) Inserting faces using (b) Perpendicular To Object Surface and (c) Perpendicular To View Plane. (1) Axonometric, (2) perspective, (3) top view.](image)
When the system determines that a Boolean union is to be executed, it internally generates a temporary solid object by extruding the source shape. The height of that object may be towards the outside or towards the inside of the face, where the insertion is to be applied. This depends on the sign of the currently selected height value. The original object and the new temporary object are used as operands for the union operator. The new solid object, which is generated by the union operation, replaces the original object and the temporary object is deleted. Face insertions that are executed as unions are illustrated in Figure 4.4.2.5.

A Boolean union is also executed when the face insertion operation produces a result in which an inserted face intersects another face of the object. One such case arises when an open curve is drawn for the face insertion, and the height used is higher than the relative height of an adjacent face. An example is shown in Figure 4.4.2.6(a). Another case occurs when a face on the side of the one where the insertion is made contains a hole, which is intersected by one of the inserted faces. An example is illustrated in Figure 4.4.2.6(c). When such cases are encountered, the system cancels the original insertion and executes it by applying a union operation, which is capable of resolving the face intersections. After the application of the union, the case of Figure 4.4.2.6(a) is resolved as shown in 4.4.2.6(b) and that in Figure 4.4.2.6(c) is resolved as in 4.4.2.6(d).

As for the outline insertions, after the completion of a face insertion, the face that is extruded is returned highlighted. If an open shape is drawn for the insertion, the extruded and highlighted face is always to its right. If a closed curve is drawn, that curve is extruded and highlighted. The **Switch Pick** operator may be used to reverse the current highlighting. A Switch Pick is executed by double clicking the mouse away from any pickable entity. When used immediately after a face insertion, the Switch Pick operation will highlight the portion of the original face that was not extruded.

To insert one or more faces into a face, the insertion face must be prepicked and must be planar. Furthermore, face/volume insertions can only be made to well formed solid objects. They have no effect on incompletely formed surface objects. If the system encounters a violation of the required conditions, it beeps, issues an error message, and cancels the execution of the insertion.

**Figure 4.4.2.5:** Inserting faces by drawing a closed shape that crosses the boundary of the insertion face. These cases are executed as a Boolean union. Using (a) positive and (b) negative height.

**Figure 4.4.2.6:** Inserting faces: (a) Insertion causes two faces to intersect and (b) a union is executed. (c) Inserted face crosses another face’s hole; (d) a union is executed.
4.4.3 Inserting holes

Holes are inserted by selecting the Insert Hole modifier.

When this tool is active and a proper shape is drawn, the system inserts a hole either to an incompletely formed but closed surface object, or to a well formed solid object, through one of its faces. Semantically speaking, a hole insertion may produce distinctly different results, depending on the type of the object to which it is applied. In all cases, the hole to be inserted is derived from a source shape that is drawn with one of the drawing tools. The source shape is required to be closed. The face to which the hole is inserted must be prepicked and planar.

A hole is inserted in a well formed solid by executing a Boolean difference. The source shape may cross the boundary of the insertion face but cannot be completely outside. The hole that is drilled is a 3D entity, the depth of which is determined by the current height value, selected from the Heights menu. If Graphic/Keyed is selected, the height is entered interactively, as for the face insertions. Most commonly, a negative height is required in order to drill a hole through the volume of an object, since that volume is usually on the negative side of the face that is used as a reference. There are also instances where desirable results may only be derived by using a positive height.

The Boolean difference is executed in a fashion similar to the union that may be executed in the place of a face insertion. A temporary extruded solid is first derived from the source shape, using the current height value. The Difference operation subtracts the temporary solid from the original object. The resulting solid replaces the original object and the temporary solid is deleted. The depth of the hole depends on the height value used. The hole insertion may create a cavity, or a hole may be drilled all the way through an object, or a hole may even be drilled through more than one “wall” of an object, when the object is a 3D enclosure. Insertion of holes to 3D solids is illustrated in Figure 4.4.3.1.

Figure 4.4.3.1: Inserting holes:
(a) The hole insertion results in a cavity.
(b) The hole is drilled through the solid.
(c) The hole drills one of the walls.
(d) The hole insertion drills the top and bottom slabs of the 3D enclosure.
As illustrated in Figure 4.4.3.2, the Insert Hole operation may split an object into two or more volumes. These volumes are always created as a single object. The option to make each volume a separate object that is available to the Boolean operators does not apply to the insert hole operation. However, the Separate Volumes tool can be used to produce an independent object from each distinct volume that may result from a hole insertion (see subsection 4.18.2).

Figure 4.4.3.2: Inserting a hole that splits a solid into two.  (a) A 3D enclosure.  (b) The front face is picked and the shape of the hole is drawn.  A -20' height is used.  (c) The hole insertion splits the enclosure into two volumes, which remain parts of a single object.  (d) A hidden line display of the result.

Note that drilling a cavity that is completely contained within the boundaries of a face, gives a result that can also be derived through the insertion of a closed sequence of faces and a negative height. In such cases, the latter operation is actually faster and should be preferred.

To insert a hole into a surface object, that surface object must be a closed shape and the source shape must be completely contained within the boundaries of the surface object. When applied to a surface object, the Insert Hole operation adds an inner outline to that object. Examples of allowed and disallowed hole insertions to surface objects are shown in Figure 4.4.3.3.

Figure 4.4.3.3: Inserting holes in surface objects:  (a) Inserting a triangular hole in a rectangle.  (b) More holes can be inserted in a shape with a hole.  (c) Holes cannot cross the boundary of the shape.  (d) Holes cannot be inserted in open shapes.
4.4.4 Inserting openings

The openings are special types of holes and are inserted by selecting the Insert Opening modifier at the right end of the second row in the modeling tool palette.

![Insert Opening]

When this tool is active and a proper closed shape is drawn on a preselected face, the system drills an opening (hole) through the “wall” defined by the preselected face and its opposite. The system automatically finds the opposite face, when a proper opposite face exists. When it can not find one, it beeps and issues an error message. The insertion of a “window” opening is illustrated in Figure 4.4.4.1.

![Figure 4.4.4.1: Example of opening insertion on a 3D enclosure:](image)

(a) A face is prepicked and a rectangle is drawn on the face, from point 1 to point 2.
(b) The opening is inserted. (c) Hidden line image of the result.

There are some similarities between the results produced by the Insert Hole (previous section) and the Insert Opening modifiers. Actually, the results are sometimes identical. However, the two operations are significantly different in the manner in which they are executed. Recall that Insert Hole requires the selection of a (typically negative) height value and is always executed as a Boolean difference. Insert Opening does not require a height value and drills a “hole” all the way through a pair of opposite (frequently parallel) faces. It also inserts the opening shape directly, without executing a Boolean operation. This makes it much faster than the hole insertion. However, to be able to insert an opening, the shape drawn for the insertion should be completely contained within the boundaries of the face in which it will be inserted. The projection of the drawn shape on the opposite face should also be completely within the boundaries of the opposite face. These requirements make the opening insertion operation less general than Insert Hole, since the latter does not have these restriction. However, the Insert Opening operation works well and quickly when window openings are inserted on models of buildings.
When inserting openings, the preselected face as well as its opposite face should be planar. If they are not, the system beeps and issues an internal error message. Attempts to insert improper openings are illustrated in Figure 4.4.4.2.

Figure 4.4.4.2: Invalid opening insertions: (a) Opening shape crosses the face's boundary. (b) Opening shape is completely outside the face's boundary. (c) Opposite face is not planar. (d) Opening shape crosses the opposite face's boundary.

Frequently, the results produced by the Insert Opening operation can also be produced by using the Insert Hole operation (while the reverse is less frequently true). The Insert Opening operation can only be applied to well-formed solids. The opening is constructed from the drawn shape, in a direction perpendicular to the selected face, through the opposite face. The opposite face is the face directly behind the selected face. The inside faces of the hole are perpendicular to the preselected face, and are also perpendicular to the opposite face if the two faces are parallel.

The main intention of the Insert Opening operation is for the creation of “window” openings on the “walls” of buildings. These walls are typically delineated by parallel faces. However, the Insert Opening operation is not restricted to holes through parallel faces only. Openings can also be drilled through pairs of non-parallel faces. Note that, since the holes are drilled in a direction perpendicular to the face in which they are drawn, the directions of such holes will vary, depending on which face is used to draw the shape of the hole. Examples are shown in Figure 4.4.4.3.

Figure 4.4.4.3: (a) Four holes inserted in a non-rectangular solid. (b) Top view showing the directions of the holes. (c) Hidden line image of the result.

After completion of the Insert Opening operation, the preselected face remains selected and highlighted, and you may continue to insert more openings through the same face. Or you can double-click the mouse (execute a Switch Pick operation) to toggle the highlight between the original and its opposite face.
4.4.5 Bypassing the face preselection requirement

The Insert Outline, Insert Face, Insert Hole, and Insert Opening operations require you to identify (preselect) the face where the insertion will be applied. This is normally done by prepicking the face before the execution of the operation. However, under certain conditions, the face selection can also be made through the drawing process.

When the **Clicking Inside Boundaries** option is selected in the **Pick Options** dialog (the one point face selection option), a closed shape is drawn for the insertion, and the first point is drawn inside the boundaries of the insertion face, then that face does not need to be prepicked. The system can use the first point to identify the face to insert into. Recall that Outline and Face insertions accept either open or closed shapes. When the insertion shapes are open, then drawing must start on a segment of the object, which does not make it easy to select a unique face. In these cases, the face should be preselected.

Insert Hole and Insert Opening accept only closed shapes. In addition, Insert Opening requires that the insertion shape lie within the boundaries of the face. This makes the automatic selection of the face always applicable. This is not true for the Hole Insertions that allow part of the insertion shape to be outside the boundary of the face.

Note that even though the automatic selection of the insertion face is in many cases possible and desirable, you may still prefer to preselect the face. This is because the manner in which you select the face establishes the direction of the Cartesian axes that will be used for the temporary reference plane on which you will be drawing. When the **Clicking On Edges** (two point) method is used for selecting a face, the first segment picked establishes the direction of the X axis. When the **Clicking Inside Boundaries** (one point) method is used, then the segment that is closest to the pick point establishes the X direction. When you draw for an insertion you have to place your first point on the position required by the shape you wish to insert. This does not allow you much flexibility for selecting a desirable X direction. This is seldom a problem with rectangular shapes. When the shapes of your insertion faces are more or less irregular, you will probably prefer to preselect them, before drawing your insertion shape.
4.5 Derivative objects

The following derivative object generation tools, which are all the tools (except one) on the fifth left row of the default tool bar, are discussed in this section. More derivative objects, generated by tools on the fifth row (left and right), are discussed in subsequent sections.

- 2D Surface
- 2D Enclosure
- 3D Extrusion
- 3D Converged
- 3D Enclosure
- Frame
- Parallel
- Projection
- Unfold

Derivative objects are generated from already existing objects or their parts. The top five tools above derive objects that can also be generated directly, as discussed in section 4.2. The last three tools offer object generation methods that are only available as derivative methods.

While derivative methods lack the immediacy of direct object generation methods, they are in many ways richer. Previously generated primitive shapes can be manipulated and/or combined into complex composites, and can then be used to derive other objects with forms that are not possible through direct object generation. Examples are shown in Figure 4.5.0.1. Derivative generation also allows positioning objects relative to other objects.

The generation of a derivative object requires a source shape, which may be a 2D shape or a 3D object. When a 2D shape, it may be a complete surface object or a part of an object. Parts that may be selected as source shapes are the point, the segment, the outline, or the face. Which part of an object is picked as a source shape is determined by the active topological level.
The derivative object generation operators work well with both the **prepick** and **postpick** methods. For the prepick method, any number of entities are first picked with the Pick tool, then the desired derivative object tool is activated and the mouse is clicked once in the graphics window. For some of the operators, the latter click of the mouse picks additional information, such as the axis of rotation for the revolutions, or the cutting shape for the sections. The system then derives an object for each of the prepicked entities, which may be at different topological levels.

In the postpick method, the desired tool is activated first and then the source shape is selected. The derivation of the new object is executed immediately. Again, some of the tools require additional information before they may proceed with the generation of the object. These requirements are discussed in their respective sections.

Picking an object as a source shape makes complete sense when that object is of the surface type. The system also allows the user to pick a solid object as a source shape, but it applies a special interpretation. It picks each face of the object as a source shape and uses each one of them for the derivation of a new object. Such cases are discussed at the end of subsection 4.5.3.

**Facetted versus smooth model types**

While there are no options in the derivative tool dialogs for selecting **Facetted** or **Smooth**, most derivative objects may be either facetted or smooth. This is true for models derived with the 2D Surface, 2D Enclosure, 3D Extrusion, 3D Convergence, 3D Enclosure, and Parallel tools. In these cases, the type of model is determined by the type of the source shape. For example, if an extrusion is derived from a facetted circle, the resulting cylindrical shape will also be facetted. If, on the other hand, the extrusion is derived from a smooth circle, the resulting 3D form is also smooth. Objects generated with the Projection and Unfold tools are always facetted.

**The status of objects settings**

Nearly all the dialogs of the derivative tools contain a tab labeled **Status Of Objects**, whose content is shown in Figure 4.5.0.2. Its options are as follows:

**Global**: When this option is on, the options set in the dialog affect all the operations. The alternative is the name of the tool from which the dialog was opened. Such a name is invoked by deselecting **Global**. When a name is displayed, the settings in the dialog affect the specific operation only.

**Operand Status**: The three options in this group determine how the one or more objects involved in an operation are treated after the completion of the operation. They may be kept as they are, they may be ghosted, or they may be deleted.

**New Object Status**: These options only affect operations that may produce results consisting of more than one part (volume). **Object Per Volume** constructs a separate object from each of the resulting volumes. **Single Object** structures all the volumes as a single object.
4.5.1 Deriving point objects and point clouds

A point cloud is a generally unstructured collection of points that exist as a single object. Characteristic of these objects is that they consist of geometry (the coordinates of the points) but no topology (the points are disconnected). Single point objects or point clouds can be derived from any existing object, solid or surface, faceted or smooth, or from parts of objects, using the following tool.

Point Cloud

This tool works at all topological levels and the result is a point cloud formed from all of the points of that level. For example, when a face is selected, the result is all the points of that face. When an object is selected, all the points of the object form the cloud. When a single point is selected (with topological level set at Point), the result is a single point object.

Either the prepick or postpick method can be used with this tool. With the postpick method, the Point Cloud tool is selected first and then a point, segment, outline, face, hole, object, or group is selected, depending on the topological level currently active. With the prepick method, any number of entities, which can be at different topological levels, are first selected with the Pick tool, and then, with the Point Cloud tool active, the mouse is clicked anywhere in the graphics window. Examples of point clouds derived from different objects and entities are shown in Figure 4.5.1.1.

Figure 4.5.1.1: Point cloud examples from (a) faceted geodesic sphere, (b) metaformz, and (c) the top face of a smooth cylinder.
4.5.2 Derivative surface objects

2D surface objects may be derived from other surface objects, segments, outlines, or faces of objects.

**2D Surface**

This tool can be used to derive surface objects from selected entities (such as faces, outlines, and segments), from the boundary of another surface object, from a *stitch* line, from a sequence of segments, or from groups of adjacent faces. The type of surface object to be derived is selected in the Derivative Surface Options dialog (Figure 4.5.2.1), invoked directly from the 2D Surface Object tool.

This operator provides an alternative method for copying surface objects, as well as an effective method for making independent surface objects from various parts of an object. These surface objects are created in exactly the same position as the original entity was located.

**From each selected entity**

When the Each Selected Entity option is active, either the postpick or prepick methods may be used. With the postpick method, the Derivative 2D Surface Object tool is activated first and then the source entity is picked. The 2D surface object is derived immediately. When the prepick method is used, any number of source shapes may be picked, followed by the selection of the 2D Surface Object tool. Then the mouse is clicked once anywhere in the graphics window.

For both the prepick and the postpick methods, the topological level can be set to Segment, Outline, Face, Object, Group, or Hole, but not to Point. When set to Object or Group, one surface object will be generated for each face of the picked object.

The generation of independent surface objects from the faces of a cuboid and a pyramid is illustrated in Figure 4.5.2.2.

**Figure 4.5.2.2:** 2D surface objects derived from the faces of (a) a cuboid and (b) a pyramid.
From boundary of surface object

A surface object may consist of one or more faces. Boundary segments of a surface object are those segments that have a face on only one side. These are the segments for which arrows are drawn when the wire frame option Show Directions is on, as shown in Figure 4.5.2.3. A surface object may be bounded by one or more sequences of such segments, which are always closed sequences.

When the Boundary Of Surface Object option is selected, a surface object is generated from one or more boundary sequences of segments. With the postpick method, select the 2D Surface Object tool and click on a boundary segment. The system will automatically find all the other segments that belong to the boundary sequence, and will derive a surface object from them. This object is always a closed surface object. If you click on an internal segment (a segment with faces on both sides) when executing this operation, an error message will be posted.

This operation can also be executed using the prepick method. Any number of boundary segments can be selected using the Pick tool. Then, with the 2D Surface Object tool active, click anywhere in the graphics window. If more than one boundary segment belonging to the same sequence of boundary segments is preselected, a single surface object is generated. That is, if a prepicked segment has already been used in the generation of a surface object, it is not used again. Examples are shown in Figure 4.5.2.4.

From closed stitch

A stitch is defined as for the round stitch operation, which is a closed sequence of internal segments that meet at angles of no less than 100°. Internal segments are those which have faces on both of their sides.

When the Closed Stitch option is on, the operation is executed by clicking on an internal segment with the 2D Surface tool active (postpick method). The system automatically derives a sequence of segments that form a stitch (provided that one exists), and derives a surface object from it. If a boundary segment is picked, an error message is posted. The operation can also be executed using the prepick method. Examples are shown in Figure 4.5.2.5.
From selected segments

This operation can be used to generate surface objects from sequences of preselected segments. While it can be executed using both the postpick and the prepick methods, the prepick method is more appropriate.

With the postpick method, select **Selected Segments** in the **Derivative Surface Options** dialog, then activate the 2D Surface Object tool and click on a segment of an object. A surface object that consists of a single segment is generated. The postpick method allows the selection of only one segment at a time, and only single segment surface objects can be generated with this method. The same result can be achieved when selecting **Each Selected Entity** in the **Derivative Surface Options** dialog, and clicking on a segment with topological level set to Segment.

With the prepick method, select any number of segments with the Pick tool. Then select the 2D Surface Object tool and click anywhere in the graphics window. An object is generated for each connected sequence of segments. Examples are shown in Figure 4.5.2.6.

From selected faces

This operation is used to generate surface objects from any number of adjacent faces. The operation can be executed using either the postpick or the prepick method. However, the postpick method can select only one face at a time, which makes the use of the prepick method more appropriate.

When using the prepick method, select any number of faces with the Pick tool. Then select the 2D Surface Object tool and click in the graphics window. A separate surface object is generated from each group of adjacent faces. Examples are shown in Figure 4.5.2.7.
4.5.3 Derivative 2D enclosures

2D enclosures may be derived from 2D surface objects, other 2D enclosures, segments, outlines, or faces of objects. Deriving a 2D enclosure from another 2D enclosure is an alternative method to copying it.

2D Enclosure

This tool is used to derive 2D enclosures from other source shapes. It follows the same rules and is affected by the same options and parameters that apply to the direct object generation method. These options are selected from the 2D Enclosure Options dialog, which can be invoked directly from this tool (Figure 4.2.1.6 in subsection 4.2.1). The postpick and prepick methods work identically to the derivative 2D Surface tool when surface objects are generated from Each Selected Entity.

Examples are shown in Figure 4.5.3.1. The examples were generated with the Justification option set to Left, and the Enclosure Wall Width parameter set to 2 feet.

Figure 4.5.3.1: 2D enclosures derived from the faces of (a) a cuboid and (b) a pyramid.
3D extrusions, convergences, and enclosures

The three direct object generation methods that produce solids, namely Extrusion, Convergence, and 3D Enclosure, have derivative counterparts. Parallel extrusions, extrusions to a point (convergences), and extrusions from 2D enclosures can be generated from open or closed surface objects, faces, outlines, or segments. They can also be generated from complete solids, in which case one object is generated from each face of the selected solid (for more discussion, see end of this section).

The direction of the extrusion may be perpendicular to the active reference plane or perpendicular to the plane of the source shape. These options are selected from the Extrusion/Convergence Options dialog, shown in Figure 4.5.4.1, which can be invoked directly from the derivative 3D Extrusion and 3D Convergence tools. For the 3D Enclosures, these options are selected from the 3D Enclosure Options dialog (Figure 4.2.1.16), invoked directly from the 3D Enclosure tool.

Perpendicular To Reference Plane: When this option is selected (default), the extrusion will be in a direction perpendicular to the active reference plane.

Perpendicular To Surface: When this option is selected, the extrusion will be in a direction perpendicular to the surface of the source shape when the source shape is a surface object, a face, or an outline (Figure 4.5.4.2). When the source shape is a segment of an object, the extrusion is always perpendicular to the reference plane (Figure 4.5.4.3). When the source shape lies on or is parallel to the active reference plane, the distinction makes no difference. When the source shape is not planar, its first three points are used to define the plane perpendicular to which the Extrusion is executed. Note that this is similar to the way the direction of the extrusion is determined for the direct object generation method.

![Figure 4.5.4.1: The Extrusion/Convergence Options dialog.](image)

![Figure 4.5.4.2: Extrusion from face with (a) Perpendicular To Reference Plane, and (b) Perpendicular To Surface.](image)

![Figure 4.5.4.3: The direction has no effect on extrusions from segments: (a) Perpendicular To Reference Plane and (b) Perpendicular To Surface on. Both produce same result.](image)
As with the direct object generation extrusions, the height of the derivative extrusion is determined either by selecting one of the numerical values in the Heights menu, or interactively by selecting the Graphic/Keyed item. With the interactive method, your shape is rubber banded as soon as you select the source shape with one of the derivative object tools active. The top end of the extruded object follows the motion of the mouse in a direction perpendicular to either the reference plane or the surface of the source shape, depending on the option selected. The next click determines the final height of the object or, instead, a height value may be entered in the Prompts palette, and the return/enter key pressed.

When prepicking more than one source shape, or when an object with more than one face is selected (in which case a derivative object will be generated from each face), the Graphic/Keyed height can be applied in either one of two ways. You may parade through each of the source shapes and determine the height of each extrusion graphically or numerically, one at a time. Alternatively, you may determine the height of only the first extrusion and have the same height applied to all the other extrusions. The latter method is applied by pressing the option (Macintosh) or ctrl+shift (Windows) keys, when you execute the derivative object operation.

**3D Extrusion**

This tool derives an extruded solid from a 2D surface object, a 2D enclosure, a segment, an outline, or a face of an object. The derivation of an extruded object from a point is not allowed. Derivative 3D extrusions are illustrated in Figure 4.5.4.4.

**3D Converged**

This tool derives a converged solid (or an extrusion to a point). It is derived as is the parallel extrusion, except that the extruded source shape is converged to a single apex that corresponds to the center of gravity of the source shape, extruded to the height selected. All other conditions are as for the parallel extrusions. Derivative 3D convergences are illustrated in Figure 4.5.4.5.

**Figure 4.5.4.4:** Derivative 3D extrusions generated from 2D surface objects.

**Figure 4.5.4.5:** Derivative 3D converged objects generated from 2D surface objects.
3D Enclosure

This tool derives 3D enclosures from 2D surface objects, 2D enclosures, segments, outlines, or faces (but not points) of objects. Derivative 3D enclosures are illustrated in Figure 4.5.4.6.

3D enclosures are derived by first generating a 2D enclosure from the source shape, when the source shape is not already a 2D enclosure. The system recognizes a 2D surface object to be a 2D enclosure when it has one or more holes. Otherwise, it recognizes the 2D object as a single line surface object, even when it was initially generated as a 2D enclosure. These cases are illustrated in Figure 4.5.4.7.

The height and direction of the 3D enclosures are determined as for the common extrusions. The derivative 3D Enclosure tool is affected by the parameters in the 3D Enclosure Options dialog (Figure 4.2.1.16), as is the direct 3D Enclosure. This dialog can be invoked directly from the derivative 3D Enclosure tool.

Note that when a 2D enclosure is used as the source shape for a 3D extrusion, the resulting object looks like a 3D enclosure, but it is of a restricted type. It can only be constructed with an open top and open bottom regardless of the current options. When the same 2D enclosure is used as a source shape for a 3D enclosure, the enclosure options apply. To produce the same result as the 3D extrusion, the enclosure must be generated with an open top and bottom.
3D solids from other solid objects

The derivative method of generating 3D extrusions, 3D convergences, and 3D enclosures (as well as 2D surface objects and 2D enclosures, see sections 4.5.2 and 4.5.3) requires the source shape to be a 2D shape. Picking a complete solid object, group, or hole as a source shape is also allowed, but is interpreted as picking all the faces of that entity. The derivative object generation procedure derives one object for each face of the selected entity. Figure 4.5.4.8 illustrates the derivation of 3D convergences and 3D enclosures from the faces of a hexagonal extrusion. In both examples, the derivative object generation operators were executed with topological level set to Object and the Perpendicular To Surface option selected in the Extrusion/Convergence Options dialog.

Recall that the default option for the generation of the derivative solids is Perpendicular To Reference Plane, which does not always generate objects for all the faces of an object. If a face is perpendicular to the currently active reference plane, no 3D solid or 2D enclosure is generated for that face. This is because if an object were generated, it would not be a true solid or a true 2D enclosure.

Figure 4.5.4.8: A hexagonal extrusion is used as a source solid object for the derivation of other solids. (a) Pyramids and (b) 3D enclosures are derived for each face of the hexagonal extrusion.
4.5.5 Parallel objects

The parallel objects may be considered to be the generalization of the 2D and 3D enclosures. They can be generated from a variety of other objects using the Parallel Object tool, found on the 5th row, left column of the modeling tool palette.

Parallel objects can be generated using either the prepick or the postpick method. With the prepick method, the Pick tool is used to preselect any number of entities, which can be at different topological levels. Then, with the Parallel Object tool selected, click in the graphics window. With the postpick method, select the Parallel Object tool, and click on the entity, which can be an object, face, outline, or segment, as determined by the currently selected topological level.

Figure 4.5.5.1: Types of parallel objects.
Parallel

This tool can be used to derive parallel objects from other objects. The original object can be a solid, a one-sided or two-sided surface, or an open vector line. Parallel objects can also be generated from faces, outlines or segments of objects. Both single and double parallel objects can be created, as summarized in Figure 4.5.5.1.

The Parallel Options dialog, shown in Figure 4.5.5.2, is invoked directly from the Parallel Object tool by double clicking on the tool. One of two types of parallel objects, as well as other parameters, can be selected from the dialog.

Single Parallel (Surface): When this type is selected, a parallel object on one side of the original object is created. The side the parallel object is generated on depends on whether Out or In is selected. If the original object is a one or two-sided surface object, then the parallel object will also be a surface object. If the original object is a solid, a solid will be produced. If the original object is a vector line, another vector line will be generated.

Double Parallel (Solid): When this type is selected, two sets of parallel faces are created. Whether both of these sets of faces are newly derived parallel faces or one is a copy of the original set depends on the justification option selected. Center creates two new sets of faces. Out and In use a copy of the original set of faces and derive a new set of parallel faces, which is at the outer or inner side of the original object, respectively. The ends of the two sets are connected with additional faces to produce a closed solid object. Thus, double parallel objects are always solids, except when derived from an open vector line, in which case two vector lines are created.

Wall Offset and Slab Offset: These fields contain the distances at which the parallel faces (or lines) are generated. Horizontal faces and faces with at most 5° slope are considered to be slabs. All other faces are considered to be walls. Different offset parameters can be used for slabs and walls.
For open vector lines, the parallel line is derived on the same plane of the vector line, when the vector line is planar. When it is not, planes are calculated for each three points of the vector line and the respective sections of the parallel lines are placed on these planes, as discussed in more detail below. Closed shapes would normally be treated as the faces of solid objects. That is, their parallel shapes are generated on planes parallel to the surfaces of the closed shapes. However, the closed shapes can also be treated as open shapes, and have their parallel lines generated on the same planes, as above, if the following option is selected.

**Parallel On Same Plane (Surface Objects):** When this option is selected and a closed surface shape is selected, its parallel shape is generated on the plane of the surface, as if it were an open vector line. If this option is off (default), the shape is generated on a plane parallel to the plane of the shape. This option is illustrated in Figure 4.5.5.3.

![Figure 4.5.5.3: Parallel objects derived from (1) open and (2) closed shapes: (a) The original shapes; single parallel and Parallel On Same Plane option (b) off and (c) on; double parallel and Parallel On Same Plane option (d) off and (e) on.](image)

**Nurbz Control Points:** This group of options determines how the control points of nurbz objects are handled.

- **Relative To Curvature:** When selected, the parallel surface is built such that control points are distributed relative to the curvature of the surface. The parallel surface will have more control points in areas of high curvature and less in flatter regions.

- **Max Normal Deviation:** Specified in degrees that determine the density of the control point distribution along the surface. Higher values produce fewer and lower values produce more control points.

- **Keep Same Number:** When selected, the parallel surface is built with the same number of control points as the original surface. Note that, unless the original surface has a large number of control points, this option can lead to undesirable results.

- **Accurate:** When selected, a very accurate parallel surface is derived. This requires the addition of a significant number of control points and may change the degree of the surface if necessary.

These options are illustrated in Figure 4.5.5.4.
The derivation of parallel objects is a challenging process since it has to deal with situations for which no straightforward geometric solutions exist. This is particularly true when parallel objects are created from meshed surfaces or solids (see section 4.14). The system tries to resolve these cases by applying certain geometric heuristics. In these cases the results are not always predictable. When you get a result which is different from what you expected, you should reevaluate the conditions of your model. The parallel object procedures will work best when there are no extreme differences in the sizes of the faces and in the slopes of neighboring faces.

To derive the parallel of an open vector line, the line is not required to be planar. With planar lines, the new parallel line will be uniformly on one side of the original line. This is not always true with lines that make radical turns in 3D space. The system makes a special effort to keep the parallel line on one side of the original. However, the mathematical recognition of a “side” in 3D space may not coincide with how it appears on its projection. Thus at some points the parallel line may appear to jump on the other side of the line. Depending on the use of the parallel line, this may actually be desirable. If not, an adjustment will be necessary.

The parallel plane operation requires planar faces. Consequently, when it encounters non planar faces, they will be automatically triangulated, before the parallel plane procedures are applied. Also, given the adjustments that the parallel object procedures have to perform in cases where all the angles of an object are not orthogonal, there is always a high probability that the resulting parallel object will have non planar faces. When this is the case the non planar faces are triangulated. Consequently, it is quite typical that, even when all the faces of the original object are planar, the resulting object will have most of its faces triangulated, because they are not planar. The only cases where the new object is guaranteed to have non triangular faces and faces that are in one-to-one correspondence with the original object are when the latter is completely orthogonal or when all its points belong to no more than three faces, none of which are coplanar.

Given that only three non coplanar planes are guaranteed to intersect at a single point and that most frequently more than three planes are involved, some special calculations have to be performed to derive parallel planes, where the term is used in a broad sense. That is, the derived planes may not always be exactly parallel to the original planes, but will have an orientation that is very close to being parallel. The computations used are a mixture of intersection points, averages of intersection points, displacements along normals, and averages of such displacements.
The range of applications where parallel objects may be useful is rather wide. The following examples illustrate a few cases.

Figure 4.5.5.5 illustrates an “architectural” application. A volumetric composition is “sketched” as shown in (a). Then the volumes are unioned into a single volume (for the Union operation, see section 4.17.1), as shown in (b), and a double parallel object is derived, as shown in (c). Note that there is no other “easy” way to derive such a model. There are other ways in which parallel objects can be combined with Boolean operations to derive different results. For example, parallel objects may first be derived from the original volumes and then unioned. Note that in this example, none of the faces has been triangulated, and the new faces are in one-to-one correspondence with the original faces. This is because all of the original faces are planar, and all the points of the original object belong to three faces, which give a “clean” point of intersection and require no other adjustments.

![Figure 4.5.5.5: Deriving a parallel object from a volumetric composition.](image)
Figure 4.5.5.6 illustrates the derivation of parallel objects from meshes (see section 4.14). In (a), a double parallel object has been derived from the meshed surface shown to the left. The result is a meshed object with “width.” In (b) three single parallel objects have been derived, starting with the surface mesh shown to the left. The second was derived from the first and the third from the second. In both cases, the objects derived have all their faces triangulated. Since most of the points in the original meshes belong to four faces and many of the original faces are non planar, the resulting faces are non planar and thus have been triangulated.

Figure 4.5.5.7 illustrates the derivation of parallel objects from vector lines and how they can be used to derive a “ribbon” that unfolds freely in 3D space. A double parallel line (shown in (b)) is derived from a non planar vector line (shown in (a)). Some of the points of the resulting lines are moved in 3D space to “break” the uniformity of the parallelism (as shown in (c)). Then the original line and the two parallel lines are used as control lines to derive the ribbon shown in (d). How curved meshes can be generated from control lines is discussed in section 4.14.

**Figure 4.5.5.6:** Deriving parallel objects from meshed surfaces.

**Figure 4.5.5.7:** Deriving parallel lines from a vector line and using them to construct a ribbon.
4.5.6 Frames

The Frame tool can be used to construct pipes around the edges of an object. It can be used effectively to create frame structures, such as trusses or geodesic domes. A cylindrical pipe of a given radius is generated around each edge of an object, with the edge forming the axis of the cylinder. At each point, a sphere with the same radius is placed. All pipes and spheres are unioned together to form a solid object.

![Figure 4.5.6.1: Frame objects.](image)

Similar to other derivative tools, the frame object is a parametric object, whose parameters can be edited after it has been constructed. This is done in the usual manner through the Query dialog. An example of an object turned into a frame is shown in Figure 4.5.6.1.

This tool can be applied using either the postpick or the prepick selection method. Using the postpick method, the frame tool is applied by selecting an object. When using the prepick method, the Pick tool is used to preselect any number of objects and then, with the Frame tool active, the mouse is clicked anywhere in the project window, to execute the operation. The Frame tool can only be applied to facetted objects.

The single parameter that affects the Frame tool can be selected from the Frame Options dialog (Figure 4.5.6.2), which is invoked directly from the tool.

**Radius**: The value entered in this field determines the radius of the pipes and spheres.

![Figure 4.5.6.2: The Frame Options dialog.](image)
4.5.7 Projections of objects

Derivative objects can be created from the projections of other objects and their views, using the Projection tool, as follows:

**Projection**

This tool can be used to derive two types of projection objects. Which type will be derived depends on the option selected from the **Projection Options** dialog (Figure 4.5.7.1), which is invoked directly from the tool.

All types of projections are executed in the same fashion. With the postpick method, the Projection tool is activated first, then the object whose projection is to be derived is picked. The 2D projection is derived immediately and is stored as one or more objects. With the prepick method, any number of objects may be picked, followed by the selection of the Projection tool, followed by a click of the mouse anywhere in the graphics window away from any pickable entity.

The Projection tool frequently produces a number of separate shapes, which may be returned as independent surface objects or as a single object, as illustrated in Figure 4.5.7.2. This is controlled by selecting either **Create One Object Per Volume**, which is the default, or **Store All Volumes As One Object** in the **Status Of Objects** tab contained in the **Projection Options** dialog.

**Figure 4.5.7.1:** The **Projection Options** dialog.

**Figure 4.5.7.2:** (a) Projections of views generated as separate objects that can be moved independently. (b) Projections generated as a single object that can only be moved together.
**Projection Of View**: When this option is selected in the *Projection Options* dialog, 2D objects are derived from views by projecting the current view of an object on to the current reference plane, which can be any of the orthogonal planes, or any arbitrarily positioned plane.

To visualize what will be derived, consider that the current view is in essence a 2D shape that resulted by projecting a 3D view on the plane of the screen. This is the shape that will be derived by this operation. However, what you will see displayed on the screen may look different. That is, because the 2D shape derived is placed on the active reference plane, what you actually see is the 3D view of your newly generated shape. When you display an orthographic projection view, your shape will look the same as your original view. Examples are shown in Figure 4.5.7.3.

*Figure 4.5.7.3:* Deriving 2D projections of the view of a cuboid (a) on the XY and (b) on the ZX plane.

**Orthographic Projection**: When this option is selected in the *Projection Options* dialog, 2D objects are derived from the orthographic projections of other objects, by simply saving as an object the shape of the projection. Projections are relative to the active reference plane, which may be any of the orthogonal planes or any arbitrarily positioned plane (see Figure 4.5.7.4.).

*Figure 4.5.7.4:* Deriving 2D surface objects from the orthographic projections of a solid:
(a) The object whose orthographic projections will be derived.
(b) XY projection, (c) YZ projection, and (d) ZX projection.
4.5.8 Unfolding objects

Derivative objects can be created by unfolding other objects, as follows:

**Unfold**

Clicking with this tool on an object unfolds it and derives one or more new objects from the unfolded pattern(s) of the selected object. Options that affect the unfolded objects are selected from the **Unfold Options** dialog (Figure 4.5.8.1) that is invoked directly from this tool. Both faceted and developable smooth objects can be unfolded and, in the dialog, there are different sets of options for each type of object.

When using the postpick selection method, with the Unfold tool active and topological level set to Object, click on the object. The object is unfolded immediately. The prepick method can also be used, by preselecting any number of objects with the Pick tool, and then, with the Unfold tool active, clicking anywhere in the window.

Unfolded objects can be derived from any solid or surface object, but not from open lines. If an object that contains such lines is selected, an error message is issued. The unfolded projections are generated on the active reference plane, which can be any of the Cartesian planes or an arbitrarily positioned plane. Examples are shown in Figure 4.5.8.2.

![Figure 4.5.8.1: The Unfold Options dialog.](image)

![Figure 4.5.8.2: Unfolding faceted objects: (a) A cylinder on the XY plane. (b) A geodesic sphere on the ZX plane. (c) A revolved sphere on the YZ plane.](image)

With faceted objects, where you click to select an object to unfold is significant, and clicking on different segments produces different unfolding patterns, as illustrated in Figure 4.5.8.3. The segment you click on is placed on the horizontal axis of the reference plane, with one of its end points on the origin. Recall that X is the horizontal axis of the XY and the arbitrary plane, Y of the YZ, and Z of the ZX plane. The selected segment also becomes the basis of the unfolding. The click point is insignificant for smooth objects.
Unfolding facetted objects

With facetted objects, four sided faces, which are the most common, allow for a regularity when they are unfolded. That is, the system will first unfold all the adjacent faces in the vertical direction, and then in the horizontal direction. In both directions it stops unfolding when a face with fewer or more than four segments is encountered, or when the sequence is exhausted. As shown in Figure 4.5.8.2, objects that consist mostly of four-sided faces, such as the cylinder in (a), and the revolved sphere in (c), unfold into a pattern that shows a certain degree of regularity. In contrast, forms that have no four-sided faces, such as the geodesic sphere in (b), unfold into an irregular pattern.

Convex forms can generally be unfolded into a single continuous piece. In contrast, concave objects and objects with holes cannot be unfolded without some of their faces overlapping. *form·Z* will always split the holes of an object and unfold them as separate pieces. It will also split other overlapping pieces, such as those resulting from concave objects. The separated pieces may remain parts of a single object, or an independent object may be created from each piece. This is determined by whether **Create One Object Per Volume** (default) or **Store All Volumes As One Object** is selected from the **Status of Objects** dialog, invoked from the **Status of Objects**... button in the **Unfold Options** dialog.

The separate pieces may remain in the positions where they were unfolded, or they may be moved by the system to a position where they no longer overlap. This is determined by an option selected in the **Unfold Options** dialog.

**Reposition Overlapping Parts**: When this option is selected (default), the overlapping parts of an unfolded object will be moved to a new position where they do not overlap. These parts are arranged vertically and then horizontally. That is, rows of disjointed parts are formed above the first part, until all of the disjointed parts have been repositioned. The size of the margin between the disjointed parts is controlled by the following option.

**Margin Between Parts**: The value entered in this field determines the size of the margin placed between the disjointed parts of an unfolded object.

*Figure 4.5.8.3*: Different unfolding patterns result when clicking on different segments of a balloon.
The unfolding of an object with holes is illustrated in Figure 4.5.8.4. Note that each hole is unfolded as a separate piece. When selecting objects with holes for unfolding, it is necessary to click on a segment that is not a segment of a hole outline. When you click on a segment that is part of a hole outline, the system will automatically replace it with a segment of the outer outline of the same face, and use the latter as the basis for the unfolding.

The unfolding of a concave object is shown in Figure 4.5.8.5(a). The unfolded object consists of five parts that have been repositioned to eliminate the overlaps. If the unfolded pattern which is produced by the system is not satisfactory, then the user can guide the unfolding pattern by breaking the object into pieces through the use of the operations executed by the Separate tool. This is illustrated in Figure 4.5.8.5(b).

**Triangulate Non Planar Faces:** When this option is selected, any non planar face that may be contained by an object will be triangulated and the object will then be unfolded.

Only planar faces can be unfolded. If the **Triangulate Non Planar Faces** option is off and non planar faces are encountered, an error message will be posted and the object will not be unfolded.

When the unfolded object is intended to be used for the construction of a physical model, the task of reattaching the edges of adjacent faces may be facilitated by connectors which can be attached to the edges of the unfolded object. This is illustrated in Figure 4.5.8.6.

**Figure 4.5.8.4:** Unfolding a cuboid with two holes on XY.

**Figure 4.5.8.5:** Guiding the unfolding pattern:
(a) A concave object unfolded.
(b) Unfolded after it was separated into four parts.

**Figure 4.5.8.6:** (a) A cuboid (b) unfolded with split connectors.
(c) When refolding it, the connectors help reattach the edges. (d) Refolding it without connectors.
**Generate Connectors**: When this option is selected, connecting strips will be attached to all or some of the outer edges of the unfolded object. This option is off by default.

**Width**: The value entered in this field determines the width of the connecting strip.

**Type**: One of two types of connectors can be selected. The Split connectors consist of a strip extending from one of the endpoints of a segment to its midpoint. Such a “half” connector is placed on each of the corresponding segments. The Continuous connectors consist of a strip that extends between the two ends of a segment; such a strip is generated only for one of the two corresponding segments. Examples of the two types of connectors are shown in Figure 4.5.8.7.

![Figure 4.5.8.7](image)

**Figure 4.5.8.7**: Including connectors with unfolded objects: (a) **Split** and (b) **Continuous** connectors. Wide connectors with the **Skip Overlapping Connectors** option (c) off and (d) on.

**Skip Overlapping Connectors**: The widths of the connectors are automatically adjusted by the program to up to one third of the value entered in the **Width** field if there are tight angles between two edges where connectors are generated, or if the size of the edge is too small. But even after these adjustments are made, connectors may overlap other portions of the unfolded objects. While there are cases when you may wish to see all the connectors, even if they overlap, most frequently you might not. Selecting this option (which is on by default) instructs the program not to generate connectors that overlap with other parts of an unfolded object. Examples of unfolding an object with relatively wide connectors and this option on and off are shown in Figure 4.5.8.7.(c) and (d).
**Unfolding smooth objects**

The unfolding of smooth objects is primarily intended for objects of inflexible and rigid material, such as sheet or plate metals. Typical applications are likely to be found in ship building and the automotive industries.

Not all smooth objects can be unfolded, but only those whose surfaces consist of *developable* faces. Such are faces whose surfaces are *singly curved*, as opposed to *doubly curved*. These surfaces can be unfolded onto a plane without stretch or tear. They have the property that every point of the surface lies on a straight line segment (*ruling*) that lies entirely on the surface. The points of a ruling share the same tangent plane to the surface. Developable surfaces are either conical, cylindrical, or special cases of ruled surfaces.

Once an object is recognized to be smooth, it is checked to determine if its faces are developable. If they are, they are unfolded as smooth surfaces, as shown in Figures 4.5.8.8 and 4.5.8.9.

*Figure 4.5.8.8:* Unfolding a cone, whose surfaces are developable. Unfolded object is placed on XY reference plane.

*Figure 4.5.8.9:* Unfolding three unioned cylinders on the YZ reference plane.
Smooth **Objects With Non Developable Faces** are processed according to the options selected from the **Unfold Options** dialog (Figure 4.5.8.1).

**Skip**: When this option is on, the entire object is skipped and is not unfolded.

**Unfold As Facetted**: When this option is on, the smooth object is converted to a facetted object and is unfolded as such.

**Unfold Developable Faces Only**: When this object is on, if the object has some surfaces that are developable, they are unfolded and all the others are skipped.

Examples of these options are shown in Figures 4.5.8.10 and 4.5.8.11.

![Figure 4.5.8.10: An object with a non-developable surface is unfolded, using the Unfold As Facetted option.](image)

![Figure 4.5.8.11: An object with partially non-developable surfaces is unfolded, using the Unfold Developable Faces Only option.](image)
Smooth unfolding proceeds with one face at a time. Once all faces have been unfolded, they are positioned on the reference plane, after the originally coincident edges are matched and placed in close proximity. If this results in overlaps, they are resolved by repositioning the offending faces elsewhere on the reference plane. The faces of holes typically cause overlaps if placed in proximity to their neighboring faces and are positioned above the rest of the unfolded faces, after allowing a margin, which is determined by the program. Example of unfolding smooth objects with holes are shown in Figures 4.5.8.12 and 4.5.8.13.

By the placement scheme implemented, neighboring faces are placed in close proximity. Faces that share straight edges are most likely positioned along such edges. Faces with curved edges most likely touch at a single end point that they share.
**Labels**: When this option is on, text objects with numeric indices are placed along the edges of unfolded surfaces. There are exactly two text objects with the same number, which identify the two edges where the object was split. These labels are intended to make it easier to figure out neighboring edges, which is needed in order to reassemble the surfaces of an object. This is particularly useful when some of the unfolded surfaces have to be positioned away from neighboring faces to avoid overlaps. The labels are drawn along open edges. In most cases, they are derived from the edge indices of the original object. An example is shown in Figure 4.5.8.14.

**Size**: This pair of options determines the height of the labels.

**Automatic**: When this option is on, the height is computed as 1/10th of the shortest dimension of the (bounding) boxes of the unfolded faces. When the numeric field is selected instead, the number entered in the field determines the height of the labels.

**Placement**: The three options in this group specify the position of the labels relative to the edges of the unfolded surfaces. The offset of the label relative to the edge is internally computed and is approximately equal to the height of the label.

**Inside Face**: When this option is on, the labels are placed inside the edges of the surfaces.

**Outside Face**: With this on, the labels are placed outside the edges of the surfaces.

**On Segment**: With this option on, the label is placed on the edge of the surface.

**Grouping**: The options in this group determine whether the labels will be grouped and how.

**None**: When this option is on, no grouping occurs.

**All As A Group**: When this option is selected, all labels are placed in one group, whose name is assigned by the program.

**All On Layer**: When this option is on, all the labels are placed on the layer selected from the menu next to this option.
4.6 Terrain models

The Terrain Model tool is discussed in this section. Any one of three types of terrain models can be generated from a group of contour lines (or topographic map), as illustrated in Figure 4.6.0.1. The terrain model is trimmed to the borders of a site that is also identified when executing the operation.

A contour line is a surface object (2D shape) and can be open or closed. In reality, a contour line, being a “slice” of some terrain at a certain elevation, is always closed. However, by convention, when viewed relative to a bounded area (site), it may appear to be open. Typically, contour lines do not intersect each other. A site is also a surface object, which is required to be closed, and may contain holes. A site can also be a face or an outline selected from a solid object. The ends of open contour lines should cross the border of the site or be very close to it. The system will beep and issue an error message if a contour that violates this condition is encountered. Closed contour lines may or may not cross the site.

Figure 4.6.0.1: Terrain model types: (a) mesh, (b) triangulated mesh, (c) stepped, and (d) triangulated contour.
4.6.1 The Terrain Model tool and its dialog

Terrain Model

This tool can be applied using either the prepick or postpick method. When the prepick method is used, any number of contour lines are selected first, the Terrain Model tool is activated next, and then the site is picked. To use the postpick method, the contour lines should first be grouped using the Group tool. Recall that this is done by setting the topological level to Group, preselecting the contour lines, activating the Group tool, and then clicking the mouse anywhere on the screen. To execute the Terrain Model operator in the postpick mode, the operator is activated first, next the group of contours is selected (while the topological level is at Group), and then the site is selected. If the site is a surface object, when selecting it, the topological level may be set to Group or Object. If it is the face or the outline of a solid object, then the topological level should be set accordingly.

Both the contour lines and the site may be drawn within form•Z or they may be created by another application and imported into form•Z. Relatively large contour maps are best produced by tracing them on a digitizing tablet.

Contour maps are typically drawn on an XY plane and their z values represent the elevations of the contours. form•Z does not have such a restriction. Contours can be drawn on any of the Cartesian (orthogonal) planes or on any arbitrarily positioned plane. The only restrictions are that each contour should be planar and that all the contours and the site should lie on parallel planes.

The three types of models that the Terrain Model tool can generate are the mesh model (which may be triangulated), the stepped model, and the triangulated contour model, shown in Figure 4.6.0.1.

The types of terrain models as well as a number of optional parameters can be selected from the Terrain Model Options dialog, shown in Figure 4.6.1.1, which is invoked from the Terrain Model tool. It contains three tabs: Options, Mesh Options, and Status Of Objects. The options in the Mesh Options tab are as in the Mesh Options dialog (Figure 4.10.1.1) and are discussed in subsection 4.10.

![Figure 4.6.1.1: The Terrain Model Options dialog.](image)
4.6.2 Terrain modeling type and options

Mesh models

The parameters of the mesh are set through the **Mesh Options** dialog, shown in Figure 4.10.1.1 (section 4.10.1). The distance at which the mesh lines are placed is controlled by the values entered in the **X** and **Y** fields. For both, the default is 8 feet. When the **XYZ Lock** is selected, which is the default, the same value will automatically be applied to both directions. The **Mesh Options** dialog contains options for controlling the placement of the mesh grid relative to the site. They are discussed in section 4.10.1. It also contains a **Triangulate** option. When selected, the mesh tiles will also be subdivided into triangles after the rectangular mesh is generated. A triangulated model is shown in Figure 4.6.0.1(b).

After a mesh has been generated for the selected site, the elevations of its points are determined by interpolating the heights of the contour lines along the mesh or the fall lines. A fall line is the shortest distance between two consecutive contours. When using the fall line interpolation method, the fall line that passes through each of the points of the mesh is calculated, when such a fall line exists. When it does not, a horizontal interpolation is applied.

When interpolating along the mesh lines, the interpolation may be applied to a single direction (**X** or **Y**) or both (**X & Y**), which is selected from the **Terrain Model Options** dialog. The default is **X & Y**. **X** is the direction of the segment where the mouse is clicked when picking the site. Interpolating along fall lines generally produces the most accurate models. Interpolating in both directions of the mesh is more accurate than interpolating in one direction. However, for large scale models and relatively dense meshes, the more accurate the method the more computing time it requires. Consequently, one way interpolation may be used for previewing, and two way or fall line interpolation may be used for the final terrain model. Accuracy also depends on the shape, the direction, and the density of the contour lines, especially in low slope areas (areas surrounded mostly by a single contour line). These distinctions are illustrated in Figure 4.6.2.1.

![Figure 4.6.2.1](image-url): The effects of interpolating the elevations in (a) Y only, (b) X only, (c) both X and Y, and (d) along fall lines.
Stepped models

The Stepped Model option produces a terrain model that “steps” at the positions of the contour lines. This type of model is sometimes also called “cardboard model.” The derivation of this type of model is equivalent to progressively inserting perpendicular faces in the shape of the contour lines. The insert face procedures are actually employed for producing these models. An example is shown in Figure 4.6.0.1(c), at the beginning of this section.

The example in Figure 4.6.2.2 illustrates how terrain models can be derived by “inserting” contours to individual faces of a solid, such as a cube. For each of the three faces, an arbitrary reference plane was generated and the desirable contour lines were drawn on it. Then the Terrain Model operation was executed three times, once for each face. The Set New Interval option was selected and the default interval of 4'-0" was used (in metric the default is 1 m). In all cases, the contours were prepicked as objects, then the Terrain Model operator was activated, and, after setting the topological level to Face, the respective face was selected as the site.

The stepped model option inserts contour faces in a direction perpendicular to the site face. When the faces that neighbor the site face are not perpendicular to it, their planarity will most likely be disturbed when the terrain model is generated. This is illustrated in Figure 4.6.2.3.
Triangulated contour models

A **Triangulated Contour Model** is derived by inserting the contour lines to the site as vector lines, while preserving their elevations, and then triangulating the resulting faces (Figures 4.6.0.1(d) and 4.6.2.4). A one pass or a two pass triangulation may be applied. The first pass triangulates in between existing points only and may produce triangles all three points of which lie on the same contour line and thus have the same elevation (the triangle has no slope). The second pass inserts additional points, where required, calculates their elevation by applying fall line interpolations, and then uses these points to generate additional triangles. These new triangles have a slope. By default, the system applies a one pass triangulation. To apply a second pass triangulation, the **Triangulate Non Sloping Triangles** option should be selected. Selection of this option produces more accurate results but, for relatively large terrain models, it adds considerable computation time. The two options are illustrated in Figure 4.6.2.4.

![Figure 4.6.2.4:](image)

**Figure 4.6.2.4:** (a) A one pass triangulation and (b) a two pass triangulation (**Triangulate Non-Sloping Triangles** option selected).

The triangulated contour model and the triangulated meshed model are two distinctly different topological structures, as illustrated in Figure 4.6.2.5. In general, the mesh models do not contain points that are in one-to-one correspondence with the points of the contour lines. Instead, they consist of newly generated points that correspond to the corners of rectangular tiles. When the triangulate option is selected for the mesh models, these tiles are decomposed to two or more triangles. After the triangulation, the resulting model preserves its overall rectangular form. Contrary to the mesh models, the stepped and the triangulated contour models preserve the shape of the contour lines. That is, these models contain sequences of segments which are in one-to-one correspondence with the segments of the contour lines and have the same elevation. This characteristic is preserved after triangulation is applied.
The height options

The terrain models are always solid objects. This is naturally true when the terrain is generated on the face of an existing solid. It is also true when the site selected for the terrain model is a surface object. The system first generates a solid by extruding the site in a perpendicular direction. It then uses the top face of the new solid to generate the terrain model. The elevation (height) to which the surface site is extruded is determined by the value entered in the **Site (Starting) Height** field of the **Terrain Model Options** dialog. This value can not be 0. The default is 1 foot.

The elevations of the contour lines used for the generation of a terrain model can be treated in one of two ways. If from the **Contour Heights** group the **Use Existing** option is selected (default) the contour lines retain their current elevation values. If the **Set New: Interval** option is selected, the system will reassign elevation values using the contour interval value appearing in the numeric field of this option (the default value is 4 feet). For the **Use Existing** option the order in which the contours are selected is of no significance. However, for the **Set New: Interval** option the selection order is very significant, since the new elevations are calculated and assigned to the contours in the order in which they are selected. When two or more contours are required to have the same elevation, they need to be grouped, and then all the contours have to be selected with the topological level set to Group. This results in the same elevation value being assigned to all the contours in the same group.

The elevation value of a contour line is relative to its position in the world space. If a contour line is drawn on the XY plane, or a plane parallel to the XY, then the elevation value is given by the z coordinate. Similarly, if a contour line is drawn on or parallel to the YZ plane, then its elevation value is given by the x coordinate. If a contour is positioned on an arbitrary plane, then its elevation is its distance from the base plane. That is, the elevation is calculated as a length along a line perpendicular to the plane of the site.
**Smoothing and intersection checks**

For **Stepped** and **Triangulated Contour** objects, the option to **Smooth** the contour lines is also available. The default is off. The smoothing increment is given by the value entered in the **By Interval** field of this option. Default is 4 feet (or 1 m in metric). The effects of applying and not applying smoothing is illustrated in Figure 4.6.2.6.

The contour lines cannot intersect each other or themselves. Using intersecting contour lines for terrain models is not meaningful and produces inaccurate results. When attempting to generate stepped or triangulated contour models with intersecting contours, in most cases the system will be unable to complete the model. The **Terrain Model Options** dialog contains the option to ask the system to check for intersecting contours before starting the model generation process. If the **Precheck For Intersecting Contours** option is selected (default is off) and intersecting contours are found, the system will beep, issue an error message, and terminate the execution of the operation. This option should be turned on whenever there are doubts about the correctness of the contours, or whenever a topographic map is used for the first time. Since such a checking process may require considerable time for relatively large contour maps, it will typically be turned off for subsequent uses of the same topographic map.

*Figure 4.6.2.6:*
The effect of the **Smooth By Interval** option:
Stepped models (a) without and (b) with the option selected.
Triangulated contour models (c) without and (d) with the option.
4.6.3 Terrain modeling examples

The terrain modeling tool has been implemented in a manner which offers extensive flexibility. While it is mainly intended to be used for the modeling of landforms, given that terrain models can be derived in any direction and on any arbitrarily positioned plane, it can also be conveniently used as a general purpose modeling tool.

The different types of terrain models can be mixed to derive models such as that illustrated in Figure 4.6.3.1. It was derived in two steps. Given the site and the contour lines shown in 4.6.3.1(a), the contours in the lower left area of the site were first used to derive the stepped model shown in 4.6.3.1(b). Then the remaining contours were used to derive a mesh model on the upper right face of the previous model, resulting in the complete model shown in 4.6.3.1(c).

Figure 4.6.3.1: Mixing the stepped and mesh model types to derive a terrain model.
The previous example contains a flat road. Modeling a road, ramp, or stair climbing up a hill is a more challenging task, and is illustrated by the example in Figures 4.6.3.2 and 4.6.3.3.

Given the site in 4.6.3.2(a), the task is to connect points A and B by a ramp of no more than 15% slope. Assuming that the contour intervals are 2', it will take about 160' to cover the 24' elevation difference. A possible path is that shown in Figure 4.6.3.2(a). The exact shape of the path (ramp) is derived from the axial path (using the derivative 2D Enclosure operator with 6' width). The contours are also adjusted to provide smooth slopes along and around the path. In practice, when roads are constructed, contour lines are adjusted to become perpendicular to the road (where they cross it) and to be evenly distributed along the road. We have also drawn additional single segment lines to be used as contours for the derivation of the road. These “local” contour lines are assumed to be at 1' intervals, and two have been drawn for each of the original contour lines.

Using the Stepped Model option, and by picking the road shape as a contour and then the site, the model shown in Figure 4.6.3.3(a) is derived. The shape of the road has now been carved into a solid model derived from the site. Next, the Triangulated Contour Model option is used to generate the road model shown in Figure 4.6.3.3(b). After selecting the Set New Interval option and the interval value is set to 1', the local contour lines are preselected, then the Terrain Model tool is activated, the topological level is set to Face, and the face of the road is selected. Note that the top of the road model has been smoothly triangulated which will result in a smoothly sloping surface when the model is rendered.
The mesh model of the land around the path is generated in two steps. The **Set New Interval** option continues to be selected, but the intervals value is changed to 2' and the **Mesh Model** type is selected. The contour lines are prepicked, then the Terrain Model tool is activated, and (with topological level at Face) the face to the left of the road is selected. The result should be as shown in Figure 4.6.3.4(a). If the contour lines have been ghosted, after the execution of the previous step, unghost them and repeat the operation by selecting the face to the right of the road. The result is as in 4.6.3.4(b).

**Figure 4.6.3.4:** Mesh models generated on (a) the left and (b) the right of the road.

A variation of the previous example, where a free form stair is created next to the ramp, is shown in Figure 4.6.3.5. Similar techniques have been used. The shape of the stair is drawn and is used to generate a stepped model which “carves” the shape of the stair into the face to the right of the ramp. Then local contour lines are drawn (one per step) and another stepped model is generated by preselecting these local contour lines, activating the Terrain Model tool, and selecting the face of the stair. The meshed areas to the right and left of the stair are then generated as before.

**Figure 4.6.3.5:** A free-form stair generated next to the ramp.
For large topographic maps, it may sometimes be appropriate to subdivide the site and derive the terrain model in pieces, as illustrated in Figure 4.6.3.6. This may make it easier to work on parts of the model, whenever additional operations need to be applied to it. In general, terrain models may easily become very large and the user should be careful to properly set the optional parameters. There will certainly be cases when dense meshes and dense contour lines are necessary. However, most of the time, using larger parametric values will produce equally satisfactory results. When lower (denser) values are necessary, it may still be appropriate to first generate “previews” of models with higher parameters before generating the final one. Such models are generated more quickly, and allow the user to evaluate alternatives before deciding on the final setup.
4.7 Objects of revolution

The following derivative object generating tools are discussed in this section:

- Revolved Object
- Helix
- Screw/Bolt and Detailed Screw/Bolt
- Gear

The objects discussed in this and the next two sections are parametric derivative objects. They are constructed from other source and path shapes and they are internally stored with the parameters that generated them. This makes it easy to revisit them and change them after their initial generation, which is done with the same tool used to create them. For the revolved objects, the Edit Controls tool can also be used for editing the revolution angle. Many of these derivatives also have the option to be created as nurbz objects. When this option is applied and they become nurbz, both the Edit Controls and Edit Surface tools are used to revisit the respective parameters.

In their general form, objects of revolution are derived by sweeping a source about a circular path. The source may be an open or closed shape, or an implicit point, producing solid/surface or wire objects, respectively.

The circular path itself may be implicit, which is the typical case, or an explicitly selected shape. When implicit, the path typically revolves around an axis line, which is selected. As the source is swept about an axis, it may retain a constant position relative to the length of the axis, or it may also be moving by a certain increment at each step. The former results in the type of objects of revolution that are also known as lathed. The latter results in helical forms.

When an axis needs to be selected for revolving a shape about it, that axis may be one of the Cartesian world axes, an axis of a reference plane, or any segment of an object. This is illustrated in Figure 4.7.0.1 and applies the same to all types of revolved objects. Note that, if different types of axes coincide, precedence rules will select one in the following order: world axis, reference axis, object segment. For example, if a line crosses an axis of a reference plane and you click on the point where they cross, the plane axis will be picked as an axis for the revolution.

Each of the objects of revolution available generates a number of variations, as discussed in the remainder of this section. Another object that is essentially an object of revolution is the spiral stair, discussed in the next section.
4.7.1 Revolved objects

Also known as lathed objects, revolved objects are typically created by rotating a shape about an axis, using the following tool:

![Revolved Object]

To derive a revolved or lathed object, with the Revolved Object tool selected, first click on the source shape, and then on the axis of revolution. The source shape may be closed or open. When closed, a “wheel” type object is produced (Figure 4.7.1.1). When open and its endpoints lie on or sufficiently close to the axis of revolution, a “ball” type object is produced (Figure 4.7.1.2). When the source shape is open and away from the axis of revolution, a surface object is generated. An example is shown in Figure 4.7.1.3.

Note that the same source shape can be used to derive different lathed objects, by selecting different axes of revolution. This is illustrated in Figure 4.7.1.4.
Several options are available in the **Revolve Options** dialog (Figure 4.7.1.5), which contains three tabs: **Options**, **Display Resolution**, and **Status Of Objects**. The options in the latter two tabs are as discussed in sections 4.1.1 and 4.5, respectively.

**The Options tab**

The options in this tab affect the shape of the revolution and the type of object to be created. Revolved objects may be parametric at two different levels. They are initially parametric as objects of revolution and can be edited to change these parameters. They may also be parametric as nurbz objects, if the respective option is selected in the dialog. They can not be both at the same time. That is, if they become nurbz, they lose their access to the revolution parameters. They may also be converted to plain facetted objects and lose both levels of parameters, which can be done using the Convert tool.

**Model Type**: These options determine the type of the model to be generated, as follows:

- **Facetted**: When this option is on (default), a model whose surface is approximated by planar polygons is generated.

- **Smooth**: With this option on, a model bounded by smoothly curved surfaces is generated. These surfaces can be simple revolved surfaces or NURBS surfaces (called nurbz in form•Z).

**Revolution Angle**: Revolved objects may be derived through a complete (360°) or partial rotation of the source shape, as illustrated in Figure 4.7.1.6. This is determined by the value entered in this field. The default is 360° or a complete revolution.

**Direction**: This pop up menu contains two items, **Clockwise** and **Counter Clockwise**. They set the direction of the revolution.

Depending on whether **Facetted** or **Smooth** objects are to be generated, one of the following two groups of options applies:

**Figure 4.7.1.5**: (a) The Revolve Options dialog with the Options tab open and the Facetted Options shown and (b) the Smooth Options.

**Figure 4.7.1.6**: A three quarter sphere and ring derived by a partial revolution (270°) of an arc and a diamond.
Generating facetted revolved objects

**Facetted Options:** These options are available when **Facetted** is on for **Model Type**, which results in a polygonal rather than a smooth object.

**# Of Steps:** This parameter controls the density or resolution of the revolved objects. The value entered in this field determines how many rotational steps will be executed. Whether this number refers to a complete 360° revolution or to the partial revolution is controlled by the following two options. Default for this parameter is 16.

**Total:** With this option on, **# Of Steps** is applied to the complete revolution. For example, if **# Of Steps** is set to 16 and **Revolution Angle** is set to 270°, the resulting object will be generated using 12 steps, which is 3/4 of 16.

**Per Angle:** When this option is selected, the value entered in the **# Of Steps** field will be applied to the partial revolution. These two options are illustrated in Figure 4.7.1.7.

**Join Adjacent Coplanar Faces:** Segments of the source shape that are perpendicular to the axis generate coplanar faces when they are revolved. When this option is selected, a single face corresponding to these segments is generated. This is equivalent to joining the coplanar faces after their initial generation. Default is off. This option is illustrated in Figure 4.7.1.8.

**Triangulate:** When this option is selected, the system automatically triangulates any non-planar faces generated by the operation. Default is off.

**Adjust Open Ends To Axis:** When open source shapes have their end points exactly on the axis of revolution, solid objects are generated. When one or both of the end points are away from the axis, surface objects are generated. When this option is on, the ends of open source shapes will always be adjusted to lie on the axis, which will always result in solid revolved objects.
Closed source shapes may not intersect the axis of revolution. If they do, the system beeps, issues an error message, and cancels the operation. The first and last segments of open source shapes may cross the axis of revolution. When they do, they are automatically trimmed at the point where they cross the axis. Other segments of open source shapes may not cross the axis. The non intersection restriction applies to true intersections only. That is, the check is applied only when the source shape is planar, and the axis lies on the same plane, which is recommended as discussed below.

Proper lathed objects are generated only when the source shape and the axis of revolution lie on the same plane. The system will derive a lathed object even when these conditions are not met. However, the result may be a more or less distorted solid that may not be accepted by the tools that require well formed and/or planar objects. Such cases are illustrated in Figures 4.7.1.9 and 4.7.1.10.

Lathed objects may be generated using either the postpick or the prepick method.

With the postpick method, the Revolved Object tool is activated first, the source shape is picked, and then the axis of revolution is selected. The object is derived immediately.

With the prepick method, any number of source shapes may be picked first using the Pick tool, followed by the selection of the Revolved Object tool, followed by the selection of the axis of revolution. The generation process is initiated as soon as the axis is selected. The prepick method makes it possible to generate composite objects of revolution with a single execution of the operation. Such an example is illustrated in Figure 4.7.1.11.
Lathed objects can be generated in ways that do not necessarily have the appearance of a lathed object. Some examples are shown in Figure 4.7.1.12. In addition, lathed objects can be generated in ways that attach them to the original objects, as illustrated in Figures 4.7.1.13 and 4.7.1.14.

**Figure 4.7.1.12:** (a) Drawing a set of triangles and lines on the YZ reference plane. (b), (c) Picking the triangles as source shapes, and the lines as axes of revolution in two different ways produces the shown objects, where the pieces have also been unioned into single objects.

**Figure 4.7.1.13:** With Snap to Face selected (window tool), a hexagon is drawn on the front face of the cuboid. Then, with Revolution Angle set to 270° (in Revolve Options dialog) and with the Revolved Object tool active, the mouse is clicked on points 1 and 2. The hexagon is revolved around the vertical front edge of the cuboid.

**Figure 4.7.1.14:** Using Snap to Point, the side face of the cube is partially traced, to draw a shape. Then, with the Revolved Object tool selected and Revolution Angle set to 270°, the mouse is clicked on points 1 and 2. The shape drawn on the face is revolved around the edge. The cuboid and the object of revolution are unioned, as shown.
Generating smooth revolved objects

**Smooth Options:** These options are available when **Smooth** is on for **Model Type**, which results in a smooth rather than a polygonal object. Such objects may also be derived through a complete (360°) or a partial rotational sweep of the source shape. Smooth revolved objects can be of two types, as follows:

**Construct As Smooth Revolve:** When this option is on, the revolved object is bounded by surfaces that are generated as curves of revolution.

**Construct As Nurbz:** When this option is selected, the revolved object is generated as a nurbz object (Figure 4.7.1.15(b)).

**Length (U) Degree:** This parameter controls the degree of the nurbz curve that is derived from the source shape of the revolution. It must be greater than zero and less than the number of points in the source shape. The default is 2.

Note that the **depth** parameter, which controls the curvature around the axis of revolution, is fixed to 2 and can not be changed. A depth degree of 2 is the only value that generates circles and therefore the only logical value for revolutions.

**Common options**

The remaining options in the **Revolve Options** dialog apply to both **Facetted** and **Smooth** models, as follows:

**Cap Start, Cap End:** These options determine how the ends of partially revolved objects (less then 360°) are treated. When on, a cap is generated to close the corresponding end. When both ends are capped, a solid object is defined. This is illustrated in Figure 4.7.1.16.

For faceted and smooth revolved objects, a cap is a planar face that is attached to the end of the revolved object. For smooth nurbz objects a cap is created by inserting extra control points at the ends of the nurbz that are converging towards the centroid of the cap. This may not work well when the ends are irregular shapes whose centroids may even be outside their boundaries. In these cases a warning message is posted before generating the caps.

**Plain Object:** By default, the revolved objects are parametric and can be edited after their initial generation to change their parameters and shape. When this option is on, a revolved object will be generated as a non-parametric or plain object, which may still be either faceted or smooth, but does not carry its revolution parameters.
Previewing and editing revolved objects

Revolved objects can be previewed before their generation is finalized, which is done when the **Edit** option is on in the **Revolve Options** dialog (see below). Revolved objects can also be edited after their initial generation, again using the Revolved Object tool. They can also be edited using the Edit Controls and the Query tools.

**Edit**: When this option is on at the time the source and the axis for the revolved object are picked, the **Revolve Edit** dialog shown in Figure 4.7.1.17 is invoked, where the revolved object can be previewed. This works as all the preview windows in **form-Z**.

**Adjust To New Parameters**: When this option is on, the settings currently selected in the **Revolve Options** dialog will be applied to the object. This option can be on with or without **Edit** on. Note, however, that both cannot be off at the same time.

The revolved object is a parametric object. That is, it is saved with the parameters that generated it and can be later edited to change it. This can be done using any one of three tools, namely, the Revolved Object, the Edit Controls, and the Query tool. All three tools are applied by clicking on the object you wish to edit.

When using the Revolved Object tool, depending on whether **Edit** or **Adjust To New Parameters** is on, the **Revolve Edit** dialog is invoked or its parameters are adjusted directly. Except for these options, the **Revolve Edit** dialog contains the same options as the **Revolve Options** dialog and one more:

![Figure 4.7.1.17: The Revolve Edit dialog with (a) Facetted and (b) Smooth on.](image)
**Source Options...**: Clicking on this button invokes the **Edit** dialog for the type of source that was used for the revolved object. For example, for the revolved object shown in Figure 4.7.1.17 the dialog in Figure 4.7.1.18 is invoked. Through this dialog parameters of the source can be changed.

When editing revolved objects with the Edit Controls tool, the object is entered into an edit mode and you can graphically change its degree of revolution as well as the shape of its source. The controls that are available for reshaping the source depend on the type of shape the source is. A revolved object in edit mode is shown in Figure 4.7.1.19.

**Figure 4.7.1.18:** The **Composite Curve Edit** dialog.

**Figure 4.7.1.19:** Editing an object with Edit Controls: (a) original object and controls, (b) adjusting the angle of revolution, and (c) adjusting the shape of the source.
4.7.2 Helixes

Revolving a source shape about an axis and incrementally moving it along the length of the axis produces a helix. In form·Z, helixes can be solid, surface, or wire objects, all generated with the same tool.

Helix

This tool is used to generate three types of helixes, which are selected from the Helix Options dialog (Figure 4.7.2.1) that is invoked from this tool. It contains the three typical tabs: Options, Display Resolution, and Status of Objects. The latter two are discussed in sections 4.1.1 and 4.5, respectively.

The Options tab

Helixes can be Facetted or Smooth types of models, which is selected from the Helix Options dialog, and works as for the revolved objects.

The three types of helixes that can be generated are Solid/Surface Helix, Wire Helix About Axis, and Wire Helix Along Path. All types are discussed in this section.

Cap Start, Cap End, Plain Object, Edit, Adjust To New Parameters: These options are as for the revolved objects.

The helixes are parametric objects that can be edited after their initial generation. This is done using one of three tools, namely, Helix, Edit Controls, and Query. They all work as for the revolved objects. When the Solid/Surface Helix Edit dialog is open and the source is a parametric shape, it can be edited by clicking on the Source Options... button, as for the revolved objects.
Solid and surface helixes

These types of helixes are generated when **Solid/Surface Helix** is selected in the **Helix** dialog. They are revolved objects generated from a source shape and an axis of revolution. The source can be any open or closed surface object, a 2D enclosure, a segment, an outline, or a face of an existing object. The axis can be any of the Cartesian (world space) axes, or an axis of an arbitrary reference plane, or any segment of an existing object, including a single segment object drawn specifically to use as an axis of revolution.

When a closed shape is selected as a source shape, a solid helix is generated. When an open shape is selected, a surface object (one sided surface) is generated. Examples of solid and surface helixes are shown in Figure 4.7.2.2.

The solid and surface helixes are generated as are the lathed objects, which is by rotating the source shape about the axis of revolution, except that, in addition, at each step the source shape is also moved along the axis. While a lathed object is revolved by a maximum of 360° (one cycle), helixes typically consist of more than one cycle.

The size of a helix can be measured by its length along the axis of revolution; its number of cycles, where each cycle represents a revolution of 360°; by the total number of revolution steps; or by the total angle of revolution.

---

**# Of Cycles, Angle:** These two parameters are correlated; when one is changed the other is adjusted automatically. To find the angle, the number of cycles is multiplied by 360, and vice versa. One cycle (or 360°) represents one complete revolution of the helix.

**Length:** The value entered in this field determines the length of a helix along the axis of revolution. The exact use of this value depends on the selection of one of the items in the pop up menu next to it, as follows:

**Total/Per Cycle:** These two items can be selected from the pop out menu next to **Length**. The former causes **Length** to be used for determining the complete length of the helix. The latter applies the value in the **Length** field to the “jump” between cycles. The total length of a helix is given by multiplying the **Length** value by the number of cycles.

**Direction:** The two items in this pop up menu (**Clockwise** and **Counter Clockwise**) determine the direction of the helix.
Depending on whether Facetted or Smooth was selected for Model Type, the options in one of the following two groups apply:

**Facetted Options**: When this option is on, a parametric facetted helix is generated, in accordance to the following parameters:

- **# Of Steps**: This value determines the resolution of facetted helixes. That is, the number entered in this field determines by how many increments the source will be rotated Per Cycle or Total length of the helix. The desired item is selected from the pop up menu next to this option.

- **Triangulate**: When this option is on, the faces of the helix are triangulated.

**Smooth Options**: When this option is selected, either a smooth helix or a nurbz helix is generated. The parameters in this group of options are as for the revolved objects (see previous section).
**Transformation:** The *path* of a solid/surface helix is the invisible helical line that corresponds to the centroid of the source shape as it is revolved about the axis. Either or both the path and the source shape can be scaled when a solid/surface helix is generated. The source shape can also be rotated as it is revolved about the axis. These parameters are selected from this group of options in the **Helix Options** dialog.

**Scale Path:** When this option is selected, the path of a helix can be scaled either along its *width*, along its *length*, or both.

**Width:** The value in this field is the scaling factor that is applied to the width (radius) of the helix path. Scaling factors less than 1.0 decrease, and factors greater than 1.0 increase the radius of the helix. The value entered in this field represents the scaling factor that is applied to the radius used at the very end of the helix, rather than the scaling factor per step.

**Length:** The value in this field is the scaling factor that is applied to the length of the helix path. This scaling factor produces the distance at which the last point of the helix is placed. Note that scaling factors less than 0.5 (50%) cause the helix to fold back toward its starting point, which is generally not desirable. Consequently, except when special effects are sought, the recommendation is against using *Length* scaling factors smaller than 0.6. Applying scaling factors to the path of a solid/surface helix is illustrated in Figure 4.7.2.3.

**Figure 4.7.2.3:** Solid helixes generated from a 10-sided polygon with *Length* = 10', using the **By Cycles** method, **# Of Cycles** = 3, and **# Of Steps** = 16.  
(a) No scaling applied.  (b) Path scaled with *Width* = 0.25.  (c) Path scaled with *Length* = 0.6.  
(d) Path scaled with *Width* = 0.25 and *Length* = 0.6.
**Scale Source:** When this option is selected, the scaling factors entered in the \( X \) and \( Y \) fields next to it are applied to the source shape as it is revolved. \( X \) is the horizontal direction, and corresponds to the direction of the \( X \) axis when the source shape is drawn on the \( XY \) Cartesian plane or on an arbitrary reference plane. Similarly, \( Y \) is the vertical direction, and corresponds to the \( Y \) axis when the source shape is drawn on the \( XY \) plane or an arbitrary reference plane. When the source shape is drawn on the \( YZ \) or \( ZX \) plane, then \( X, Y \) correspond to \( Y, Z \), and \( Z, X \), respectively. Scaling factors can be applied to either or both the \( X \) and \( Y \) directions. The scaling factors entered in the \( X \) and \( Y \) fields represent the factors that will be applied to the last position of the source (rather than scaling factors applied per step). When the lock option is on, the same scaling factor is applied to both the \( X \) and \( Y \) directions. The per step of revolution scaling factors are calculated automatically by the system. Examples of solid/surface helixes with scaling factors applied to the source shape are shown in Figure 4.7.2.4(b) and (d).

**Rotate Source:** When this option is selected, the source shape is also rotated on its own plane as it is revolved about the axis of revolution. The rotation angle is determined by the value entered in the field next to it. Whether the rotation angle is applied to each step of revolution or to the whole revolution is determined by the item selected from the pop up menu next to this option. There are two items: **Total** and **Per Step**. Note, however, that these apply only when a Facetted helix is generated. For Nurbz, this pop up menu is dimmed, and the rotation value is always interpreted as **Total**.

The **Total/Per Step** pop up menu is available in both the Helix and the Helix Edit dialogs. In these dialogs, switching from one item to the other behaves differently. In the Helix Edit dialog, when switching from **Total** to **Per Step**, or vice versa, the rotation value is also adjusted so that its effect as previewed in the dialog remains the same. In the Helix dialog, no automatic adjustment to the rotation value is made.

**At 1st Point:** This option is only available when Rotate Source is on. When At 1st Point is off, which is the default, the source shape is rotated about its centroid. When it is on, the source shape is rotated about its first point.

Helixes with rotated source shapes are illustrated in Figure 4.7.2.4(c) and (d). Helixes where both the path and the source are scaled are shown in Figure 4.7.2.5.

**Figure 4.7.2.4:** Solid helixes generated from a source with same parameters as Figure 4.7.2.3.
(a) No scaling applied.
(b) Source scaled with \( X \) and \( Y \) factors = 0.1.
(c) Source rotated \( By \) 22.5°.
(d) Source scaled and rotated by previous values.

**Figure 4.7.2.5:** (a) A surface helix and (b) a solid helix generated by scaling the path and scaling and rotating the source.
Wire helixes about an axis

These are objects of revolution that are generated by revolving a point about a helix, and moving the point as it is revolved. They are also called axial wire helixes. These helixes are always open surface objects (vector lines). Their parametric options are selected from the Helix dialog. An axial helix is generated by a single mouse click which is on the axis of revolution. The axis can be any of the world or reference plane axes, or a segment of a previously drawn object. Where the mouse is clicked is significant and determines the position of the starting point of the helix. The axial wire helix also uses a radius value entered in the Helix dialog to determine the distance at which the revolved point will be placed.

The axial helix is always constructed by starting at the point where the mouse is clicked, and moving in a direction which is determined by the direction of the axis line. When the axis is one of the Cartesian axes, the helix is always generated in the direction of the positive axis.

**Wire Helix Radius**: The value entered in this field determines the distance from the axis of revolution where the point to be revolved is placed.

**By Steps, By Cycles, By Angle**: These are three methods by which the size parameters of a wire helix can be defined, and work as for the solid/surface helix.

**Length: Total, Per Cycle, Per Step**: As for the solid/surface helix.

**Scale Path: Width, Length**: As for the solid/surface helix.

**Scale Source, Rotate Source**: These options do not apply, and are dimmed.

Axial wire helixes and the effects of different combinations of parameters are illustrated in Figure 4.7.2.6. Note that, as with the solid/surface helixes, when a scaling factor less than 0.5 is used for the Length of the path, the helix folding and turns inwards. This is illustrated in Figure 4.7.2.7.

![Figure 4.7.2.6: Axial wire helixes: (a) No scaling is applied. The path is scaled (b) by Width factor = 0.1, (c) by Length factor = 0.75, and (d) by Width factor = 0.1 and Length factor = 0.75.

![Figure 4.7.2.7: Scaling the Length of the path by a factor less than 0.5 causes the helix to bend inward.](image-url)
Wire helixes along a path

In contrast to the axial wire helixes that are always generated about a straight line, these are wire helixes that are generated following an arbitrarily shaped path, which can be open or closed. As a result, these helixes can be open or closed surface objects (vector lines).

A wire helix along a path is generated by selecting (clicking on) the path object. Where you click is insignificant, since the along path helix is generated from one end of the path to the other. This helix uses the **Wire Helix Radius** entered in the **Helix** dialog, but ignores the **Length** parameter.

**Wire Helix Radius:** **By Steps, By Cycle, By Angle:** As for the axial wire helix.

**Length:** This option does not apply to the wire helix along path and is dimmed. The length of this helix is determined by the length of the path object selected for the operation.

**Scale Path:** When this option is selected and the path is an open shape, the width (radius) of the along path helix will be scaled as the helix point is revolved, using the value entered in the **Width** field. As for the other types of helixes, this scaling factor applies to the position of the last point of the helix, and the scaling factors for the points between the two ends are automatically calculated by the system. The **Length** scaling factor does not apply to the helix along path, since, by design, it must extend from one end of the path to the other. Neither scaling factor applies to closed paths.

Examples of **Wire Helixes Along Path** and the effect of different combinations of parameters are illustrated in Figure 4.7.2.8. Helixes along path can easily be generated about non planar paths, as shown in Figure 4.7.2.9. These helixes can then themselves be used as paths for the Sweep tool (see subsection 4.9.1) to construct objects such as telephone cables, as the example in Figure 4.7.2.10 illustrates.

*Figure 4.7.2.8:* Wire helixes along paths, generated with **Wire Helix Radius** = 2', using the **By Cycles** method, # Of Cycles = 40, and # Of Steps = 16.

(a), (b) Open paths and no scaling. (c), (d) Open paths and scaling by Width factor = 0.1.
(e), (f) Closed paths, which can not be scaled.
Figure 4.7.2.9: Wire helixes generated along non-planar paths: (a) Open path and (b) closed path.

Figure 4.7.2.10: A telephone cable created by sweeping a polygon along a path, which was created as a helix along a path.
4.7.3 Screws and bolts

The screws and bolts are composite helical objects of revolution. They consist of three parts: the **body** or **thread**; the **neck** or **grip**; and the **head**. One end of the screw body may have a **tip**, where the threads are scaled, resulting in a pointed end. The parts of a screw/bolt are shown in Figure 4.7.3.1.

All screws/bolts must have a body or they can not be generated. The other parts, namely the neck, the head, and the tip, are optional, and may or may not exist. The same algorithm is used for the construction of the threads for both screws and bolts. Whether a screw or a bolt is generated depends on the shape produced by the parameters selected. Screws typically have a tip and a conic head. Bolts typically have flat ends or very short tips, and round heads. Both can have either slotted or Phillips grooves on their heads. Needless to say, there are many other variations of screws and bolts.

Screws that consist of only a body and its threads are useful for attaching to objects such as bottles or other types of containers, after drilling a hole through them, as illustrated in Figure 4.7.3.2. While there is no direct way to generate nuts, they can be easily constructed by differencing (Boolean Difference) the threads of a bolt from an object with the appropriate shape for a nut. Variations of screws, bolts, and nuts are shown in Figure 4.7.3.3.

**form-Z** offers two tools for generating screws and bolts: the Screw/Bolt and the Detailed Screw/Bolt tool. The former generates faceted objects only, while the latter both smooth and facetted objects. The former are generally quicker to execute, while the latter require more time, due to their more refined details.
Screw/Bolt

Clicking with this tool on an axis generates a screw/bolt. The axis can be one of the coordinate axes or a segment of an object, as for the other revolved objects. The direction in which the screw/bolt is generated is determined by the direction of the axis. Where you click when selecting the axis is significant as it determines the position of the head of the screw/bolt.

The parameters for the screws and bolts are selected from the Screw/Bolt Options dialog (Figure 4.7.3.4) which is invoked directly from the Screw/Bolt tool. Note that the dialog includes some graphics that illustrate the different parts of the screw/bolt.

The three major parts of a screw (Head, Grip, and Thread) require Diameter and Length parameters, which are entered in the respective fields. The dialog also contains a non-editable information field labeled Total Length. The value in this field displays the total length of the screw/bolt, which is the sum of the lengths of the thread, grip, and head. It is updated every time one of the Length values (except for Tip) is changed.

The Tip requires only a Length parameter. The value entered for the tip length represents the portion of the thread that will be scaled to produce a pointed end. The length of the tip cannot be greater than the length of the thread.

One of five different shapes of heads can be attached to a screw/bolt: round conic, semi-spherical, round flat with round edges, hexagonal, or square head. The desired shape is applied by selecting the respective item in the popup menu. Any shape of head can be Plain, or have a Slotted or Phillips groove. Note that the grooves are generated by applying a Boolean Difference to the head.

Threads: The thread characteristics of a screw/bolt are specified using one of two methods.

By # Of Cycles: When this method is selected, the screw/bolt will be constructed with the number of cycles shown in its field. The distance between the threads will be determined by dividing the length of the body of the screw/bolt by the number of cycles.

By Distance: When this method is selected, the value entered in its field determines the distance between the threads. The total number of cycles is determined by dividing the length of the body of the screw by the thread distance.
**Shaft:** The value entered in this field is a percentage representing the portion of the space between two threads that will be occupied by the shaft. This value should be less than 100.

**Depth:** The value entered in this field determines the depth of the thread.

**Points Per Cycle:** The value in this field represents the resolution of the screw/bolt. That is, this parameter determines the number of revolution steps per cycle (360°).

**Left Handed:** By default screws and bolts are right handed (clockwise). When this option is on, they generate left handed (counter clockwise).

Screws/bolts are parametric objects that can be edited and adjusted after their initial generation. This is done by reusing the Screw/Bolt tool, or using the Query tool, through a preview dialog (Figure 4.7.3.5), as for the other revolved objects. However, the option to generate them as plain objects is also available.

**Plain Object:** When this option is off, which is the default, the screw/bolt is generated as a parametric object. When it is on, a plain facetted object is created.

As with the other revolved objects, a preview dialog is available for the screws/bolts, where parameters can be changed and previewed before the final generation of the object. The same dialog is used when editing a screw/bolt.

**Edit:** When this option is on at the time an axis is selected for the derivation of a bolt/screw, the preview dialog shown in Figure 4.7.3.5 is invoked, where the screw/bolt can be previewed. This dialog works as all the preview windows in form-Z.

**Adjust To New Parameters:** When this option is on, the settings currently selected in the Screw/Bolt Options dialog will be applied to the object. This option can be on with or without Edit being on. Note however, that both cannot be off at the same time.
Detailed Screw/Bolt

Clicking with this tool on an axis of a reference plane or on a line generates a screw or bolt positioned along the selected axis or line. Clicking on a reference plane generates a screw or bolt perpendicular to the reference plane.

The parameters of a detailed screw or bolt are set in the Detailed Screw/Bolt Options dialog, shown in Figure 4.7.3.6, which can be invoked directly from the tool. The dialog contains two tabs: Options and Resolution. The latter is as the same tab in other dialogs. The options in the former tab are as follows:

Model Type: One of two options can be selected from this group, Facetted or Smooth, which determines the type of model to be generated.

Dimensions: Selecting any number of the following items results in the respective part being included in the screw or bolt to be generated. The sizes of the various parts of a screw or bolt are also entered in the fields of these options. The fields are arranged in two columns. The diameter of the respective part is entered in the field on the left column and the length of the part in the field on the right column.

On the right of the right column are radio buttons, one of which can be selected, indicating that its length will be calculated from the values entered for the other parts.

Head: When this option is on, the screw or bolt will have a head. Its diameter and height are entered in the fields next to this option.

Nominal: This is the length of the bolt or screw beneath the working surface of the material being joined. This will be the sum of the length of the thread and grip, for bolts. For screws, it also includes any counter sunk depth of the head. There is always a nominal part and its length can be set in the respective field, whenever its radio button is not on.

Grip: This is the unthreaded section of the bolt/screw between the head and the threaded body. Its Diameter and Length may be specified independently of other sections of the screw/bolt.
**Thread**: This is the threaded section of the screw or bolt and its **Diameter** and **Length** may be specified.

**Tip**: This is a threaded section of the barrel of the bolt/screw that is intended to engage in softer materials, either by self-tapping a hole or fitting into a tapered pilot hole. At the top of the barrel the diameter is as specified for the thread and it changes linearly toward zero at the tip.

**Total**: This non editable field displays the sum of the grip and thread, as well as any height of the head that protrudes above the surface of the material being joined.

**Head**: The head of a screw/bolt is commonly used to provide some mechanical connection to fasten it. A number of common head shapes for bolts/screws can be selected from this popup menu. They are **Flat**, **Round**, **Oval**, **Cheese**, **Pan**, **Button**, **Hex**, **Square**, and **None**. The **Diameter** and **Length** of most of these may be specified independently. Exception are **Round** and **Oval**, whose length depends on the diameter. The available head shapes are illustrated in Figure 4.7.3.7.

![Figure 4.7.3.7: Examples of heads.](image)

**Drive**: This is a recessed shape in the head that allows some mechanism, such as a screwdriver, to be engaged with the head. The desired type is selected from the popup menu, which contains **Slot**, **Phillips**, **Allen**, and **Square**. Values for **Depth** and **Width** complete the definition of most of these drives. Each value is independent of the other, except for **Phillips** whose **Width** is determined by its **Depth**. The available drives are shown in Figure 4.7.3.8.
**Threads**: The size and shape of threads are determined by the cross section of the thread profile and the density of the threads along the length of the barrel. The depth of the threads are a function of the next two parameters.

- **Density**: This value is the number of threads per unit length.
- **Pitch**: This is the distance between points on two adjacent threads.

Note that **Density** and **Pitch** are reciprocals of each other. **Density** is typically used for English units of measure and **Pitch** is used for metric. Which of these two fields is shown at any given time is determined by the current **Working Units**.

- **Shape**: This is the cross section of the thread profile. There are three options, as follows:
  - **Machine**: This is a standard 60° ISO profile, typically used for both English and Metric static fasteners, such as nuts and bolts.
  - **Acme**: This is a stronger profile, often used to transfer motion with feedscrews or to position fixtures and vises.
  - **Wood**: This profile is covered with a 60° tooth in only 50% of its surface, which is appropriate for smoother materials, such as wood.

Examples of thread shapes are shown in Figure 4.7.3.9.

**Facetted Model Options**: The single option in this group applies only when **Facetted** is selected. It is dimmed otherwise.

- **Points Per Cycle**: The value entered in this field determines the resolution of the thread of a facetted screw/bolt.
4.7.4 Gears

The Gear tool creates simple involute tooth gears, the size and shape of which are governed by three parameters: the **diametral pitch**, the **number of teeth**, and the **pressure angle**.

The diametral pitch determines the number of teeth a gear has for a unit of diameter. For example, a diametral pitch of 20 teeth/inch means that a gear with 20 teeth will have a diameter of one inch, measured at the effective point of contact. A larger pitch value will result in a gear with more teeth for a given diameter. Consequently, the diameter of the pitch circle (d) is the product of the diametral pitch (pd) and the number of teeth (N): \( d = (pd)(N) \).

The pressure angle is the angle between the two faces of contacting gears along the center-line between the two gears. An example of a pair of mating teeth with a 20° pressure angle is shown in Figure 4.7.4.1(a). The example in Figure 4.7.4.1(b) has a gear with a 14.5° pressure angle, on the left, and a 26° pressure angle on the right. The shape of the involute curve of the tooth face is the same, but a gear set with a lower pressure angle will engage lower on the tooth face. Figure 4.7.4.1(c) shows another example of a pair of mating gears with a pressure angle of 14.5°.

![Figure 4.7.4.1](image)

*Figure 4.7.4.1: Gear teeth with pressure angle at (a) 20°, (b) 14.5° and 26°, and (c) 14.5°.*
**Gear**

This tool may be executed in two different modes: either through a single click, which generates a gear with preset parameters, or dynamically, which allows the parameters of a gear to be set through graphic input. Which method will apply and other settings are selected from the **Gear Options** dialog, shown in Figure 4.7.4.2, which can be invoked directly from the tool.

**Model Type**: Either **Facetted** or **Smooth** can be selected from this group of options, as with other object generating tools of **form·Z**.

**Type**: Either **Surface** or **Solid** can be selected from this group of options, each generating the respective type of object.

**Creation Method**: One of two input methods that are available for the generation of gears can be selected from this group of options, as follows:

- **Preset**: When this method is selected, a gear is generated with a single mouse click, using the following preset parameters. The position of the click represents the center of the base of the gear, which is generated relative to the active reference plane.

  - **Width**: This parameter is only available when **Solid** is selected and represents the distance between the two planar faces of the gear or the distance of the extrusion. When a solid gear is generated relative to the XY reference plane, the width can be thought of as height.

  - **Number of Teeth**: The number entered in this field is the integer tooth count. It is used in determining the outer diameter of the gear and the gear ratio of any final gear train.

  - **Outer Diameter**: This is a non-editable information field and displays the size of the outer diameter. This is calculated from the number of teeth and the diametral pitch.

- **Dynamic**: When this method is selected, the gear is generated through two or three mouse clicks, depending on whether it is a surface or solid object, respectively. The first click sets the center of the base. The second click establishes the diameter (size) of the gear. If it is a solid gear, the third click sets the width (extrusion distance) of the gear.

![Figure 4.7.4.2: The Gear Options dialog.](image-url)
The remaining options, found at the lower end of the Gear Options dialog, determine the shape of the teeth of the gear.

**Diametral Pitch:** This parameter defines the ratio between the number of teeth and a unit diameter distance. If all other parameters are unchanged, a larger pitch results in a smaller outer diameter.

**Pressure Angle:** This parameter determines the ‘steepness’ of the gear teeth. Smaller values will produce thinner gears that have less drag but are also weaker. 14.5° and 20° are common standard values. Note that gears that are intended to mesh should have identical values for both the Pressure Angle and Diametral Pitch.

**Base Fillet Radius:** This value is used to create a fillet at the bottom corners of the land between teeth. It is ignored if there is insufficient clearance between teeth for the requested value. Generally, large numbers of teeth or large pressure angles will create crowding and require smaller base fillet radius values.

**Center Bore Radius:** This option will create a central hole in the gear blank. It will be ignored if a value is given that is larger than the outer diameter of the gear.
Editing gear objects

Gears are parametric objects and the Query tool can be used to modify them. When clicking on the Edit... button found in the Query dialog, the Gear Edit dialog, shown in Figure 4.7.4.3, is invoked. It contains the same options found in the Gear Options dialog, except for the Dynamic method of generation.

The Update button and the Automatic option allow you to view the changes you make in the dialog, as in all other Edit dialogs.

![Gear Edit dialog](image)

*Figure 4.7.4.3:* The Gear Edit dialog.
4.8 Stairs

Stairs are structures that allow us to climb from one elevation to another, most commonly from one building floor to another. They are parametric derivative objects that are stored with their parameters and can be later edited, which is done with the same tool that generated them.

In form\textbullet{}Z there are two tools that can be used to generate stairs:

- Spiral Stair
- Stair from Paths

These tools are discussed in this section. A third tool that can also be used to generate stairs is Terrain Model; it is discussed in section 4.6.

A stair is a sequence of steps, each elevated by a constant increment, relative to the previous step. The elevation increments of the stairs are called risers. Risers tend to have the same width, which is also the width of the stair. The typically horizontal, flat surface of a step is called the tread. The tread also has a length, which is the horizontal distance traveled when climbing a step of the stair. All the treads taken together represent the length of a stair and all the risers taken together represent the height of a stair. The flight of a stair is typically the section of a stair that connects two building floors. With spiral stairs it may sometimes take more than one flight to connect two floors. Thus, from a different perspective, a flight is a cycle of a stair that returns it to the starting point when viewed from a top projection (floor plan). The height of a stair may be defined by the flight or for the complete stair.

In form\textbullet{}Z, spiral stairs are generated from (or about) an axis of revolution and straight stairs are generated from a vector line called the path line of a stair. While in real life stairs only function when they climb along the direction of gravity, which is typically the Z axis, there are no restrictions in form\textbullet{}Z relative to the direction in which a stair is generated. Spiral stairs follow the direction of their axis and stairs from path are generated relative to the plane of the path line. If the latter is a single segment, they are generated relative to the reference plane.
4.8.1 Spiral stairs

The spiral stairs are special objects of revolution produced by rotating entities at different step rates and angles.

Spiral Stair

Clicking with this tool on an axis generates a spiral stairs about that axis. The axis can be one of the Cartesian axes, an axis of a reference plane, or any segment of an existing object. Even though spiral stairs only make functional sense when generated along the Z axis (or whatever axis corresponds to the height of a structure), there is no such restriction, and they can be generated in any direction. This is illustrated in Figure 4.8.1.1. Where you click to select the axis is significant and represents the starting point of the spiral stair.

The parameters of the spiral stair are set in the Spiral Stair Options dialog (Figure 4.8.1.2), which is invoked directly from the Spiral Stair tool. It contains five tabs: Options, Tiles And Sides, Railings, Limits, and Status Of Objects. All except the last tab are discussed in this section. The last one is discussed in section 4.5.
The Options tab

**Step Width**: The parameters in this group determine the size of
the stair steps and their distance from the axis of revolution

**Inside Radius, Outside Radius**: The values entered in
these fields represent the distance of the inside and outside ends
of the stair from the axis of revolution (Figure 4.8.1.3).

**Width**: This is an information field that displays the total width
of the step (outside minus inside). It is updated every time the
**Inside** or **Outside** values change.

**Height Control**: The parameters in this group determine the height of the complete stair
and/or of each riser.

**Total Height**: The value in this field sets the height of the stair.

**Riser Height**: When this option is on, the value in its field determines the height of each
riser (step), and the total **# Of Steps** is given by **Total Height** over **Riser Height**.

**# Of Steps**: When this option is on, the number entered in its field determines the total
number of steps. The height of each riser is **Total Height / # Of Steps**.

**Length Control**: This group of parameters determines by how many degrees the spiral stair
should be revolved to cover the required height. Three methods are available for specifying the
degrees of the revolution.

**Total Angle**: When this method is selected, the value entered in its field determines the
total number of degrees through which the stair will be revolved. The angle of revolution per
step is given by dividing this number by the total number of steps.

**Step Angle**: When this method is selected, the value entered in its field determines the
cycle of revolution per step. The total angle of the complete stair is given by multiplying this
value by the number of steps.

**Step Width**: When this method is selected, the angle per step is determined by the required
width of each step, which is measured halfway between the inside and outside ends of the step.
The total cycle of the complete stair is given by the angle per step, times the number of steps.
**# Of Segments Per Step**: This parameter determines the roundness of the ends of the steps. A value of 1 results in four-sided steps, when the inside and outside ends of the steps consist of a single segment. A value of 3 (default) produces steps whose ends consist of three segments each. Higher values may be used to produce more rounded stairs. The effects of the # Of Segments Per Step parameter are illustrated in Figure 4.8.1.4.

**Direction**: The two items in this pop up menu determine the direction of the spiral stair, which can be Clockwise or Counter Clockwise.

**Triangulate**: When this option is on the non planar faces of the spiral stair are triangulated.

**Plain Object**: When this option is off, which is the default, the spiral stair is generated as a parametric object. When it is on, a plain facetted object is created.

As with the revolved objects, a preview dialog is available for the spiral stair, where parameters can be changed and previewed before the final generation of the object. The same dialog is used when editing a spiral stair.

**Edit**: When this option is on at the time an axis is selected for the derivation of a spiral stair, a preview dialog is invoked, where the spiral stair can be previewed. This dialog works as all the preview windows in form·Z.

**Adjust To New Parameters**: When this option is on, the settings currently selected in the Spiral Stairs dialog will be applied. This option can be on with or without Edit being on. Note however, that both cannot be off at the same time.
The Tiles And Sides tab

The options in this tab are shown in Figure 4.8.1.5.

With Tiles: When this option is selected, tiles that have shapes parallel to the steps they sit on, are generated. They are specified by a Height parameter and Front and Sides extensions. A stair with tiles is shown in Figure 4.8.1.6(a).

With Side Beams: When this option is selected, protective beams along the two sides of the stair will also be generated, as shown in Figure 4.8.1.6(b). Their exact dimensions are determined by a Width parameter (distance between the two sides of the beam) and Above and Under parameters, which specify by how much the beam extends above the top and under the bottom end of the stair.

The Limits tab

The options in this tab are shown in Figure 4.8.1.7.

Warn When Exceeding Limits: When this option is on, a warning is issued when the risers or treads of the stair exceed certain limits set in the numeric fields of this tab. The limits are Tread Min and Max, Riser Min and Max.
The Railings tab

The options in this tab are shown in Figure 4.8.1.8.

**Railings**: If this option is on, railings are also generated with the stair.

This tab contains two groups of three icons each, labeled **Rails** and **Columns**. They represent profiles that can be used to generate rails and columns. Of special interest are the first icons in these groups that generate wire columns and rails. These can subsequently be used as paths to generate solid versions of rails or columns by sweeping customized profiles. The wire columns can also be transformed to solid columns using the **Per Length Of Seg** option of the Place tool.

**X, Z**: These parameters in the **Rails** group apply when a rectangular or circular profile is selected. They represent the width and height of the profile that will be swept along the railing. Note that, when the circular shape is used, its radii in each direction are half of the values in these fields.

**X, Y**: As above, for the columns. **X** is the direction parallel to the length of the step. **Y** is parallel to the side of the stair.

**Position**: Recall that the sides of each step of a spiral stair may consist of one or more segments. The parameters in this group determine how many columns will be generated at each step. Note that this may vary between the **Out** and **In** edges of the spiral stair.

**Per Step**: When this option is on, one column is generated at the respective edge of each step and is placed in the middle of the edge.

**Per Segment**: When this option is on, one column is generated for each segment on the side of the stair. If only one segment is used at the side of the stair these options produce the same result.

**Height**: The value in this field determines the height at which the railings will be generated, as well as the height of the columns.

**Distance From Edge**: The value in this field specifies the distance from the side of the stair at which the wire or the axis of the columns will be positioned.
4.8.2 Stairs from paths

A variety of stairs can be derived from path lines that can be drawn anywhere in 3D space, but are typically drawn on a plane parallel to XY.

This tool can be used to generate stairs from path lines that can be open or closed shape. The stair is generated along the length of the path and multiple flights can be created from closed paths. A stair from path can be generated using either the postpick or prepick methods.

Using the postpick method, with the Stair from Path tool active, you click on the path line. The stair is generated immediately. Alternatively, you can use the Pick tool to prepick any number of path lines. Then, with the Stair from Path tool active you click anywhere in the window. A stair is generated from each of the path lines you had preselected.

The Stair from Path tool can be used to generate different types of stairs, which is determined by selections made in the Stair From Path Options dialog (Figure 4.8.2.1). This dialog is invoked directly from the tool and contains the same tabs the Spiral Stair Options dialog does.

The Options tab

**Type:** One of six items can be selected from this pop up menu. **Solid Stair, Beam Stair, Steps Only, Steps & Risers, Solid Ramp,** and **Beam Ramp.** These are types relative to the structure of the stair, with the last two being special cases of climbing structures. The six types of stairs and ramps are illustrated in Figure 4.8.2.2.

![Figure 4.8.2.1: The Stair From Path Options dialog.](image)

![Figure 4.8.2.2: Types of stairs from path: (a) Solid Stair, (b) Beam Stair, (c) Steps Only, (d) Steps & Risers, (e) Solid Ramp, and (f) Beam Ramp.](image)
**Layout**: One of three items can be selected from this pop up menu. These are types relative to how the steps (treads) of a stair are laid out along the segments of a stair. These types are significant for a simple segment stair. They are illustrated in Figure 4.8.2.3.

**Per Section**: This arrangement consists of complete numbers of stairs in each section of the stair.

**Continuous**: This layout is derived by subdividing the total length of the stair by the number of required treads. The steps are arranged in a continuous fashion, which may cause some steps to be on two different sections of the stair.

**With Landings**: This layout is constructed by first generating flat steps called *landings* at the turns of a stair and then a number of steps between the landings.

**Align**: The sides of a stair are parallel to the path line from which the stair is derived. The three items in this pop up menu determine how the two side lines are justified relative to the path line. The path line may be placed to the *Left*, *Center*, or *Right* of the stair. These options work as for the enclosures (see subsection 4.2.1). Examples of the different alignments are shown in Figure 4.8.2.4.

**Width**: This parameter determines the distance between the side lines of a stair, which also determines the width of its treads and risers (Figure 4.8.2.5).

**Height**: This value determines the vertical distance that the stair climbs. This is the distance of its top most step from the ground. This parameter may refer to the whole stair or a single flight, which is determined by the item selected from the pop up menu under it (*Total* or *Per Flight*).
# Of Flights: When the path line is a closed shape, this value determines how many flights will be generated from it. If the path is open, this value has no effect, since only single flights can be generated from open path lines.

To generate a single flight from a closed path line, that line is initially broken at its “first” point. Then the stair is generated as for the open path lines (Figure 4.8.2.6). To generate multiple flights, the path line is extended by separating itself one cycle per flight. The multi-flight stair is then generated as a continuous stair, as shown in Figure 4.8.2.7(a).

While multiple flights cannot be directly generated from an open path line, since there is no continuity to repeat the cycle, the single flight can be easily multi-copied vertically after it is generated. This is shown in Figure 4.8.2.7(b).

Also note that path lines with multiple cycles that either completely coincide or loop around irregularly can be drawn from the beginning. Such shapes need to be processed as single flights, which will happen automatically when they are open. Examples are shown in Figure 4.8.2.8.

Beam Height: This parameter that only appears when Beam Stair is selected in the Type pop up menu, determines the vertical height of the beam, as shown in Figure 4.8.2.9.
Stairs & Risers Thickness: This parameter that appears only when Steps Only or Steps & Risers appear in the Type pop up menu determines the thickness of the slabs from which these types of stairs are constructed.

Ramp Height: This value only applies to the Beam Ramp and specifies the height of its beam.

Step Calculation: The three mutually exclusive options in this group determine the method and value used to calculate the actual layout of the stair; that is how many, how high, and how deep is each step the stair requires.

Riser: When this option is selected, the given height of the riser determines how many steps will be constructed, which is given by the total height of the stair divided by the riser height. The depth of the step is given by the total length of the stair over the number of steps.

Tread: When this option is selected, the depth of the tread becomes the basis of the calculation. The total number of the steps is equal to the total length divided by the tread depth. The riser height is total height divided by number of steps.

# Of Steps: When this option is selected, the number entered in its field gives the number of steps the stair will have. Then the height of the riser and depth of the tread are calculated as above.

Landing Extensions: These options are only available when With Landings is selected in the Layout pop up menu. They determine whether the landings (larger steps generated at the turn of the stair) will end at exactly the inside point of the turn or will have an extension at front or back.

Front: When this option is selected, the landings have a front extension whose length (depth) is determined by the value entered in its field. If the value in the field is 0, then it is equivalent to deselecting this option (see Figure 4.8.2.10(b)).

Back: As for Front, but it affects the back end of the landings, which may have extensions at either, or both, or none of their ends (see Figure 4.8.2.10(c)).

Triangulate: When this option is on, the faces under beam stairs are triangulated.

Plain Object: When stairs include tiles, side beams, rails, and columns, by default they are all generated as a single object, which is parametric and can be edited. When this option is selected, a plain object is generated instead. When a plain object is generated, the Per Part option, which generates independent objects for the parts of the stair, is also available. These parts will be: the main body of the stair, the tiles, the side beams, one each for the rails, and the columns. Generating independent objects makes it possible to individually articulate parts of the stair, especially its rails and columns, especially when they are generated as wires.
**Edit, Adjust To New Parameters:** These options are as for the spiral stairs. When Edit is on, the Stair From Path Edit preview dialog is invoked, where the stair can be previewed before its final generation. The options in this dialog are as in the Stair From Path Options dialog, except that the Edit dialog has the following additional option:

**Reverse Direction:** When this button is clicked, the direction of the path of the stair is reversed, which caused the stair to run the opposite direction.

**The Tiles And Sides tab**

Similarly to the spiral stair, the stair from path may also have additional tiles that cover or extend beyond its steps, and additional side beams. The parameters of these features are selected from the Tiles And Sides tab, which is as for the spiral stair, except for one additional option, Flat At Landings. This is as for the railings (see below). Examples of tiles and sides are shown in Figures 4.8.2.11 and 4.8.2.12.

**The Railings tab**

The Railings tab for the stairs from path is shown in Figure 4.8.2.13. It is similar to that for the spiral stairs, except that it is missing the Position group of the spiral stair options and it has one more:

**Flat At Landings:** When this option is on, the section of the railings at the landings is parallel to the landing. If the option is off, which is the default, the railing at the landings has a slope.

**The Limits tab**

This tab is as for the spiral stairs (Figure 4.8.1.7).

**The Status Of Objects tab**

This tab is as for many other operations and is discussed in section 4.5.
4.9 Sweeps, skins, lofts, and caps

The following derivative tools are discussed in this section:

- Sweep
- Draft Sweep
- Skin
- Loft
- S-Loft
- Cap

Like the objects in the previous two sections, the objects generated by these tools are *parametric derivative objects*. They are derivative because they are constructed from other objects. They are parametric because they are internally stored with the parameters and shapes that generated them. They can be either *faceted* or *smooth*.

The sweeps and skins share similarities but they are also quite different. Both types of objects are generated from sources that are swept or placed along paths. However, different placement methods are used in each case. Each type has a few additional variations, such as axial sweep, two-path sweep, two-source sweep, etc. for the sweeps; along paths and cross skinning for the skin operation. While the sweep is parametric as a sweep, it can also be created as a nurbs entity. The latter is also true for the skins.

The lofts also share some similarities with the sweeps and the skins, but they are different in a significant way. They do not use explicit paths, but only source shapes, which are swept between each other, almost as if it were along implicit paths.

The S-Loft tool combines characteristics of the loft and the skin operations. They may be generated with source shapes only or they may include paths. One distinct characteristic is their ability to branch, which makes them unique among this group of operations.

The Cap tool generates a surface from a closed sequence of edges that can be open (free) or edges shared by two faces. The generated surface may also be optionally aligned to neighboring surfaces.
4.9.1 Sweeps

A 3D sweep is a solid or surface object that is derived by sweeping a source shape along a path. Both the source and the path shape can be open or closed, except for one type of sweep, the boundary sweep, which requires that the source shape be an open shape.

This tool is used to generate a 3D sweep by selecting a source shape and a path. The source shape may be a surface object, a 2D enclosure, a segment, an outline, or a face of an object. The same types of entities and shapes can also be used as paths, except for the 2D enclosures. The system will not object to picking a 2D enclosure as a path, but will only use its outer curve for the path of the sweep. The Sweep works with both planar and non-planar paths, except for the boundary sweep that requires planar paths.

The Sweep tool can be used to generate four different types of sweeps: axial sweeps, two source sweeps, two path sweeps, and boundary sweeps. The type of sweep and other parameters can be selected from the Options tab of the Sweep Options dialog (Figure 4.9.1.1), which can be invoked directly from the Sweep tool. All types of sweeps can be executed using either the prepick or the postpick method.

Model Type: Facetted, Smooth: These options are as for the revolved objects and only affect some of the sweep types, as summarized in the following table:

<table>
<thead>
<tr>
<th>Sweep Type</th>
<th>Facetted</th>
<th>Smooth Sweeps</th>
<th>Smooth Nurbz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axial</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Two source</td>
<td>√</td>
<td>-</td>
<td>√</td>
</tr>
<tr>
<td>Two path</td>
<td>√</td>
<td>-</td>
<td>√</td>
</tr>
<tr>
<td>Boundary</td>
<td>√</td>
<td>√</td>
<td>-</td>
</tr>
</tbody>
</table>

Cap Start, Cap End, Plain Object, Edit, Adjust To New Parameters: These options work as for the revolved objects.

The sweeps are parametric objects that can be edited after their initial generation. This is done using one of three tools: the Sweep, Edit Controls, and Query tool.

Figure 4.9.1.1: The Sweep Options dialog.
Axial sweep

With the postpick method, this sweep is executed by clicking first on the source and then on the path shape. With the prepick method, preselect any number of source shapes using the Pick tool. Then with the Sweep tool active, click on the path shape.

Figure 4.9.1.2: 3D axial sweeps along (a) a closed and (b) an open path.

The axial sweep preserves the source shape as originally drawn and, without making any adjustment, sweeps it along the path. Both the source and the path can be either open or closed shapes. When open source shapes are swept, the axial sweep produces surface objects. Examples of axial sweeps are shown in Figure 4.9.1.2.

Paths that consist of a single segment require special treatment. Because single segment paths contain only two points and three points are needed for the definition of a plane, the system needs to use a third point or to make a certain assumption about what plane the path lies on. For single segment paths the plane of the path is determined as follows:

- If the path lies on one of the orthogonal planes, that plane is used.
- If the path is parallel to an orthogonal plane, a plane parallel to that orthogonal plane is used.
- If the path lies on or is parallel to one of the orthogonal axes, which makes it parallel to two orthogonal planes, one of those two planes, or a plane parallel to one of those two planes is used. The XY plane takes precedence over YZ, YZ over ZX, and ZX over XY.
- If none of the above conditions occurs, the origin of the world space is used as the third point for the determination of the plane of the path.
In addition to being perpendicular to the plane of the path, the source shape is always swept in a position such that its plane is also perpendicular to the segments of the path. The direction of the path is significant and affects how the source shape is positioned. The source shape is initially placed on the first point of the path and is swept towards its end point. This is true for both open and closed paths. As illustrated in Figure 4.9.1.3(a), the same path drawn twice in opposite directions produces different results. The same is true when the path is a closed shape, as illustrated in Figure 4.9.1.3(b).

The path may or may not be planar. When non planar, as the source shape is swept, it is oriented according to the local orientations of the segments which converge to the point where a copy of the source is placed. Examples are shown in Figure 4.9.1.4.
Source Shape: The parameters in this group of options offer transformations that can be applied to the source shape as it is swept along the path.

Scale: When this option is selected, the source shape is scaled as it is swept along the selected path. The scaling factors are entered independently in the $X$ and $Y$ fields and, for open paths, they apply to the last placement of the source shape. Scaling factors for placements between the two ends of the sweep are calculated automatically by the system. When the lock option is selected, the same scaling factor is applied for both the $X$ and the $Y$ directions.

Figure 4.9.1.5: Scaling and rotating the source shape of axial sweeps.
(a) The paths and the source. (1-3) Open paths. (4-5) Closed paths. (3 and 6) Non planar paths. Axial sweeps (b) with no scaling and no rotation applied to the source; (c) the source is scaled; (d) the source is rotated; and (e) the source is both scaled and rotated.
When the path of the sweep is a closed shape, and the Scale option is selected, the scaling factors entered in the \textbf{X} and \textbf{Y} fields are applied to the source shape placed half way along the path. For example, if the scaling factors used decrease the size of the source shape, the copy of the source placed on the first point of a closed path will be the largest, and the source placed half way around the path will be the smallest. The calculation of half way is based on a count of the points in the path, and not on the size of the path. In addition, it is rounded to the next point when the path has an odd number of points. Application of scaling factors to sweeps along both open and closed paths are illustrated in Figure 4.9.1.5.

\textbf{Rotate:} When this option is selected, the source shape is rotated as it is swept along the path. The angle of rotation is entered in its field and determines either the \textbf{Total} or the \textbf{Per Step} angle of rotation, depending on which of these options is selected from the pop up menu. When \textbf{Per Step} is selected, the total angle of rotation is given by the angle times the number of segments in the path. As for the helix, changing \textbf{Total} to \textbf{Per Step} and vice versa behaves differently in the \textbf{Sweep Options} and in the \textbf{Sweep Edit} dialogs. In the latter, switching between these items also adjusts the value of the Rotate angle, which does not happen in the \textbf{Sweep Options} dialog.

For open paths, the total angle of rotation is applied to the last placement of the source shape, and the angles of rotation for the placements between the two ends are calculated automatically. For closed paths, any number of complete rotation cycles are applied, where a cycle is 360°. For example, if you enter 10° for \textbf{Total} rotation, the source will be rotated by 360°. If you enter 400°, it will be rotated by 720°, which is two rotation cycles. Sweeps with the Rotate Source option selected are shown in Figure 4.9.1.5.

\textbf{Perpendicular To Plane:} When this option is selected, the orientation of the source shape is not adjusted to the orientation of the segments that converge to the point where the source is placed, but is kept perpendicular to the active reference plane. Note that when the path contains segments which are perpendicular to the reference plane, this option does not generate satisfactory results in the area of the perpendicular segment.

This option has no effect when the path is planar and its plane is parallel to the reference plane. This option can be used in conjunction with the Scale Source and/or the Rotate Source options to either scale, rotate, or both scale and rotate the source shape as it is swept. An example of the effect of the Perpendicular To Plane option is shown in Figure 4.9.1.6.

\begin{figure}[h]
\centering
\includegraphics[width=0.7\textwidth]{sweeps}
\caption{Axial sweeps with \textbf{Perpendicular To Plane} (a) off and (b) on.}
\end{figure}
**Preserve Height:** This option is dimmed for the axial sweeps. It applies to two path sweeps.

**Alignment:** There are four options for aligning source shapes to the path of a sweep. The first three use a reference point on the source to align it to the path. That point may be the Origin, the Centroid (default), or the First Point of the source shape. These three options are illustrated in Figure 4.9.1.7.

![Figure 4.9.1.7: Alignment options of 3D sweeps: (a) at origin, (b) at centroid, and (c) at first point.](image)

While alignment relative to the centroid is possibly the most frequently used option for axial sweeps, alignment about the origin offers some otherwise hard to achieve possibilities. This is when we need to sweep more than one sources in a manner such that the relative position is maintained as they are swept. This is illustrated in Figure 4.9.1.8.
When the fourth **Alignment** option (**As Positioned**) is selected, the source shape is expected to be positioned relative to the start point of the path, exactly where the start of the sweep should be. This option affects the axial as well as the two source sweep (see below). Examples are shown in Figures 4.9.1.9 and 4.9.1.10, respectively.
Two source sweep

This type of sweep generates an object from two source shapes swept along a path in a manner such that their shapes are interpolated with each other as the sweep progresses along the path. When using the postpick method, this sweep is executed by selecting two source shapes and then a path. When using the prepick method, select any number of source shapes using the Pick tool. Then, with the Sweep tool active, click on the path shape. When preselecting source shapes, the total number of sources should be an even number, or the last source selected will be ignored. The **Two Source Sweep** is executed by taking pairs of sources in the order in which they are selected. The source shapes in a pair should be either both open or both closed shapes. The two source shapes in the pair may or may not have the same number of points. When they do not, the system automatically inserts new points to the source with the fewer points.

The **Two Source Sweep** is executed by sweeping a source whose shape changes from the source shape selected first, to the source shape selected second, as the source is swept along the path. For best results, the two source shapes should have the same direction, and their first points should align, or else twisted objects will be produced.

The **Scale** option does not apply to the **Two Source Sweep**, and is dimmed. All other options are as for the **Axial Sweep**. Examples of two source sweeps are shown in Figure 4.9.1.11.

*Figure 4.9.1.11:* (a) Open and closed paths, and two different source shapes.
(b) Two source sweeps shown in both hidden line and wire frame.
Two path sweep

This type of sweep generates an object by sweeping a single source along two paths. This is very similar to what other applications call bi-rail, especially when the two path sweep is generated as a nurbs object, which is an option available for the sweeps (see below).

When using the postpick method, this sweep is executed by selecting a source shape and then two path shapes. When using the prepick method, first select any number of source shapes using the Pick tool. Then, with the Sweep tool active, click on two path shapes to complete the selection. The two path shapes should either be both open or both closed shapes. They may or may not have the same number of points. When they do not, additional points are inserted automatically by the system to the path shape with the fewer points. The paths should have the same directions, or twisted objects will be produced.

The Two Path Sweep is executed by placing source shapes oriented along the lines that connect corresponding pairs of points of the paths. These copies of the source shape are also scaled to match the size of these lines. The scaling factor is determined using a bounding box that bounds the source shape. The width of the bounding box over the length of the line where it will be placed determines the scaling factor. For closed source shapes, the centroid of the shape is placed in the middle of the line that connects corresponding points of the two paths. For open source shapes, the midpoint of the line connecting the two endpoints of the source is placed in the middle of the line that connects corresponding points of the paths.

The scaling factor calculated as above, for each placement of the source, is always applied to the horizontal direction of the source shape. It is also applied to the vertical direction by default. However, if the Preserve Height option is selected, the scaling factor is not applied in the vertical direction, and the height of the sweep remains constant throughout the swept object. This is the only option from the Source group that applies to the two path sweeps. Examples of two path sweeps are shown in Figure 4.9.1.12.
Figure 4.9.1.12: Two path sweeps: (a) Three sources. (b) Pairs of paths: (1,2) open and (3,4) closed. (c) Selecting source A and outer path first. (d) Selecting source A and inner path first. (e) Selecting source A, inner path first, with Preserve Height option on. (f) Selecting source B. (g) Selecting closed source C. (h) As in (g) with Preserve Height option on.
**Boundary sweep**

The boundary sweep requires the source shape to be open, while the path shape may be open or closed. It generates a solid object always, by sweeping the source shape along the path line and then adding faces to close the object. When the path shape is open, it must have at least two segments and it should be possible to draw a line connecting its two ends, without crossing any segment of the path line. When this condition is not satisfied the system cannot generate a boundary sweep and issues an error message. This requirement is illustrated in Figure 4.9.1.13.

The boundary sweep is executed exactly as the axial sweep. However, the two types of sweeps generate significantly different objects, as illustrated in Figure 4.9.1.14. Recall that, when the source shape is open, the axial sweep generates surface objects. In contrast, the boundary sweep generates solids. The axial sweep does not have the restriction discussed above and, as shown in 4.9.1.14(b), it was able to generate an object for each of the three path lines. The boundary sweep could only generate an object for the top path. The other two paths fail the requirement that the end-to-end segment not intersect them (Figure 4.9.1.14(c)).

Even though open path lines are treated as closed when generating boundary sweeps, open and closed path lines produce different forms, as illustrated in Figure 4.9.1.15. When using a closed path line, the source shape is swept around the whole object. When using an open path line, the end of the object that is closed by the system is a flat surface. Additional examples are shown in Figure 4.9.1.16.

**Figure 4.9.1.13:** Connecting ends of path: (a) does not and (b) does intersect.

**Figure 4.9.1.14:** (a) Paths and a source. (b) Axial sweeps. (c) Boundary sweep can only be generated for top path line.

**Figure 4.9.1.15:** Boundary sweeps generated for (a) an open and (b) a closed path line.

**Figure 4.9.1.16:** Boundary sweeps: the same source shape is used for both open and closed paths.
Facetted versus smooth sweeps

As already mentioned, sweep objects can be generated either as facetted or as smooth models, which can be smooth sweeps or nurbs. The desired type of model is selected from the Sweep Options dialog. Whether Facetted or Smooth is selected as model type, the respective options are displayed in the dialog, as shown in Figure 4.9.1.17. All sweeps can be reduced to plain facetted objects either by selecting the Plain Object option in the Sweep Options dialog, or by using the Convert tool.

**Facetted Options:** The one option in this group applies when Facetted is selected for Model Type, which results in a facetted sweep, such as the one shown in Figure 4.9.1.18(a):

- **Triangulate Non Planar Faces:** When this option is on the non-planar faces of a facetted sweep are triangulated. Non-planar faces are typically generated when the path of the sweep is a non-planar shape.

![Figure 4.9.1.18](image)

*Figure 4.9.1.18:* (a) A facetted and (b) a nurbs swept object.

- **Join Adjacent Coplanar Faces:** When this option is on, any adjacent faces that are coplanar are joined. This option works as the same option that is available for the objects of revolution. Examples are shown in Figure 4.9.1.19.

![Figure 4.9.1.19](image)

*Figure 4.9.1.19:* (a) A source and four paths. Axial sweeps generated with Join Adjacent Coplanar Faces (b) off and (c) on.
**Smooth Options:** The options in this group apply when **Smooth** is selected for **Model Type**, which can be either a **Smooth Sweep** or a **Nurbz**, such as the one shown in Figure 4.9.1.18(b). When **Nurbz** is selected and the source objects are smooth curves (plain smooth, nurbz curves or spline curves), then the degree of the source curves is used to construct the new smooth sweep object. If the sources are plain faceted objects, the nurbz object is created according to the degree values entered in the **Length (U) Degree** and **Depth (U) Degree** fields.

**Reconstruct Path:** To achieve good results with a nurbz sweep, the path object must have sufficient control points to represent the 3D swept object. While the a few control points may be enough to construct the desired path curve, they are usually not enough to construct the sweep object. When this option is selected, the path curve(s) is reconstructed to generate sufficient control points along the path to achieve a more accurate sweep. This maintains the shape of the path while increasing the number of control points. This option is on by default. In cases where the control points in the original curve are significant, this option should be disabled. Figure 4.9.1.20 illustrates the effect of this option.

![Figure 4.9.1.20](image_url)

*(a) Source circle and nurbz path object with controls shown. Nurbz Sweep with Reconstruct Path (b) on and (c) off.*
Previewing and editing sweep objects

When the Sweep tool is executed while the Edit option is on in the Sweep dialog, the Sweep Edit dialog, shown in Figure 4.9.1.21, is invoked. This dialog offers the opportunity to preview the sweep object, as well as to experiment with different variations, including switches between the faceted and nurbz types. The preview dialog is also used when editing a swept object after its initial generation.

![Figure 4.9.1.21: The Sweep Edit dialog.](image)

The Sweep Edit dialog, on its left, contains the usual graphic window for previewing the object to be generated. Unique to this tool, the sweep preview dialog also contains a smaller window where the source shape is displayed and its placement can be viewed and graphically manipulated, using the following tools found under it.

- **Move**: Clicking in the window and dragging while this icon is active, moves the source shape.
- **Rotate**: With this icon active, clicking in the window and dragging in a circular motion rotates the source shape about the path.
Scale: With this icon active, clicking in the window and dragging scales the source up or down, relative to the path.

Flip Horizontal: Clicking on this icon flips the shape horizontally.

Flip Vertical: Clicking on this icon flips the shape vertically.

Reverse Direction: Clicking on this icon reverses the direction of the source shape.

Fit View: Clicking on this icon adjusts the view of the source shape to fit in the window.

The Scale, Rotate, Preserve Height, and Perpendicular To Plane options correspond to the same options in the Sweep Options dialog. They can be turned on or off from within the preview dialog for the sweep types to which they apply. Likewise, the Facetted and Smooth groups of options are the same with those outside the preview dialog and can be freely turned on or off. The resulting effects can thus be previewed before accepting the final shape of the swept object.

For Source Shape the options 1st and 2nd are available, and both are active when a two source sweep is previewed. The option 2nd is dimmed and inactive for all other sweeps. Selecting 1st or 2nd activates the display of the respective source shape in the preview window, and allows you to manipulate its position and shape, while the other source shape remains inactive in the background and is displayed in the ghost color. Pressing the option key on the Macintosh or ctrl+shift on Windows while executing a Rotate, Scale, Flip Horiz, Flip Vert, or Reverse operation will apply the operation to both source shapes simultaneously.

For Path Shape the options 1st and 2nd are available and work as for Source Shape. They are both active when a two path sweep is previewed. Only the option 1st is active for all the other sweeps. When the two paths selected for the two path sweep have opposite directions, the Reverse Direction of Path operation can be used from within the preview environment to correct the oversight. If the direction of the sweep needs to be reversed, then the directions of both paths need to be reversed.

After their initial generation swept objects can be edited using the Sweep or the Query tool. Both tools invoke the Sweep Edit dialog, where the parameters of a sweep object can be changed.
4.9.2 Draft sweep

The Draft Sweep tool constructs a new smooth object by offsetting a source object as it is swept along a path object. This tool works similarly to the Axial Sweep and always produces a smooth object, regardless of the types of the source and path objects.

Draft Sweep

Sweeping with draft allows the profile to be expanded and contracted as it is swept along the path. A practical use of draft sweep is for molded items. As the profile is swept, the ending profile is offset an equal distance, which facilitates removal of the item from a mold. The options to specify offsets by functions or curves are available and they make it possible to model a variety of mathematical and free form surfaces.

Either the prepick or postpick method can be used. With the postpick method, the Draft Sweep tool is selected first and then two edges, two outlines, two faces or two surface objects are selected. If the By Curve option is selected, an additional edge, outline, face or object is picked. With the prepick method, any number of edges, outlines, faces, or objects are picked first, followed by the selection of the Draft Sweep tool and a mouse click anywhere in the graphics window. The Draft Sweep tool handles the selected entities in pairs (or triplets, if the By Curve option is selected) in the order they were picked.

The Draft Sweep tool works only on open paths. Draft sweeps work best with G1 continuous sources and paths. The results of non-G1 continuous paths or sources may not be as expected and a warning message is posted when they are encountered.

The Draft Sweep Options dialog (shown in Figure 4.9.2.1) consists of three tabs: Options, Display Resolution, and Status Of Objects. The first tab contains the options that affect the generation of Draft Sweep objects. The second tab contains the common parameters that dictate how the faceted version is displayed on the screen. The third tab contains the options that determine how the operands picked for the operation are treated.
**Source Shape:** This group of options contains **Draft Sweep Options** and transformations that can be applied to the source shape as it is swept along the path.

**Draft Sweep Options:** This group of options determines the method and amount by which the source shape is offset as it is swept along the path.

**By Angle:** The draft angle is a real number that represents the angle with which the swept profile is to draft while sweeping. For closed profiles, a positive draft angle causes a draft **out**, while a negative draft angle causes a draft **in**, as shown in Figure 4.9.2.2.

**By Distance:** The draft distance specifies an offset distance for the border of the swept surface relative to the path of the sweep. It is similar to draft angle, except that an offset distance is used instead of an angle. The larger the defined offset, the wider the draft distance. Zero distance creates a swept surface parallel with the path. A negative offset creates a swept surface, which is closer to the path at the end than at the beginning.

**By Function:** The draft function specifies the offset of the profile as it moves along the path. The user may either choose from a pre-defined set of functions or type in a new function.

**Pre Defined:** This menu contains four options: **Sine**, **Cosine**, **Half-Sine**, and **Half-Cosine**. Examples of all these are shown in Figures 4.9.2.3 and 4.9.2.4.

**Frequency:** This specifies the number of repetitions wanted along the length of the path curve.

**Amplitude:** This specifies the maximum distance the profile is offset at any point along the path.
Figure 4.9.2.3: Draft sweep by Pre Defined functions using a 10cm circular source and a straight line path. (a) Cosine with Amplitude = 5cm. (b) Sine with Amplitude = 10cm. (c) Cosine with Amplitude = 2.5cm. (d) Sine with Amplitude = 5cm. In all cases Frequency = 2.

Figure 4.9.2.4: Examples of Draft Sweeps of a circle along helix using the Pre Defined functions: (a) Sine (b) Cosine (c) Half–Sine and (d) Half–Cosine, shown in wireframe and shaded Render.
**Custom:** When this option is selected, an arithmetic expression is typed by the user in its alpha field. This expression represents the function to be executed in generating a draft sweep and it should be syntactically correct or otherwise it can not be executed. The following table lists the 5 arithmetic symbols and a number of functions that are available. Using these, expressions are written in the standard algebraic syntax. See examples in Figure 4.9.2.5.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Add</td>
</tr>
<tr>
<td>-</td>
<td>Subtract</td>
</tr>
<tr>
<td>*</td>
<td>Multiply</td>
</tr>
<tr>
<td>/</td>
<td>Divide</td>
</tr>
<tr>
<td>^</td>
<td>Exponent</td>
</tr>
<tr>
<td>log</td>
<td>Common logarithm (to the base 10)</td>
</tr>
<tr>
<td>ln</td>
<td>Natural logarithm (to the base e)</td>
</tr>
<tr>
<td>abs</td>
<td>Absolute function</td>
</tr>
<tr>
<td>sin</td>
<td>Sine function</td>
</tr>
<tr>
<td>cos</td>
<td>Cosine function</td>
</tr>
<tr>
<td>tan</td>
<td>Tangent function</td>
</tr>
<tr>
<td>cot</td>
<td>Cotangent function</td>
</tr>
<tr>
<td>sec</td>
<td>Secant function</td>
</tr>
<tr>
<td>csc</td>
<td>Cosecant function</td>
</tr>
<tr>
<td>sinh</td>
<td>Hyperbolic Sine function</td>
</tr>
<tr>
<td>cosh</td>
<td>Hyperbolic Cosine function</td>
</tr>
<tr>
<td>tanh</td>
<td>Hyperbolic Tangent function</td>
</tr>
<tr>
<td>coth</td>
<td>Hyperbolic Cotangent function</td>
</tr>
<tr>
<td>sech</td>
<td>Hyperbolic Secant function</td>
</tr>
<tr>
<td>csch</td>
<td>Hyperbolic Cosecant function</td>
</tr>
<tr>
<td>arcsin</td>
<td>Arcsine function</td>
</tr>
<tr>
<td>arccos</td>
<td>Arccosine function</td>
</tr>
<tr>
<td>arctan</td>
<td>Arctangent function</td>
</tr>
<tr>
<td>arccot</td>
<td>Arccotangent function</td>
</tr>
<tr>
<td>arccsc</td>
<td>Arcccosecant function</td>
</tr>
<tr>
<td>arccsc</td>
<td>Arcccosecant function</td>
</tr>
<tr>
<td>arcsinh</td>
<td>Hyperbolic arcsine function</td>
</tr>
<tr>
<td>arccosh</td>
<td>Hyperbolic arccosine function</td>
</tr>
<tr>
<td>arctanh</td>
<td>Hyperbolic arctangent function</td>
</tr>
<tr>
<td>arccoth</td>
<td>Hyperbolic arccotangent function</td>
</tr>
<tr>
<td>arccsch</td>
<td>Hyperbolic arccosecant function</td>
</tr>
</tbody>
</table>

Certain functions like “tan(x)” are not continuous and hence can not be used directly for the custom function. However, they can be used in combination with other functions to produce continuous results. For instance “tan (cos (x))” produces a continuous result and is shown in 4.9.2.5 (c).

The popup menu shows the recent list of strings used for the custom function (a maximum of 10).
Figure 4.9.2.5: Example of Draft Sweep of a circle along an arc using custom functions: (a) \( \frac{x^2}{1000} \), (b) \( 10\cos(4\sin(x/20)) \), (c) \( 10\tan(\cos(x/10)) \), and (d) \( -x/10 \). Shown in wireframe and shaded render.

By Curve: When this option is on, an additional curve needs to be picked, after the selection of a source and a path. This curve, called draft curve, specifies the offset of the profile as it moves along the path. Instead of typing in a function, this option lets the user specify the draft offset in a geometric way. The wire objects picked for this option are required to be drawn on the XY plane. An example is shown in Figure 4.9.2.6.

Rotate, Alignment, Cap Start, Cap End, Plain Object, Edit, Adjust To New Parameters: These options work as for the sweep objects.

Figure 4.9.2.6: Sweeping a square along the line using the By Curve.
Previewing and editing draft sweep objects

When the Draft Sweep tool is executed while the **Edit** option is on in the **Draft Sweep Options** dialog, the **Draft Sweep Edit** dialog, shown in Figure 4.9.2.7, is invoked. This dialog offers the opportunity to preview the sweep object, as well as experiment with different variations.

The **Draft Sweep Edit** dialog contains three preview windows. The largest on the left displays the object to be generated. The preview window in the upper-middle displays the source shape and the preview window on the lower-right shows the draft function as a graph in the xy plane, when **By Function** is on. It remains blank when another option is selected.

**Update**: Clicking on this button applies the new parameters and refreshes the preview window. This button is dimmed when the parameters of the preview are up to date.

**Automatic**: When this option is on, the preview is automatically updated every time a parameter is changed. When this option is on, the **Update** button is dimmed.

*Figure 4.9.2.7: The Draft Sweep Edit dialog.*
4.9.3 Skinning

Skinning may be thought of as the process that transforms a wire representation of an object to a facetted representation. Recall that a wire representation is simply a collection of lines, called wires, which trace the edges of a shape. A facetted representation is a collection of bounded surfaces, known as faces or facets, that completely enclose a volume or cover a surface. Because the facets are surfaces, the facetted representation is able to represent solids and meshed surfaces, which the wire representation can not do. However, the wire representation can describe a form as well as the facetted representation does. Frequently, it is actually easier to sketch a form with wires, which is where the skinning operation comes in handy; that is, it is used to transform the wire to a facetted representation and, consequently, to a more meaningful solid or surface object. This is illustrated in Figure 4.9.3.1.

In (1a), the shape of a vase has been outlined with wires. That these are wires becomes visible when they are separated, as shown in (1b). The form in (2a) is identical with that in (1a), except that it is shown in hidden line, which verifies that it consists of bounded surfaces. Again, that these are facets (and not wires) becomes visible when they are exploded, as in (2b). The form in (2a) was actually derived from the wires in (1a) through skinning.

The example in Figure 4.9.3.1 shows a fully defined skinning operation. Thinking of the closed shapes in (1b) as the source shapes, and of the open shapes as paths, to each vertex (point) of each path corresponds a point of a source, and vice versa. Corresponding points are actually pairs of coincident points. In other words, there is no “free” point for which a shape will have to be generated through interpolation. Neither are corresponding points close enough, as opposed to being exactly coincident.

Skinning can also be derived from sources and paths that may not fully satisfy the above conditions; however, the more the conditions are satisfied, the more accurate results can be expected. This principle should be kept in mind when using the operation. While skinning is a very powerful operation, it is also more sensitive than other operations to conditions such as the above. The user needs to keep in mind that the full set of source and path shapes (as in Figure 4.9.3.1) is the normal starting point of the operation. While it will also work with less, it will always need sufficient information to be able to properly fill the missing parts. These principals are discussed in more detail in the remainder of this subsection.
Skin

The Skin tool, located next to the Sweep tool, can be used to derive an object from sets of properly positioned profile shapes. Source shapes are swept along path shapes, which makes the skin operation in some ways similar to the sweep. However, with the skin operation the source shapes are also geometrically interpolated both relative to themselves and relative to the paths, when necessary.

There are two types of skin operations: skinning along paths and cross skinning. They are illustrated in Figure 4.9.3.2. The former is of a linear, while the latter is of a rotational character. The latter, actually resembles the objects of revolution and can be thought of as a revolving shape that is interpolated along a closed path, which is not exclusively a circular shape. On the drawings, throughout this subsection, we use $p$ to label the paths and $s$ to label the source shapes.

*Figure 4.9.3.2:* The two types of the Skin operation: (a) skinning along path and (b) cross skinning.

*Modeling* • Sweeps, skins, lofts, and caps
The skin operation is primarily intended to let you draw the outlines (profiles) of a 3D form, from which the program can create an object, by generating a skin from these profiles. This implies that both the source and the path shapes are drawn exactly in the positions where they will be used. This is called placement by current position and is the recommended way for executing the skin operations.

Because drawing the source shapes in their exact positions may sometimes be tedious, the option to draw them elsewhere and have the program place them on the paths is also available. It is called placement by anchor points. When using this method, you still need to tell the program where to place the source shapes. As the name implies, you do this by marking points on both the source shapes and the paths, which will be used as anchor points. An anchor point of a source is a point which will lie exactly on a point of a path (which is also an anchor point), when the source is placed. You mark the anchor points on both the source and the path shapes using the Set/Clear Point Marker tool (see subsection 4.21.8). An option to display the marked points is also available and can be selected from the Wire Frame Options dialog. The program needs to be able to detect no more and no less than the necessary number of anchor points, on both the sources and the paths, in order to be able to properly place the source shapes. These requirements are discussed in more detail in the respective section. The two methods of placement are illustrated for the skinning along paths operation in Figure 4.9.3.3.

In 4.9.3.3(a) three paths and two sources are used, which are already placed in position (placement by current position). The source and the paths touch accurately at corresponding anchor points. The shapes in the example actually outline the surface to be generated.

The example in 4.9.3.3(b) generates the same object as in the previous example using a single source and the same three paths. The source is not in position; it has been drawn away from the paths. This operation is executed using the placement by anchor points method. With the Set/Clear Point Marker tool we mark the points highlighted with the bullets. Note that we mark three points on the source, one for each path. We mark one point on each path, corresponding to the single source we are using. In general, as many points as the number of paths should be marked on each of the sources. Once the marking of the profile shapes has been completed, the skin operation is executed producing the surface shown.

Figure 4.9.3.3: Skinning open shapes with (a) placement by current position and (b) placement by anchor points.
Normally, when using the placement by current position method, anchor points of sources should be exactly on anchor points of paths. However, the program will also try to pair corresponding anchor points even when they are not exactly on top of each other. It does so using the value in the **Tolerance** field. The higher this value is the more permissive the program becomes, but also the risk of confusing which points should be paired becomes greater. While tolerance based point matching is available for convenience, it may sometimes produce unanticipated results (even though they are mathematically correct). For complete control of the results, placing corresponding anchor points on top of each other is recommended.

Both skin operations can be executed by having the same numbers of points on corresponding portions of the source and the path shapes, or by having different numbers of points and letting the program generate additional points, where necessary. Note that the condition of same numbers of points is relative to corresponding portions of the shapes that are between anchor points. It is possible that two source shapes with the same total number of points, not to have the same number in sections between anchor points. This is illustrated in Figure 4.9.3.4.

The two circular shapes that are used as sources have the same number of points and so do the four paths. In (a), the paths are positioned in such a way that the anchor points subdivide the source shapes into four portions, each with four segments. Thus corresponding portions of the sources, that is, portions between the same pair of paths, have the same number of points and the program does not need to generate new points. In (b), because of the way the paths are positioned relative to the sources, this is not true. Corresponding portions of the sources have two and six segments. Thus the program has to create additional points on the portions that have two segments. This is done as shown in (c).

Following is a summary of the recommended ways for using both skinning operations:

- Place the sources exactly where you want them; that is, whenever possible use the placement by current position method.
- Place the corresponding pairs of anchor points on top of each other; that is, avoid using a tolerance to match corresponding anchor points.
- Have the same number of points in corresponding portions of both the source and path shapes; that is, avoid making the program generate new points.

Meeting these conditions gives you more complete control of the resulting form. The skinning operations are quite permissive in many different ways, including adjusting the directions whenever possible, and will execute fine even when the above conditions are not met. However the results may be different from what you have anticipated.

---

**Figure 4.9.3.4**: Skinning with sources whose corresponding portions have
(a) the same and (b) different number of points. (c) New points are inserted so that corresponding portions have the same number of points.
Executing the Skin operation

To execute a skin operation follow these steps:

• Invoke the **Skin Options** dialog (Figure 4.9.3.5) and select the desired options to tell the program:
  
  • the type (**Facetted** or **Smooth**) you will generate and which operation you will execute (**Skinning Along Paths** or **Cross Skinning**);
  
  • how many sources and paths you have;
  
  • whether the sources are already placed (**By Current Position**) or whether anchor points will be used to place them (**By Anchor Points**); in the latter case make sure you mark the anchor points on your shapes before executing the operation.

• Next you can use either the prepick or the postpick method to execute the skinning operation.

  • With the prepick method, use the Pick tool to select the sources and paths, in this order. Then, with the Skin tool, click anywhere in the window.
    
    • If less objects than the number in the dialog are selected, an error message is posted.
    
    • If more objects are preselected, the additional objects will be ignored.

Note that the program has no way of knowing which objects are the sources and which are the paths, other than by the order you select them. If the dialog indicates *n* sources and *m* paths, the first *n* objects selected will be used as sources and the next *m* objects as paths. If these numbers do not correctly correspond to your intentions, either a nonsense object will be produced or the program may be able to detect the mistake and post an error message.

  • With the postpick method, use the Skin tool to select the sources and the paths, in this order. The program will let you pick as many objects as indicated in the dialog and will proceed with the execution of the operation as soon as the last object is selected. The above note about correctly selecting the objects applies to the postpick method also.

While the sources need to be picked first and the paths second, the order in which the sources and the paths by themselves are picked is insignificant for the placement by current position, but significant for the placement by anchor points.

Note that, whether you select a **Facetted** or a **Smooth** model type, the skin operation is executed the same way. The two types are discussed some more at the end of the section.
Types of the Skin operation

One of the two available variations of the skin operation and its parameters are selected from the Skin Options dialog (Figure 4.9.3.5).

Skinning Along Paths

This type of skinning is generated from one or more source shapes and three or more paths. The numbers of shapes to be used are entered in the # Sources and # Paths fields in the Skinning Along Paths box. Note that a minimum of three paths are required and the dialog will not allow you to enter a number less than 3 in the # Paths field.

Either or both the source and the path shapes can be open or closed. Closed sources with closed paths generate solid objects (Figure 4.9.3.6). Open sources with open or closed paths generate surface objects (Figure 4.9.3.7). Closed sources with open paths may generate surface or solid objects. They generate solids when both the Cap Start and the Cap End options at the bottom of the Options tab are on (Figure 4.9.3.8).

When open sources are used, their end points should be on (or close enough to) points of a path. When placement by anchor points is used, the endpoints of open source shapes should be marked as anchor points. Figure 4.9.3.8 illustrates the placement by anchor points method for skinning along open paths, with both open and closed source shapes.

Figure 4.9.3.6: Skinning with closed sources and closed paths produces a solid object.

Figure 4.9.3.7: Skinning with open sources and open paths produces a surface object.

Figure 4.9.3.8: (a) Skinning By Anchor Points: open paths (b) with open sources produce a surface; (c) with closed sources and the Cap Start and End options on produce a solid object.
It is not necessary to place source shapes at the ends of open paths, even though doing so should be considered good practice. The skinning along paths operation actually needs source shapes at the ends of open paths. When none is placed, the program automatically makes a copy of the source shape closest to the respective end and places it at that end. The implication of this is that the section of the skinned object between the end of the path and the position of the closest source shape takes the form of that source shape. This is illustrated in Figure 4.9.3.9, where a single source shape is placed between the ends of four open paths. Copies of this source shape are first placed at the ends of the paths and the skinning operation is then executed. The resulting object takes its shape from the single source shape.

Regarding the directions of the source and path shapes used in a skinning operation, it is good practice to keep them consistent and to line up the first points of closed shapes. However, these conditions are not always significant and in many cases the program has the ability to correct inconsistencies in directions.

When placement by current position is used, in most cases, form\textsuperscript{Z} is able to detect inconsistencies in directions of both sources and paths and to correct them. It is unable to make corrections when the inconsistency is on closed paths, and less than three sources are used, or on open paths and only one source is used. For closed shapes, the program can also adjust the positions of the first points, in order to line them up.

With placement by current position, the order in which the set of sources and the set of paths are selected is not significant (but the sources should be selected before the paths). Since their position in space is already specified, the program can detect their order, even if they are not picked in order. Which source is picked first and which path is picked first is significant when corrections in directions need to be made. All the sources and all the paths take the directions of the first source and path, respectively. When the skin operation generates a surface object, the directions of the sources and paths affect the direction of the resulting surface and which side it faces (which side is positive). The directions of the sources and paths have no effect when the resulting object is a solid, which always faces outside.

When placing by anchor points, the order in which the sources and the paths are selected is significant. They should be selected in the order the shapes are intended to be swept. The source shape selected first will be placed on the first marked points of the paths. The anchor points of the source will be matched in the order they appear to anchor points of paths, in the order the paths are selected. If any of these entities are not in the proper order, a nonsense object may result.
Two different interpolation methods are available and can be selected from the **Skinning Along Path** box in the **Skin Options** dialog.

**Centroid Based:** When this method is selected, a rotational interpolation method is used, when placing portions of a source between two corresponding paths.

**Boundary Based:** When this method is selected, the interpolation is based on the boundary slopes of the source shapes being interpolated.

Even though the first method handles smoothly curved shapes better, the second handles shapes with sharp angles better. In most cases the two will produce almost identical results. However, there are also cases where the result produced by each will show some visible differences. In general, the denser the paths are (more paths are used) and the more evenly their points are distributed, the more likely it is that the two interpolation methods will produce similar results. When fewer skinning profiles are provided (fewer paths and fewer sources), the program has to do more “guessing”, in which case the differences between the two interpolation methods become more noticeable. Some of these differences are illustrated in Figure 4.9.3.10.

---

**Figure 4.9.3.10:** (a) Skinning along paths using
(b) **Centroid Based** and (c) **Boundary Based** interpolations.
Cross Skinning

This type of skinning is generated from one or more sources and a single path, which is always a closed shape. The number of sources to be used is entered in # Sources field in the Cross Skinning box of the Skin Options dialog. For cross skinning, centroid based interpolation is always applied; there are no other options specific to this type of skinning to be selected.

While cross skinning uses a single path, any number of source shapes can be used, which can be open or closed. Possibly the most typical case is when two closed source shapes are used, as shown in Figure 4.9.3.11. The following conditions can be observed:

• Each of the source shapes touches the path at two anchor points (highlighted with the small bullets).
• The two sources touch and cross each other at two additional points, to be referred to as the apex points. There is one above and one below the path shape (highlighted with the larger bullets).

The position of the apex points relative to the path is significant for deriving "clean" cross skinned objects. The line that connects the two apex points (shown as a dash line in Figure 4.9.3.11) can be thought of as an axis about which the source shapes are revolved. In the example, this axis line is exactly perpendicular to the plane of the path and passes through its centroid (marked with the white bullet). While these conditions represent the cleanliest way of setting up the profile shapes of a cross skinning operation, they are not necessary conditions. Cross skinning will also work when the apex points do not project exactly on the centroid of the path, and will work when the axis line is not exactly perpendicular to the plane of the path. However, extreme deviations from these conditions may produce unanticipated results. Deviations from these conditions are illustrated in Figures 4.9.3.12 and 4.9.3.13.

Note that, above, whenever we talk about the plane of the path, we assume that the path is planar. However, this again is not a necessary condition. Whenever the path is not planar and its plane is needed, the average plane of its points is used.
When cross skinning is executed, the closed source shapes are actually broken at their apex points and are swept along the paths as open shapes. The example in Figure 4.9.3.11, which uses two closed sources, is actually executed using four open source shapes. It is, consequently also possible (and frequently desirable) to use open source shapes directly, when executing cross skinning. This is illustrated in Figure 4.9.3.14, where three open source shapes are swept along a closed path. Note that this type of cross skinning lends itself better to the generation of non-symmetrical forms. The required conditions are analogous to those for cross skinning with closed sources:

- Each of the open source shapes should touch the path at one anchor point.
- Corresponding end points of all the open sources should coincide (or be close enough). These constitute the apex point of the cross skinning operation.

It is also possible to execute cross skinning operations with a single source, which may be a closed or an open shape. In these cases the positions of the apex points are determined as follows:

- **Closed source**: The apex points are those whose projections to the path plane are the closest to the centroid of the path. The cleanest case is when there are two points on the source (one above and one below the path) that project exactly on the centroid of the path. Such a case is illustrated in Figure 4.9.3.15.

- **Open source**: The apex points are at the end points of the open shape. Again, the closer the projections of these points are to the centroid of the path the cleaner the skinned object will be. Such an example is illustrated in Figure 4.9.3.16.
Cross skinning can also be executed with **half** sources of both closed and open source shapes. A half of a closed source is derived by breaking it at the anchor points where it touches the path. Or it can be drawn directly as a half shape. The half of a closed source is an open shape, which has its end points on the path. Similarly for open sources. Their half is derived by breaking them at the point where they touch the path.

While complete closed and open sources always produce solids, their half shapes produce surface objects. The use of halves of closed and open source shapes is illustrated in Figure 4.9.3.17 and 4.9.3.18, respectively. Use of half sources should satisfy the following conditions:

**Halves of closed sources:**
- The end points of the sources should be on two points of the path (anchor points).
- The source shapes should all cross at one point (the apex).

**Halves of open sources:**
- One of the end points of the sources should be on a point of the path.
- The other end points of the sources should coincide (the apex).

When a single half of a closed or a single half of an open source shape is used, the determination of the apex point is analogous to when complete sources are used. However, the axis is always perpendicular to the plane of the path. Cross skinning with single half sources are illustrated in Figures 4.9.3.19 and 4.9.3.20.
Cross skinning can also be executed using the placement method, but only when closed sources or half closed sources are used. Open source shapes simply do not carry enough geometric information to be able to place them.

When using the placement by anchor points method, a sufficient number of anchor points should be marked on the source shapes and on the path with the Set/Clear Points tool. How many anchor points are required is summarized below:

**Complete closed sources:**
- Each source should have four anchor points; two to be placed on the path and two for the apex points.
- The path should have two anchor points for each source. That is, if four sources are used, the path should have $2 \times 4 = 8$ anchor points.

**Half closed sources:**
- Each source should have three anchor points; two of these should be at the ends of the sources and will be placed on the path; the third anchor point is for the apex.
- The path should have two anchor points for each source, as for the complete closed sources.

While the order in which the sources are selected is insignificant when placement by current position is used, this is not true for placement by anchor points. Both the order of selection and direction of the shapes is significant. They determine the sequence in which the source shapes are placed on the path and their orientation. These matters are discussed in more detail and are illustrated in the respective section, below.
**Placement types**

As already discussed, for both skinning along path and cross skinning, there are two ways in which the operation can be executed, with respect to how the source shapes are positioned.

**Anchor Points:** For this type of placement, it is insignificant where the source shapes are currently positioned. They will be properly placed relative to the paths by the program, before execution of the skin operation proceeds. The program places the sources using anchor points, which must have previously been marked on both the source and path shapes, using the Set/Clear Point Marker tool.

**Place Only:** This option represents a variation of the placement by anchor points. When on, it only places the source shapes relative to the paths and exits. It does not execute the skin operation. This option, which is off by default, is useful for previewing purposes. You can use it to verify that the source shapes will be placed where intended, or you may want to make adjustments before actually executing the skinning operation.

**Current Position:** This type of placement assumes that the source shapes are already positioned where they are intended to be. It proceeds with the execution of the skinning operation immediately, provided the program can properly pair points on the sources with points on the paths. While for some forms it may be tedious to accurately place sources relative to the paths, this is the recommended (and default) placement type. It gives better control to the user (Figure 4.9.3.23). For best results, key points of the sources should be exactly on corresponding points of the paths. If this condition is not met, the operation will fail. However, the option to also use tolerances for matching corresponding points, which are not exactly coincident, is also available.

**Tolerance:** When anchor points are not exactly on top of each other, the program will match points whose distance is less than or equal to the value entered in this field. This feature can be used to compensate for small imperfections in positioning the source shapes. However, it may sometimes produce results that are mathematically correct, but different from those intended. We recommend that you only use conservative tolerance values, such as about 1% of the largest dimension of your shape.

*Figure 4.9.3.23:* Sources carefully placed on the paths produce the best results.
Point pairing

The execution of both skinning operations requires an equal number of points on corresponding portions of both the source and path shapes. **Corresponding portions** are those that are between anchor points positioned on the same pair of paths or sources. This is illustrated in Figure 4.9.3.24. Note that the total number of points on the sources or the paths, and whether they are equal, is insignificant, whenever the sources and the paths are not evenly distributed relative to each other.

When the equal number of points condition is not met, the program will fill in the missing points. That is, it will insert new points where they are missing. It will not make any changes to the positions of existing points. New points are inserted between existing points and their positions are determined proportionally.

While we have recommended that it is good practice to always use matching numbers of points on corresponding portions of the source and path shapes, there are also cases where it makes perfect sense to use unequal numbers. The most notable such case is when it is desired to use straight lines, either in portions of sources or in paths. When these lines are mixed with curved lines, the program will generate on the straight lines new points by distributing them proportionally to the points on the curved lines. In such cases the program can generally do a better job positioning these points than if they were inserted manually. Two such examples are shown in Figure 4.9.3.25.
Sequence of selection

For all types of skinning you select the sources first and the paths second.

There are cases where the sources and the paths are interchangeable. That is, profiles selected as sources in one execution of the skin operation may be selected as paths in another execution, and vice versa. This may even happen by mistake, when the same number of sources and paths are used for a skin along paths operation. In these cases the result produced is typically the same, as illustrated in Figure 4.9.3.26. In (a), the four closed shapes are selected as sources (first) and the four open lines as paths (second). In (b), the open lines are selected as sources (first) and the closed shapes as paths (second). The results are the same.

Different results are derived when sources and paths are switched in cross skinning, as illustrated in Figure 4.9.3.27. Shapes A, B, and C can be selected in six different combinations. Three of these are shown. In (a), A is selected first, B second, and C third. This causes A and B to be used as sources, and C as path. In (b), the shapes are picked in order B, C, and A. In (c), they are selected in order C, A, and B. Each case produces a different form.

Figure 4.9.3.26: When skinning along paths with placed sources, reversing the selection of sources and paths produces the same result.

Figure 4.9.3.27: When cross skinning with placed sources, different forms are generated when the profiles are picked in a different order.
When placement by current position is used, the order in which the sources are picked and the order in which the paths are picked is insignificant. However, this is not true when placement by anchor points is used. The order of picking is very significant, for both skinning along paths and cross skinning. If the paths are picked out of order, a nonsense object will be produced. If the sources are picked in a different order, a different object will result. The sources are placed on the paths in the order they are selected and a different selection sequence produces different forms, as illustrated in Figure 4.9.3.28. In (a), the sources are selected in A, B, C order; in (b) in B, C, A order; and in (c) in C, A, B order. Each different selection sequence produces a different form.

Likewise for the cross skinning operation. Selecting the sources in a different order produces different forms, as illustrated in Figure 4.9.3.29. The two sources A and B, shown in (a), are selected in A, B order for the object in (b). They are selected in B, A order for the object in (c). How the source profiles are placed is shown to the left and the actual objects generated are shown to the right. Note that both have also been triangulated.

**Figure 4.9.3.28:** Skinning along paths and placing by anchor points: Different forms result when the sources are picked in a different order.

**Figure 4.9.3.29:** Cross skinning and placing by anchor points: Different forms are generated when the sources are picked in a different order.
Generating smoother skinned objects

While the skinning operations do not include the option to increase the resolution of the objects beyond the resolution found in the profiles, smoother skinned objects can be derived by smoothing the profile shapes before the skinning operation is executed.

Any of the curve types available from the C-Curve tool can be used, however, using one of the Quick Tangent curves is strongly recommended. This is because the tangent curves are the only ones that preserve the positions of the control points. All the other curves generate all new points. The implication of this is that, if you have already properly placed your profiles with the anchor points of the source shapes on top of the corresponding anchor points of the paths, smoothing your profile shapes with a tangent curve will preserve the positions of the anchor points. Smoothing with another curve type, does not preserve the anchor points.

When using placement by anchor points and you have already marked your anchor points, the points where the marks are will be preserved, when using tangent smoothing. The marks themselves will also be preserved. Smoothing with other curve types will not preserve the marked points. This is simply because the original marked points do not exist on the new curve and thus cannot be transferred.

There is one more detail you need to be aware of. When you mark points directly on a c-curve, these markings will be lost if you edit and regenerate the c-curve. This is again a logical consequence of the fact that each time you edit a c-curve new points are created and the previous points do not exist any more.

Examples of both placement by current position and by anchor points, where profiles have been smoothed using tangent curves, are shown in Figures 4.9.3.30 and 4.9.3.31.

Note that smoother (higher resolution) objects can also be derived by using the Q-Subz tool. In this case the skinned object is smoothed after it has been derived, as opposed to smoothing its profile shapes.
Facetted versus nurbz objects

As with the sweep operations, the options to create facetted or smooth skins are available in the Skin Options dialog (Figure 4.9.3.5). Examples are shown in Figure 4.9.3.32.

Facetted Options: When Facetted is selected at the top of the Skin dialog the single option in this group becomes available.

Triangulate: When this option is selected, the facetted object is triangulated.

Smooth Options: When Smooth is selected, the options in this group become available, including the selection of among two types of smooth objects.

Construct As Smooth Skin: With this option on, a smooth parametric object is generated.

Construct As Nurbz: With option selected, a NURBS object is generated from the selected profiles. Its degrees are set by the following parameters.

Length (U) Degree: This parameter controls the degree of the nurbz curve that is derived from the source shapes of the skin. The degree must be greater than zero, and less than the number of points in the source shape. The default degree is 2.

Depth (U) Degree: This parameter controls the degree of the nurbz curve that is derived from the paths of the skin. It must be greater than zero, and less than the number of points in the path. Default is 2.

Cap Start, Cap End: These options determine how the ends of the skin object are treated when the paths are open and the sources are closed. When selected, a cap is generated to close the corresponding end of the object. When both ends are closed, a solid nurbz results (Figure 4.9.3.33).
4.9.4 Lofting

The Loft tool constructs a new smooth object by forming a surface from the boundary of a sequence of source shapes. Lofting is similar to skinning except that the former uses a bulge factor and a direction setting for each source, rather than paths to determine the shape. These settings are selected in the **Loft Options** dialog, shown in Figure 4.9.4.1, that can be invoked from the tool.

The sources can be selected at the segment (edge), outline, or face level, from solid, surface, or wire objects. The sources must either be all closed or all open and can not contain holes. The result of the loft operation is a plain smooth object. Examples are shown in Figure 4.9.4.2.

Either the prepick or postpick method can be used with this tool. With the postpick method, the Loft tool is activated first and then two sources are selected. With the prepick method, any number of edges, outlines, or faces are picked first, followed by the selection of the Loft tool and a mouse click anywhere in the graphics window. The execution of the operation is affected by the following options:

**Bulge**: This group of options determines the magnitude of the curvature of the loft surface as it passes through the sources.

- **Automatic**: When this option is on, the bulge curvature is automatically calculated by the program.
- **Custom**: When this option is on, the bulge curvature is set by parameters entered by the user. This associated slider may be used to set the size of the bulge. A bulge of 0 results in no bulging and a sharp transition. The higher the bulge, the more distinct the effect on the loft. It is recommended that a value of no more than 100 is used. However, larger values can also be entered, if more extreme results are desired. Examples of custom bulges are shown in Figure 4.9.4.3.

Figure 4.9.4.1: The **Loft Options** dialog.

Figure 4.9.4.2: Loft examples between (a) closed shapes and (b) surface edges and open wires.
Direction: This group of options determines the direction of the loft surface as it passes through the sources. Note that the Bulge and Direction parameters are initially applied uniformly to all of the sources, however, when the Preview option is enabled (see Figure 4.9.4.8), the parameters at each source can be individually controlled.

Automatic: When this option is on, the direction and the bulge are automatically calculated by the system to give the smoothest results for the given sources. This option will generate different results when the sources are surfaces or edges of surfaces and different results when they are wires. For surface sources the ends of the loft surface are coplanar to the source surfaces, while for wires of the same shape they are tangent to the wires. This is illustrated in Figure 4.9.4.4. Source edges generate different results that are tangent to the surface edge where the same edge as a wire does not as shown in Figure 4.9.4.5.

Source Plane Normal: When this option is on, the loft direction is perpendicular to the plane of the source. If the source is non-planar, its average plane is used.

Reference Plane Normal: When this option is selected, the loft direction is perpendicular to the current reference plane (to the positive side of the plane).

Reverse: When this option is on, the direction is reversed. For example, with the Reference Plane Normal option, the direction is perpendicular to the negative side of the plane.
**Surface Parameterization:** This pair of options determines the positioning of the surface control points in the V (depth) direction, as shown in Figure 4.9.4.7.

**Use Sources:** When this option is on, which is the default, the orientation of the control points follows the orientation of the sources (Figure 4.9.4.7(b).)

**Use Surface:** When this option is selected, the control points are evenly distributed along the resulting surface (Figure 4.9.4.7(c).) This option is generally slower because it requires additional calculations to distribute the control points along the surface.

**Make Solid:** When this option is selected and the sources are closed, faces are constructed at the first and the last source to make a solid.

**Plain Object, Edit, Adjust To Parameters:** These options work as for all other Derivative tools that create controlled objects. When Edit is on and the Loft tool is applied, the Loft Edit dialog, which is a preview dialog, is invoked. Its options are as follows:

**The Loft Edit preview dialog**

This dialog is partially similar to the Loft Options dialog. Below, only the options unique to this dialog are discussed.

**Source Parameters:** This group of options controls how the parameters of the loft object are handled during editing through the preview dialog.

**Uniform:** When this option is selected, the parameters of all of the sources are edited simultaneously. That is, all the parameters in the preview dialog are applied to all the sources.

**Per Source:** When this option is on, the parameters for each source are edited individually. All of the parameters in the preview dialog are applied to the current source only. The current source is initially the first source. The Previous and Next buttons are used to navigate through the sources. The current source is indicated by the x of y information field, where x is the current source y is the total number of sources. The current source is also displayed in the graphic preview window in the highlight color.
**Direction**: This group of options in the Loft preview dialog is identical with the same group in the Loft Options dialog, except for the following additional option:

**Custom**: This option can be used to specify a custom direction for the source. The direction is entered in the X, Y, and Z text fields. These values, taken together with an implicit 0,0,0 point define a vector, which represents a direction. For example the values X=0.0, Y=0.0, Z=1.0 define a direction along the positive Z axis. Examples are shown in Figure 4.9.4.9.

**Update**: Clicking on this button applies the new parameters and refreshes the preview window. This button is dimmed when the preview is up to date with the parameters.

**Automatic**: When this option is on, the preview is automatically updated every time a parameter is changed. When this option is on, the Update button is dimmed.

**Figure 4.9.4.8**: The Loft Edit dialog.

**Figure 4.9.4.9**: Loft Direction: (a) source wires; (b) Uniform and Direction set to Reference Plane Normal; (c) Per Section and Custom of X=1.0, Y=0.0, Z=0.5 for sources 2 and 4.
4.9.5 Skinned lofting

S-Loft

The S-Loft tool constructs a new controlled object by fitting a surface through a series of source shapes, with or without using explicit or implicit path shapes or guides. This tool can be thought of as a combination of skinning and lofting, from which its name is derived.

The S-Loft tool is affected by settings in the S-Loft Options dialog, shown in Figure 4.9.5.1, which is invoked from the tool. The sources can be selected at the segment, outline, or face level, from solid, surface, or wire objects. The sources must either be all closed or all open and can not contain holes.

Either the prepick or postpick method can be used with this tool. With the postpick method, the S-Loft tool is activated first and then exactly the number of sources specified in the # Sources field are selected. With the prepick method, any number of sources are picked first, followed by the selection of the S-Loft tool and a mouse click anywhere in the graphics window. If options such as Guide Curves, Path, or Branched are on, additional picking is required, which is discussed in detail with the respective options.

There are five types of s-lofts, which are selected from the S-Loft Options dialog:

**Basic**: This type, being the simplest, fits a surface through a series of source curves. If only two sources are given, the resulting surface is ruled. The following additional options may complement the generation of this type of s-loft. An example of this type is shown in Figure 4.9.5.2.

**Draft**: This option provides the ability to control the take-off vectors of the two outer profiles. This is an angle defined off the plane of the source at every point along the skinning profile. In addition to the user supplying the angles by entering values in the Start and End fields, one may also supply a magnitude for the take-off vector. For examples see Figures 4.9.5.3 and 4.9.5.4.
**Figure 4.9.5.2:** (a) Original source objects and (b) an object derived by a Basic s-loft.

**Figure 4.9.5.3:** (a) Original source objects. Basic s-lofts derived with Draft angles Start, End = (b) 0° and (c) 45°.

**Figure 4.9.5.4:** (a) Original source objects. (b) Basic s-loft with Draft angles Start = 45, End = 30.
**Guide Curves:** When this option is on, a method that locally controls the shape in the V direction of the loft surface in between the input sources is applied. The surface follows the guide curves and has a type C1 continuity at every point, even across the guide itself. A guide curve affects the geometry of the surface created between the series of edges to which the curve is attached. No other surfaces are affected.

With this option on, when using the postpick method, with the S-Loft tool active, you first pick the number of sources indicated in the # Sources field and you continue to pick as many guides as indicated in the # Guides field. With the prepick method, with the Pick tool you preselect the number of sources you desire and then, with the S-Loft tool you pick as many guides as shown in the # Guides field.

**Virtual:** When this option option is on, the selected guide has a global effect on the lofted surface. If one guide curve is selected, it is propagated across the profiles to each edge vertex. This effectively extends the control of a guide curve from local to global.

Following are some considerations on selection of guide curves:

- The guide curves must touch each of the profiles selected for the s-loft operation.
- They must start and end on the first and last profiles, respectively.
- Any number of guide curves can be used.
- The guide curves need not be consistent in the V direction but they must not loop.

![Image](image.png)

**Figure 4.9.5.5:** (a) Original source objects. (b) Basic s-loft with Guide Curves on and 2 curves. (c) As in (b) plus Virtual on.

![Image](image.png)

**Figure 4.9.5.6:** (a) Original source objects. (b) Basic s-loft with Guide Curves on and set to 1. (c) Basic s-loft with Guide Curves and Virtual on.

*Modeling • Sweeps, skins, lofts, and caps*
**Ruled**: This type of s-loft places ruled surfaces between each pair of profiles. Any number of source profiles can be selected. If only two profiles are picked, this command defaults to a basic s-loft and generates a ruled surface between the two profiles. An example of a Ruled s-loft is shown in Figure 4.9.5.7.

**Normal**: This type of s-loft provides the ability to constrain the take-off vectors on each profile to the profile’s normal. All profiles must be planar and non-degenerate. No control of the magnitude of the take-off vectors is provided. It determines the magnitudes that yield surfaces with the maximum minimum radius of curvature. Whether the normal effect is applied to all or some of the profiles is determined by a selection from the Normal pop up menu. It contains four items: **First**, **Last**, **First & Last**, and **All**. Examples of s-lofts with different combinations of the normal effect are shown in Figure 4.9.5.8.

**Path**: This type of s-loft provides the ability to constrain the take-off vectors at each profile based on a path curve, which is also picked. The resulting surface does not follow the path exactly; rather a constant vector field is placed on each profile. The vector is defined as the tangent vector of the path curve at the point in which the curve intersects the profile’s plane.

When using the postpick method, the path is picked with the S-Loft tool after the number of sources set in the # Sources field have been picked. When using the prepick method, any number of sources are first selected with the Pick tool. Then the S-Loft tool is activated and a path is picked. An example of this type of s-loft is shown in Figure 4.9.5.9.
**Branched**: This type of s-loft provides the ability to skin multiple profile paths. The user first specifies a profile path for the trunk, consisting of one or more wires. Two or more branches, each of one or more profiles, are specified next. As expected this operation has special picking requirements in order to be able to communicate to the program the exact sequences in which the different branches will be lofted. The picking processes are as follows:

When using the postpick method, with the S-Loft tool active, you first click on the profiles of the trunk (which may be one or more). You signal the program that you are done picking the trunk profiles by double clicking away from a pickable entity after you picked the last profile. Next you proceed with picking as many branches as indicated in the `Branches` field. For each branch you may pick one or more profiles and you signal the completion of the selection of each branches profiles by double clicking in the graphics window. After the last branch has been picked the operation is executed.

Referring to the example in Figure 4.9.5.10, here is how the profiles were picked: With the S-Loft tool active, we clicked on 1 and 2, then double clicked. We continued by clicking on 3, double clicking, clicking on 4, and double clicking. This completed the selection of the profiles for a s-loft with 2 branches. More examples are shown in Figures 4.9.5.11 and 4.9.5.12.

---

**Figure 4.9.5.10**: (a) Original sources and the order in which they were picked. (b) **Branched** s-loft (trunk constructed from 2 profiles, and each of the branches from 1 profile).

**Figure 4.9.5.11**: (a) Original source objects. (b) **Branched** s-loft with a trunk and three branches, each constructed from a single profile.
Closed option

If the closed option is selected, the user would be prompted to pick an additional wire.

Figure 4.9.5.12: (a) Original source objects. (b) A Branched s-loft with six branches.

Figure 4.9.5.13: (a) Original source objects. (b) Branched s-loft with 6 branches (Trunk: 2 wires, each Branch: 2 wires, second one being the same for all).
**Make Solid:** A solid object is created if this option is on, and a surface object is created if it is off.

**Closed:** The closed option may be used when the user needs to construct a solid body closed in v. A solid body is constructed only when all the wires supplied are closed; otherwise this option is ignored. The default is an open (not closed) body. Figure 4.9.5.14 shows an example of a closed skin constructed from four profiles. The surface is continuous at each profile.

**Periodic:** The periodic option allows constructing loft bodies that are periodic in V. This implies that the loft bodies close back on themselves smoothly (continuously) at the start and end profiles.

![Figure 4.9.5.14:](image)

*a* Original source objects.  
*b* Basic s-loft with Closed on.  
*c* Basic s-loft with Periodic on.

**Plain Object, Edit, Adjust To New Parameters:** These options work as for all other controlled objects.
Previewing and editing s-loft objects

When the S-Loft tool is executed with the Edit option on in the S-Loft Options dialog, the S-Loft Edit dialog, shown in Figure 4.9.5.15, is invoked. This dialog offers the opportunity to preview the s-loft object, as well as to experiment with different variations. This dialog contains a standard preview window on the left, the options found in the S-Loft Options dialog, and a few additional options, as follows:

**Update**: Clicking on this button applies the new parameters and refreshes the preview window. This button is dimmed when the parameters of the preview are up to date.

**Automatic**: When this option is on, the preview is automatically updated every time a parameter is changed. When this option is on, the Update button is dimmed.

*Figure 4.9.5.15*: The S-Loft Edit dialog.
4.9.6 Deriving surfaces from boundaries

This tool can be used to create a surface between a closed loop (circuit) of boundary entities. The boundary loop can consist of one or more smooth or facetted segments (edges). The input boundary entities can exist as wire objects, as edges on the same object, or edges on different objects. They can be open ends of single faces or they can belong to two neighboring faces and the boundary loop picked for this tool may be a mixture of these types of edges. The resulting surface is always a smooth object that may optionally be stitched to the original object, under certain circumstances, discussed below. The different ways in which this tool can be applied are determined by options selected from the Cap Options dialog that is invoked directly from the tool. It is shown in Figure 4.9.6.1

This tool can be used either in prepick or postpick mode. When using the prepick method, with the Pick tool active and topological level set to Segment, preselect a closed sequence of segments. Then with the Cap tool active click anywhere in the project window to create the surface. Note that the edges can be selected in any order, but a surface will be generated only when the program can form a closed loop with them. When using the postpick method, activate the Cap tool and pick the boundary edges, which can be in any order. As soon as the program detects that a closed loop can be formed with the selected edges, a surface is created.

The Cap Options dialog contains three tabs two of which (Display Resolution and Status Of Objects) are as in other dialogs. The Options tab is as follows:

**Use Surface Info**: When this option is on, the new surface that will be generated will be tangent to the adjacent surfaces, whenever possible. Tangent continuity can only be maintained when the selected edge belongs to a face and is open. That is, it belongs to no other face. For example, edges selected from a solid, such as a cube, are associated with two faces and are thus not open. No tangent surface can be generated from such edges.

**Tension**: This slider, which is only available when Use Surface Info is on, controls the tension of the tangent surface. Higher tension produces flatter surfaces than surfaces with a low tension.
**Use Wires**: When this option is on, the resulting surface will ignore any surface information from the selected edges, which will be treated as wires. That is, as if they belong to no faces. The resulting surface will not be tangent to any neighboring surfaces.

**Use Cover**: When this option is on, the Cap tool creates the same surface as the Cover tool, discussed in section 4.21.10. This option is only available when the **Use Wires** option is on.

**Stitch**: When this option is on, the resulting surface will be stitched to the adjacent surfaces, whenever possible. Stitching is only possible when the original input edge is associated with exactly one face. For example, it is not possible to stitch when we use edges of a solid. When the input edge belongs to a faceted object, the entire object is converted to a smooth object in order to complete the stitch operation.

**Preview**: When this option is on, which is the default, the **Cap Edit** dialog, shown in Figure 4.9.6.2, is invoked to allow interactive editing of the cap parameters. This dialog has a standard graphic preview window on its left and the options of the **Cap Options** dialog, with a few additional items:

**Reverse**: When this option is selected, the direction cap surface is reversed.

**Update** and **Automatic**: As in other **Edit** dialogs.

Examples of cap surfaces generated from boundaries are shown in Figures 4.9.6.3 through 4.9.6.7.

**Figure 4.9.6.2**: The **Cap Edit** dialog.

**Figure 4.9.6.3**: Surfaces from boundaries:
(1) A closed spline wire and (2) an extruded spline.
(a) Original loops. (b) **Use Surface Info** and (c) **Use Wire** options. Note results for (1): same.

**Figure 4.9.6.4**: Surfaces from edges of four cubes:
(a) Original objects and how edges are picked.
(b) Resulting surface (with original cubes ghosted)
Figure 4.9.6.5: Cap surfaces from boundaries: (1) A parallel extrusion and (2) curvy top end. (a) Original boundary loops at top of objects. (b) Use Surface Info and (c) Use Wire options. Shown in wire frame and smooth rendering.

Figure 4.9.6.6: Cap surfaces from boundaries, which are open edges of four surfaces touching at their corners. (a) Original objects and (b) with the surface generated between them. Shown in wire frame (left) and smooth rendering (right).

Figure 4.9.6.7: Surfaces from boundaries, which are the open loops at the ends of a cylindrical object (pipe). (a) Original object and (b) with the surfaces generated at its ends with the Use Surface Info option. Shown in wire frame (left) and smooth rendering (right).
4.10 Meshes and subdivisions

This section discusses the following five tools that relate to meshes:

- Mesh
- Q-Subz (Quadratic Subdivisions)
- T-Subz (Triangular Subdivisions)
- Reduce Mesh
- Triangulate

These are tools that either generate new meshes on surfaces of objects, thus increasing their resolution or topological density, or reduce the mesh resolution of an object.

The meshes generated by the first tool are more specifically called *plain meshes* to distinguish them from the *controlled, nurbz*, or *patch meshes*. They are rectangular meshes. In contrast, the meshes generated by the Q-Subz and T-Subz tools are not. While the former are also four-sided, their shape depends on the shape of the surface that is subdivided to produce them. They can also be curved as they are generated. T-Subz generates triangular meshes, which can also be curved.

The Reduce Mesh tool in a way does the opposite of what the first three tools do. It deletes parts of a mesh to reduce its density. It does so on the basis of angles between faces or at edges.

The last tool generates a special type of a mesh that is specifically intended for recovering the planarity of faces that may have been disturbed. Given that many other programs are only capable of handling three and four-sided faces, and certainly not faces with true holes, which is a *form-Z* trademark, the ability to triangulate such faces makes them exportable to other programs.
4.10.1 Meshes

These are one of the types of meshes that can be generated in form\textsuperscript{Z}; they are rectangular tiles generated on the complete surfaces of an object or on individual faces of objects. They are referred to as plain meshes and they are always flat. Even though there is a tool that specifically generates smooth meshes (see next subsection), the plain meshes can also be applied as a first step towards the smooth deformation of a surface. That is, after a mesh has been generated on a flat surface, the Move Mesh or Deformation tool can be used to smoothly curve the surface. These operations are discussed in section 4.11.

![Mesh]

This tool is used to generate rectangular meshes on the surface of an object or on a single face of an object. Using the postpick method, a mesh can be generated by selecting the entity to be meshed while the Mesh tool is active. The entity picked for meshing can be a complete object (surface or solid) that is selected at the Object topological level, or the face of an object, selected with topological level set to Face. When more than one entity is to be meshed with a single operation, the prepick method may be used. With the Pick tool, you select any number of entities, at the Object or Face topological levels, then, with the Mesh tool active, you click anywhere in the graphics window.

The density and the positioning parameters of the mesh are set through the Mesh Options dialog (Figure 4.10.1.1), which can be invoked from the Mesh tool.

\(\text{X, Y, Z:}\) These options control the direction in which the mesh will be generated. It can be generated in all three directions (X,Y,Z), in a single direction, or in any combination of two directions. This is controlled by these switches, all of which are on by default. The distance at which the mesh lines are placed is controlled by the values entered in the X, Y, and Z fields. All three default to 4 feet (or 1 m in metric).

**All Directions:** When all the directions are not on, selecting this option turns them all on.

**XYZ Lock:** When this option is on, all three distance parameters are locked to the same value. That is, if one changes, they all change.

![Mesh Options]

\textbf{Figure 4.10.1.1:} The Mesh Options dialog.
The meshes produced by the Mesh tool are always orthogonal. Their orientation is determined through one of two methods selected from the **Mesh Options** dialog.

**Mesh Direction:** These options determine the method that will be used to establish the orientation of the mesh.

**From Picked Segment:** When selected, the direction of the mesh is determined by the segment picked to select the object or face on which the mesh will be generated. When a face is selected, if the **Clicking On Edges** pick option is used, the direction is determined by the segment selected first. If the **Clicking Inside Boundaries** pick option is used, the direction is determined by the segment that is closest to the click point. This is the direction of the X axis of the mesh. The Y and Z axes are perpendicular to the X axis. As illustrated in Figure 4.10.1.2, for the same face, the mesh can take as many distinct directions as the number of sides with distinct directions.

**From Angle:** When this option is selected, the direction of the mesh is determined by the slope of the line defined by the values entered in the **Altitude** and **Azimuth** fields. **Altitude** is the angle measured from the +X axis (3 o'clock) on the ZX plane. **Azimuth** is the angle measured from the +X axis on the XY plane. The altitude and azimuth angles define a line in 3D space. When a mesh is generated on the surface of a face, the projection of the 3D line on that surface is used to determine the direction of the mesh.

This option is intended to facilitate the generation of meshes with the same direction on different objects, and to allow the directions of the meshes to be independent of the directions of the segments of an object. This is illustrated in Figure 4.10.1.3. With **Altitude** = 0°, **Azimuth** = 30°, and the shapes drawn on the XY plane, clicking on four different segments (where marked with the dots), produces meshes that have the same direction.
Note that when the **From Angle** option is used, the 3D line defined by the azimuth and altitude angles is projected onto the surface that will be meshed, and the direction of the projected line determines the direction of the mesh. It is possible that a 3D line may project to a single point. For example, if both the azimuth and altitude angles are set to 0°, which defines a line coincident with the X axis, a mesh generated on a surface parallel to the XY or ZX plane will work fine. However, if an attempt is made to generate a mesh on a surface parallel to the YZ plane (such as the surface shown in Figure 4.10.1.4), the system will be unable to determine a direction since the 3D direction line projects to a point. In such cases an error message is issued and the mesh is not generated.

When the **From Angle** option is used to generate a mesh on the face of a solid, the direction of the mesh is calculated as for the surface objects: by projecting the direction line onto the face. This is illustrated in Figure 4.10.1.5 (a) and (b), where Azimuth=Altitude=30° is used. Note that how the faces are selected has no effect on the **From Angle** option.

When a mesh is generated on a complete solid object, one of its faces is used as the basis for determining the direction of the mesh. Which face is used is determined by the segment where you click to select the object. Clicking on different segments will produce different mesh directions. This is illustrated in Figure 4.10.1.5 (c), (d), (e), and (f), where the click points are marked with dots. Whenever it is desired to generate meshes with the same direction on a number of solid objects, then the objects should be selected in a manner such that the base faces used for the generation of the meshes are parallel (such as all parallel to the XY plane, etc.).

As already illustrated, the generation of a mesh on a complete object and the generation of a mesh on a surface (face) are executed differently. In the first case, a three dimensional mesh is generated. In the second case, the mesh is two dimensional. That is, for the complete object, after the position of the orthogonal axes are established on the basis of the picked segment, the increments of the mesh are applied in directions parallel to the axes for all directions. If the faces of the object are not parallel to the orthogonal axes, the dimensions of the mesh patches will be distorted. In contrast, when faces are meshed individually, 2D meshes are generated on their surfaces and the dimensions of the patches are preserved.
The **Mesh Options** dialog contains four options which offer variations for placing the mesh relative to a face or an object.

**Normal Alignment**: When this option is selected (default), the X axis is placed at the minimum y value and the Y axis at the minimum x value of the face. The mesh origin is where the two axes intersect, and the mesh is generated incrementally in the X and Y directions, until the maximum x and y values are reached.

**Center On Line**: When this option is selected, the origin is placed at the center of the face, and the mesh originates from that point.

**Center Between Lines**: When this option is selected, the origin is also placed at the center of the face, but the first mesh lines are placed at half their distance from it, and the mesh is centered on that point.

When generating a mesh on a complete object these options have the same effect, except that the origin is placed at either the minimum x, y, and z point, or at the centroid of the object.

**Fit Increment**: The previous options tend to produce tiles that are not full sized at the perimeter of the face or object. If this is undesirable, then this option can be used to cause the size of the tiles to be adjusted so that all the tiles have exactly the same dimension in each of the X, Y, and Z directions. However, the tiles will be clipped by the boundaries of the entity being meshed, and will appear in their full size only when the object is rectangular. The mesh positioning options are illustrated in Figure 4.10.1.6. Figure 4.10.1.7 illustrates how the four alignment options available in the **Mesh Options** dialog are applied when the **From Angle** option is used.

By default, the tiles of the mesh are orthogonal. A **Triangulate** option is also available in the **Mesh Options** dialog. When selected, the mesh tiles are decomposed into triangles.
Examples

The generation of meshes on faces is illustrated in Figure 4.10.1.8 for both a surface object and a solid object. In the latter example, after a mesh was generated for the front face of the cube, the increment parameters were reduced and another mesh was generated on a face which was produced by the previous application of the Mesh operator. This operation was repeated twice.

Surfaces or complete objects that contain holes can also be freely meshed. This is illustrated in Figure 4.10.1.9. In both examples the entities to which the Mesh operator was applied were picked as objects.

Any type of a 3D form can be meshed, as the example in Figure 4.10.1.10 illustrates. Note how the orthogonal mesh mixes with the original edges of the object of revolution. If the resulting structure is not desirable, more control of the directions of the mesh lines can be achieved by meshing each face individually.

As Figure 4.10.1.11 illustrates, a meshed object can be meshed again, in a different direction or at a different density. The example shows a hexagonal extrusion that was meshed three times, each time clicking the mouse on a different segment.
4.10.2 Quadratic subdivisions

Quadratic subdivisions are one more type of mesh that can be generated in form•Z. While they share some similarities with the plain meshes in that they both increase the face resolution of an object, subdivisions use a distinctly different meshing algorithm. In addition, the geometric positions of their points may be adjusted as part of the meshing operation. While plain meshes superimpose an orthogonal grid over an object or face, subdivisions always work at the face level and subdivide a face recursively up to a user specified depth.

**Q-Subz (Quadratic Subdivisions)**

The Q-Subz tool can be used to change the resolution of an object by subdividing its faces. It subdivides a face by splitting it with lines that start at its center and end at the midpoints of its boundary edges, as shown in Figure 4.10.2.1.

![Figure 4.10.2.1: Subdivision of faces for the Q-Subz operation.](image)

Note that in the example, the triangular face (a) is subdivided once, the square face (b) is subdivided twice, the pentagon (c) three times, and the octagon (d) four times. Also note that, regardless of the initial shape, once it is subdivided, it consists of 4-sided faces.

Using the postpick method, the subdivision operation is executed by selecting an object or a face with the Q-Subz. The prepick method can also be used to subdivide more than one entity, which can be complete objects or faces. You use the Pick tool to preselect any number of entities and then, with the Q-Subz tool active, you click anywhere in the window to execute the operation. A subdivided object is a parametric object. It remembers the original object and also the parameters that were used to subdivide it. Thus, it can be subsequently edited and revised by applying the Q-Subz tool again.

The parameters that affect the Q-Subz tool can be selected from the Q-Subz Options dialog (Figure 4.10.2.2), which can be invoked directly from the tool.
**Smoothing:** When this option is on, the curvatures of the faces derived from the subdivisions are adjusted according to the angles at their edges. The new faces become tangent to the angles of the original edges and vertices of the object. An example is shown in Figure 4.10.2.3. When this option is off, the mesh generated by the face subdivision remains flat, as in Figure 4.10.2.4.

**Curvature:** This slider bar can be used to adjust the amount of curvature of the mesh. At 50%, which is the default, the curvature will be evenly distributed throughout the object. At 0% there will be less curvature around the centers of the original faces and more around the edges. A 100% value results in the opposite (more curvature around the centers and less around the edges). Figure 4.10.2.5 shows meshes with curvatures of 0%, 50%, and 100%.

**Meshing:** The options in this group determine the mesh resolution of the subdivisions.

**Max # Of Subdivisions:** The value entered in this text field indicates the maximum number of subdivisions that will be applied to a face. This number of subdivisions will actually be applied, unless one of the other options restricts them. It is recommended that you be conservative with this number, since even a small number can generate a large number of faces. For example, a four sided face, subdivided 8 times will result in $4^8 (= 65,536)$ faces. It is also recommended that you restrict excessive numbers of faces by specifying additional restrictions through the following options.
Max Segment Length: When this option is on, the subdivision of a face is terminated as soon as all the segments of a face derived from the original face are less than the value specified in the text field. This option supersedes the maximum number of subdivisions specified by the previous option and subdivision terminates before the maximum number is reached.

Max Face Angle: When this option is selected, the subdivision of a face is terminated, as soon as all the angles between faces derived from the original face are less than the value specified in the text field. This option will generate a denser mesh in areas of strong curvature and a sparser mesh in areas where the curvature of the mesh is close to flat. An example is shown in Figure 4.10.2.6.

Triangulate Mesh: When this option is selected, the resulting mesh is triangulated. The triangulation method used here is analogous to the fourth (far right icon) that is selectable from the Triangulate Options dialog, invoked from the Triangulate tool (see section 4.10.5). All the faces of the mesh are triangulated, regardless of whether they are planar or not. To triangulate it, each face is subdivided by inserting a point at its center and connecting it to the corners of the face. This triangulation method handles the curvatures of the subdivided surfaces more accurately. An example is shown in Figure 4.10.2.7.

Edit: When this option is on, and the Q-Subz operation is executed, the Q-Subz Edit dialog, shown in Figure 4.10.2.8, is invoked and allows you to preview the results of the operation, before accepting them. The parameters in the Edit dialog are actually the same with those in the Q-Subz Options dialog and can be changed while previewing your object. When this option is off, the subdivisions are applied directly.

Adjust To New Parameters: When this option is selected, the subdivision parameters stored with an object from a previous subdivision operation are overwritten by the parameters currently specified in the Q-Subz Options dialog. If this option is off, the old parameters are maintained.

As already mentioned, when Edit is selected in the Q-Subz Options dialog, as soon as you execute the operation, the Q-Subz Edit dialog, shown in Figure 4.10.2.8 appears. This dialog lets you preview the results of your operation and your settings. It contains the same options as the Q-Subz Options dialog and, in addition, it has a preview window with typical view manipulation tools and two more options that determine what you will be previewing.
**Original Object:** When this option is on, the unmeshed object is shown in the preview.

**Meshed Object:** When this option is selected, the meshed object is displayed. If any of the smooth meshing parameters are changed since this option was last selected, the updated version of the object is generated first and then displayed.

**Update:** When you click on this button, the updated version of the meshed object is generated and displayed in the graphics window. This button is only active if the **Meshed Object** option is selected and a parameter was changed since the object was last displayed.

The graphic preview of the object can be edited with the standard view manipulation icons. In addition, this preview window includes the arrow tool, for selecting/deselecting individual faces.

Faces can be selected/deselected with the Arrow tool and either one or two mouse clicks, depending on whether the **Clicking On Edges** or **Clicking Inside Boundaries** option is selected in the **Pick Options** dialog.

Pressing `command + A` (Macintosh) or `option + shift + A` (Windows) selects all the faces. Pressing `command + tab` (Macintosh) or `option + shift + tab` (Windows) deselects all faces.

To select faces that may be behind another face, the pick parade method can be used, as discussed in subsection 4.3.3.

When individual faces are selected rather than the complete object, only these faces are subdivided. Examples of objects with all and only some of their faces subdivided are shown in Figure 4.10.2.9. In this example, object (c) was derived by selecting all but its top and bottom faces.

**Figure 4.10.2.8:** The Q-Subz Edit dialog.

**Figure 4.10.2.9:** (a) Subdividing an object by selecting (b) all and (c) only some of its faces.
4.10.3 Triangular subdivisions

Triangular subdivisions are one more method available in form\textsuperscript{-}Z for generating meshes. They are similar to the Quadratic subdivisions in that they both increase the face resolution of an object, by subdividing its faces. However, each uses a different subdivision algorithm. The former is based on a subdivision logic that results in rectangular faces. The latter always produces triangular faces. They both adjust the geometric positions of their points, which results in smoothly curved surfaces.

**T-Subz (Triangular Subdivisions)**

The T-Subz tool can be used to increase the resolution of an object by subdividing its triangular faces. It subdivides a face by connecting the midpoints of each edge of a triangular face resulting in four new faces. The next level of subdivision is created by applying the same process to the four newly created triangles. This process continues recursively until the desired result is achieved, as shown Figure 4.10.3.1. In the example, the first triangle (a) is subdivided once, the second (b) is subdivided twice and the third (c) is subdivided three times. The last triangle (d) is subdivided four times resulting in 256 triangles from the original triangle.

The T-Subz tool only works on triangular faces. All non-triangular faces are first triangulated before subdivision begins, as shown in Figure 4.10.3.2. Note that in the example, the initial triangulation is in bold and all the shapes were subdivided just once.

Using the postpick method, the subdivision operation is executed by selecting an object with the T-Subz tool. The prepick method can also be used to subdivide more than one object. You use the Pick tool to preselect any number of entities and then, with the T-Subz tool active, you click anywhere in the window to execute the operation. A subdivided object is a parametric object. It remembers the original object and also the parameters that were used to subdivide it. Thus, it can be subsequently edited and revised by applying the T-Subz tool or the Query tool.
The parameters that affect the T-Subz tool can be selected from the **T-Subz Options** dialog (Figure 4.10.3.3), which can be invoked directly from the tool.

**Smoothing:** This option controls which type of smoothing is applied to the object. The smoothing determines how the geometric position of the points are calculated. The magnitude of the smoothing is controlled by the **Curvature** setting described below.

- **In:** This type of smoothing smooths the object by moving the points inward from its original shape as shown in Figure 4.10.3.4 (a). This smoothing is based on the Loop approximation subdivision algorithm. Under this scheme, the smoothing of the surface approximates a three directional box spline surface. The more levels of subdivision that are applied, the closer approximation of the spline surface.

- **Out:** This type of smoothing smooths the object by moving the points outward from the original shape as shown in Figure 4.10.3.4 (b). This smoothing is based on the Butterfly interpolation subdivision algorithm. With this type of smoothing, only the newly introduced points created by the subdivision are repositioned. The original points of the object remain in their original location. Since the points are derived by interpolation, the resulting surface does not represent a parametric surface.

**Curvature:** This slider bar can be used to adjust the amount of curvature of the subdivision mesh. At 100%, which is the default, the object is smoothed to its maximum curvature. At 0% there is no curvature and the faces retain their original shape. Settings between 100% and 0% result in proportional curvature as shown in figure 4.10.3.5.
**Triangulate By Center:** When this option is on, the initial triangulation uses a method which inserts a point in the center of the face to create triangles rather than just connecting the existing edges. This method helps to keep the original shape of the face throughout the subdivision. Note that for concave faces, a point will be added in each area of concavity. Examples of this option are shown in figure 4.10.3.6. This option is off by default.

**Limit To Surface:** When this option is on, the mesh points are computed as if the surface had reached its theoretical limit (infinite subdivision). This option is useful when it is desirable to have the points be geometrically on the spline surface without over meshing the object. Note that this may yield a rougher looking surface at lower levels of subdivision because the geometric positions of mesh points are pushed to locations that would only be naturally derived at higher levels of subdivision. This option is only available when the **Smoothing** option is set to **In**.

**Boundary Type:** This setting determines how the boundaries of surface objects are handled in a T-Subz object. This setting does not affect solid objects. Figure 4.10.3.7 shows examples of all three types

- **Natural:** When this option is selected, the boundaries are determined by the natural result of the subdivision process using the Smoothness setting (In or Out) and the curvature setting. This option is on by default.

- **Cubic:** When this option is selected, the boundaries are constructed as cubic b-spline curves.

- **Linear:** The boundaries are not smoothed and they retain their original linear nature.
**Meshing:** The options in this group determine the mesh resolution of the subdivisions.

**Max # Of Subdivisions:** The value entered in this text field indicates the maximum number of subdivisions that will be applied. This number of subdivisions will actually be applied, unless one of the other options restricts it. It is recommended that you be conservative with this number, since even a small number can generate a large number of faces. For example, a cube, subdivided 5 times will result in 12,288 faces (12 triangles x $4^5$). It is also recommended that you restrict unnecessarily excessive amounts of faces by specifying additional restrictions through the following options.

**Max Segment Length:** When this option is on, the subdivision of a face is terminated as soon as the length of all the segments of a face are less than the value specified in the text field. This option supersedes the maximum number of subdivisions specified by the previous option and subdivision terminates before the maximum number is reached.

**Max Face Angle:** When this option is selected, the subdivision of a face is terminated as soon as all the angles between faces derived from the original face are less than the value specified in the text field. This option will generate a denser mesh in areas of strong curvature and a sparser mesh in areas where the curvature of the mesh is close to flat.

**Max Face Area:** When this option is selected, the subdivision of a face is terminated as soon as the area of the face is less than the value specified in the text field. This option will generate a denser mesh in areas of strong curvature and a sparser mesh in areas where the curvature of the mesh is close to flat.

---

**Figure 4.10.3.8:**
T-Subz meshing examples: (a) Original object, (b) meshed with **Max # Of Subdivisions** = 3, (c) meshed **Max # Of Subdivisions** = 3 + **Max Segment Length** = 4'-0", (d) meshed with **Max # Of Subdivisions** = 3 + **Max Face Angle** = 30°, (e) meshed with **Max # Of Subdivisions** = 3 + **Max Face Area** = 600.
It is important to note that the meshing parameters can affect the realization of the curvature of the object. This is most dramatic when using subtle (smaller) curvature settings because it takes more subdivisions to get enough points evaluated to see the effect of the curvature. In general, the less curvature applied, the more levels of subdivision are necessary to see the results. For example, Figure 4.10.3.9, illustrates the different results for different levels of subdivision. A curvature of 35% is used in this model.

![Meshing Examples](image)

**Figure 4.10.3.9:** T-Subz meshing examples: (a) Original object, meshed with **Maximum # Of Subdivisions** (b) = 1, (c) = 2, (d) = 3, (e) = 5 + **Maximum Face Angle** = 30°.

**Edit:** When this option is on, and the T-Subz operation is executed, the **T-Subz Edit** dialog, shown in Figure 4.10.3.10, is invoked and allows you to preview the results of the operation, before accepting them. The parameters in the **Edit** dialog are actually the same with those in the **T-Subz Options** dialog and can be changed while previewing your object. When this option is off, the subdivisions are applied directly.

**Adjust To New Parameters:** As with Q-Subz.
Previewing and editing T-Subz objects

T-Subz objects can be previewed before their generation when Edit is selected in the T-Subz Options dialog. When you execute the operation, the T-Subz Edit dialog, shown in Figure 4.10.3.10 appears. This dialog lets you preview the results of your operation and your settings. It contains the same options as the Q-Subz Options dialog and, in addition, it has a preview window with typical view manipulation tools, a reset option for the curvature and a set of options to control the preview window. This dialog can also be invoked from the Edit button in the Query Object dialog.

![Image of T-Subz Edit dialog]

**Figure 4.10.3.10:** The T-Subz Edit dialog.

**Reset:** This button resets the curvature for all edges and points of a T-Subz object which may have been edited using the Edit Controls tool (see next section). If all edges and points have the same curvature, this item is dimmed and the curvature slider is enabled. If the object contains different curvatures at the edges and points, then this button is enabled and the curvature slider bar is disabled. Upon selecting this option, the curvature for the object is made uniform, the button becomes disabled and the slider bar becomes enabled.

**Preview:** This group of options controls the display of the preview window.

**Show Original Object:** When this option is on, the un-meshed object is shown in the preview in the project’s ghost color.
**Show Subdivision Surface**: When this option is selected, the current version of the meshed object is displayed.

**Update**: When you click on this button, the updated version of the meshed object is generated. This button is only active if a parameter was changed since the object was last updated.

**Automatic**: When this option is selected, the meshed object is automatically updated each time a parameter is changed in the dialog.

Some of the parameters of the T-Subz object can be edited graphically through the Edit Controls tool (see section 4.3.7). The editable controls are the points of the original control object and the curvature at each edge. In addition, the curvature at each original point can be changed for T-Subz objects that use the smoothing in method.

To edit the controls, select a T-Subz object with the Edit Controls tool. This displays the T-Subz controls as shown in Figure 4.10.3.11. The original object which represents the control net is shown in the primary control editor color. The curvature controls are shown in the secondary editor color. The curvature controls are positioned Perpendicular to each edge of the original shape. For T-Subz objects that use the smoothing in method, there are also curvature controls at each point of the original shape. This control is not available for T-Subz objects that use the smoothing out option because the original points of the object do not change.

To reposition a point of the original object using the click-and-drag method, click on a control and move the mouse while pressing the mouse button. To use click-and-click method, first select as many controls as you wish to reposition and then click on one of the selected controls and move it to the desired position. Figure 4.10.3.12 shows the edit controls after repositioning 4 of the points.

![Figure 4.10.3.11: T-Subz edit controls](image)

![Figure 4.10.3.12: Repositioned control points](image)
All T-Subz objects are initially created with a uniform curvature determined by the Curvature setting in the **T-Subz Options** dialog when the object is created. The curvature of the mesh can be controlled by adjusting the curvature in the area of the mesh influenced by an edge or point of the original object. The curvature controls consist of a line and a circle along the line. These controls function like slider bars in a dialog. As the circle is moved along the line, the curvature is adjusted moving the mesh in the direction that the circle control is moved.

For T-Subz objects that use the smooth in method, when the circle is at the end of the line farthest from the edge or point, this area will have 0% curvature and the mesh will have a hard edge or point. When the circle is at the end of the line closest to the edge or point, the area of the mesh will have 100% curvature and will be smooth. Positions between the ends yield proportional results. For T-Subz objects that use the smooth out method, the control is opposite. That is, when the circle is at the end of the line farthest from the edge, the edge will have 100% curvature and when the circle is at the end of the line closest to the edge, the edge will have 0% curvature.

To edit a curvature control with the click-and-drag method, click on a curvature control and move the mouse while pressing the mouse button. To use click-and-click method, first select as many curvature controls a you wish to edit and then click on one of the selected controls and move it to the desired setting. When editing multiple smooth controls, all of the selected controls will get the same setting as the one that is tracked by the mouse.
Figure 4.10.3.14: Smoothing a face model using T-Subz:
(a) Original cage object, (b) meshed with **Smoothing In** and **Maximum # Of Subdivisions = 1**, (c) meshed with **Smoothing In** and **Maximum # Of Subdivisions = 4 + Maximum Face Angle = 10°**, (d) meshed with **Smoothing Out** and **Maximum # Of Subdivisions = 1**, (e) meshed with **Smoothing Out** and **Maximum # Of Subdivisions = 4 + Maximum Face Angle = 10°**.
4.10.4 Reducing the resolution of meshes

The tools discussed in the two previous subsections generate meshes and so do a variety of other form•Z operations, which essentially increase the resolution of an object. Some times, there is also a need to reduce the resolution or density of a mesh, which can be done with the following tool.

Reduce Mesh

The Reduce Mesh tool can be used to reduce the number of faces of an object, by merging adjacent faces that enclose an angle that is less than a specified threshold. It can also be used to reduce the number of segments in a face by merging adjacent segments. The Reduce Mesh tool can be executed at the Object or Face topological levels. If executed at the Object level, all faces of the object are considered. If one or more faces are selected, only these faces are optimized. This tool can be applied using either the postpick or prepick selection method and the latter is necessary when a number of faces need to be selected.

The parameters that affect the Reduce Mesh tool are selected in the Reduce Mesh Options dialog (Figure 4.10.4.1), which is invoked directly from the tool.

Face Angle: The value entered in this field determines, whether two adjacent faces can be merged. If their normals enclose an angle which is smaller or equal to this value, the two faces are merged into a single face.

Edge Angle: The value entered in this field determines, whether two adjacent segments can be merge. If their normals enclose an angle which is smaller or equal to this value, the two segments are merged into a single segment. Only segments which do not connect to another face (boundary segments), are considered.

Merge Faces With The Same: This group of options contains matching conditions that can be selected to reduce the merging of faces.

- Color: When this option is selected, only faces which have the same color and enclose a small enough angle are merged.
- Texture Map Control: When this option is selected, only faces which are part of the same Texture Map Control and enclose a small enough angle are merged.

Show Preview: When this option is on and the Reduce Mesh operation is executed, the Reduce Mesh Preview dialog, shown in Figure 4.10.4.2, is invoked, allowing you to preview the results of the operation, before accepting them. When this option is off, the Reduce Mesh operation is executed immediately.
The **Reduce Mesh Preview** dialog contains the same options as the **Reduce Mesh Options** dialog and, in addition, it has a preview window with its typical view manipulation tools. It also has three more options, labeled **Show Preview**, that determine what you will be previewing, and two information fields.

**Show Original Object**: When this option is selected, the original, non optimized object is shown in the preview window.

**Show Reduced Object**: When this option is selected, the optimized object is displayed. If any of the parameters in the dialog have been changed since this option was last selected, the updated version of the object is generated first and then displayed.

**Update**: When you click on this button, the optimized version of the object is generated and displayed in the preview window. This button is only active if the **Show Reduced Object** option is selected and a parameter was changed since the object was last displayed.

**# Of Faces Merged, # Of Edges Merged**: These are information fields that display how many faces and edges have been merged by the current settings. The information is updated every time a change is made to the dialog.

The graphic preview of the object can be edited using the standard view manipulation icons, located under it. They are discussed in subsection 2.1.6 of the form•Z User’s Manual. In addition, this preview window includes the arrow tool, which can be used to select/deselect individual faces. This is done as in the **Q-Subz Edit** dialog, discussed in the previous subsection. When individual faces are selected rather than the complete object, only the topology of the selected faces is reduced.

The Reduce Mesh tool is frequently used to eliminate unnecessary topology and can thus be viewed as a resolution optimization procedure. Some topology can be considered unnecessary if the shape of a 3D model remains the same after this topology has been eliminated. Sometimes the term “necessary” is also relative and depends on the scale at which a model is rendered or how it is used. For example, models intended for some types of animation, such as games, work better when they are of a relatively low resolution.

Frequently the Reduce Mesh operation can be viewed as a complement to other operations, many of which have to apply a procedural regularity that may result in unnecessarily high mesh density.
One example would be the objects of revolution, as shown in Figure 4.10.4.3. As the same number of rotations have to be applied uniformly to all the segments of the source profile, the angles of faces that are generated from segments closer to the axis are sharper than the angles of faces created from segments that are farther away from the axis. This implies that for the same degree of smoothness more steps of revolution would be required for the closer to the axis segments. While the revolution process can only be applied uniformly, the Reduce Mesh operation can be applied after the revolution to balance the mesh density of the revolved object, as shown in Figure 4.10.4.3(b).

Another example is shown in Figure 4.10.4.4. In (1) a rectangular mesh has been shaped using a profile and the Move Mesh tool (see next section). The mesh movement is by necessity applied uniformly along the curved shape of the profile, regardless of the curvature and smoothness requirements at different points of the curved mesh. Applying a mesh reduction procedure adjusts the density of the mesh, as shown in 4.10.4.4(b). The same process is illustrated by the example in (2), except that the shape of this mesh was derived by applying a circular wave disturbance.

Figure 4.10.4.3: Reducing the mesh resolution of an object of revolution: (a) before and (b) after.

Figure 4.10.4.4: Reducing the resolution of (a) a mesh to which mesh move was applied, and (b) a mesh to which a wave disturbance was applied. (1) Before and (2) after the reduction at Face Angle = 5°.
Other modeling operations that by definition apply procedures that result in higher than the necessary mesh resolutions would be grid based terrain modeling (see section 4.6) and the metaformz (see section 4.16). Figures 4.10.4.5 and 4.10.4.6 show examples of resolution reductions applied to such models.

**Figure 4.10.4.5:** Reducing the mesh of a terrain model. (a) The original model. Reduced with Face Angle set to (b) 1°, (c) 2°, (d) 3°, (e) 4°, and (f) 5°.
Figure 4.10.4.6: A metaformz dinosaur model before and after mesh reduction.
(a) Before: 51,462 faces, 103,164 edges, and 51,704 points.
(b) After: 9,788 faces, 23,707 edges, and 13,955 points.
4.10.5 Triangulating

Triangulation is another rather special type of meshing. It is the process that subdivides a face into triangular pieces. Triangulation guarantees planarity, since three point shapes are always planar. Consequently, the most common use of the triangulation operation is for recovering the planarity of faces that may have been disturbed. Some of the object generation tools, such as the Terrain Modeling and the C-Mesh tools, have a Triangulate option available to them. When it is not used, the resulting object can still be triangulated after it has been generated, using the tool discussed in this subsection.

**Triangulate**

This tool is used to apply triangulation to either complete objects or individual faces. Triangulation is by its nature a face based operation. When applied to complete objects, its faces are essentially triangulated individually. Both the postpick and prepick selection methods can be used with the Triangulate tool. When using the post pick method, you click on the entity you wish to be triangulated, with the Triangulate tool. This can be a group, an object, or a face. When using the prepick method, with the Pick tool you preselect any number of faces and/or objects, you then activate the Triangulate tool and click anywhere in the project window. Groups of objects are triangulated as individual objects. Executing the operation with the topological level at entities other than Face, Object, or Group results in a beep and an error message. The Triangulation operation is applied to the object itself rather than to a copy of an object. If a copy is desired, you should make a copy of the object before triangulating it.

You may instruct the program to triangulate the non-planar faces only, or all faces regardless of planarity. This and other options are selected in the Triangulate Options dialog (Figure 4.10.5.1), which is accessible directly from the Triangulate tool. There are also options for selecting a triangulation method.

**Non Planar Faces Only:** When this option is selected, only the non planar faces of an object are triangulated. This option is on by default.

**Strict Planarity Test:** This option is only available when Non Planar Faces Only is on. When selected, the triangulation process is more sensitive to smaller disturbances in the planarity of a face than it is when this option is off. In general, selection of this option will cause more faces to be triangulated. This option is off by default.
**Triangulate All Faces:** When this option is selected, all the faces of an object will be triangulated, regardless of their planarity.

**Triangulation Method:** The four icons in this group of options represent four different methods for applying triangulation. These are the same methods used for exporting (see subsection 3.13.2). A summary of the results produced by the four triangulation methods when applied to different shapes is shown in Figure 4.10.5.2.

Examples of triangulation applied to complete objects are shown in Figures 4.10.5.3 and 4.10.5.4.
4.11 Deformations, displacements, and morphing

The following tools are discussed in this section:

- Define Profile
- Move Mesh
- Disturb
- Deform
- Bend Curve
- Morph
- Displace

All these tools represent operations that reshape the geometry of meshed objects, typically in a continuous and/or smooth fashion.
4.11.1 Defining profiles

Areas of meshes can be moved smoothly, using the shape of a preset profile. A profile is a 2D open shape which is drawn in the usual manner, and then stored in the Profiles Library, which is accessible through the Profiles palette (Figure 4.11.1.1). Before the execution of a mesh movement, a profile is selected from the Profiles palette to be used for the movement.

**Define Profile**

This tool is used to define profiles and store them in the profiles library. Profiles are dimensionless (parametric) shapes that are created from open surface (line) objects. You first draw an open shape on any reference plane, but preferably on the XY plane. If not already open, you open the Profiles palette (from the Palettes menu) so that you may see your profile being generated. You then set the topological level to Object, select the Define Profile tool, and click on the object you wish to be used for the creation of the profile. You will see your profile generated and placed in the first vacant position of your Profiles palette.

You can also use the prepick method to generate more than one profile with a single execution of the operation. Preselect the objects you wish to be used for the generation of profiles, select the Define Profile tool, and click the mouse anywhere in the graphics window. You will see your profiles generated, one at a time, and placed in the Profiles palette. The drawings used for profiles can only be picked at the object level and should be open surface objects. If any of these requirements are violated, the system beeps and issues an error message. In prepick mode, an error message will be issued for each item found in violation of these requirements and the program will proceed with the next object selected.

The profile is dimensionless and is always shown filling a square bounding box. That is, regardless of how you draw the shape, it will be adjusted to the dimensions of a square. This is illustrated in Figure 4.11.1.2. The dimensions of the profile are actually insignificant, since they will be assigned by the dynamic process applied when moving meshes.

**Figure 4.11.1.1:** The Profiles palette.

**Figure 4.11.1.2:** (a) A shape as drawn, (b) as adjusted to the dimensions of a square, and (c) the profile created from it.
The profile is half of the shape that will actually be used for mesh movement. The complete shape is created by mirroring the profile about a vertical axis. That axis is shown in the Profiles palette as a dashed red line on the left side of the profile box. How the mesh movement shapes are created from profiles is illustrated in Figure 4.11.1.3. These are also examples of proper ways to draw shapes for profiles. In contrast, the drawings in Figure 4.11.1.4 are not proper and they have to be adjusted before they can be used as profiles. The program applies the following process:

• A bounding box is constructed around the drawing. The vertical left side of the box is the axis of symmetry that will be used to produce the final shape for the mesh movement. The point of your drawing that touches the left side of the box (this is the point with the minimum x) becomes the first point of the profile and any points before that are clipped (Figure 4.11.1.4(a)). Best results (intuitively understandable) are obtained when that point is one of the end points of your drawing.

• The drawing is also clipped at the lower (horizontal) side of the bounding box, if necessary (Figure 4.11.1.4(b)). This happens when the drawing turns inwards (x decreases), after the point with the minimum y (this is the one that touches the lower side of the box).

• Concavities relative to the Y direction will also be “filled” by connecting the outer (right) most points, as illustrated in Figure 4.11.1.4(c).

• If the last point of the profile is not on the lower side of the bounding box (the horizontal axis) a vertical line is inserted extending from the last point to the horizontal axis (Figure 4.11.1.4(d)). See also examples in Figure 4.11.1.5.
Shapes to be used for profiles can be drawn on any reference plane. However, the profiles are defined in the X and Y directions and any shape drawn on a plane other than the XY will be adjusted to x and y coordinates.

Once a profile is defined, it becomes part of the form·Z Profiles library that is accessed through the Profiles palette and the Profiles dialog (Figure 4.11.1.6). When profiles are created the program assigns names by using the word “profile” and a sequential index. You can change a name by typing another one in the Name field. The dialog also contains Width and Height fields that are used to assign dimensions to a profile, before it is used for mesh movement. The initial default values for width and height are derived from the actual dimensions of the shape used for the profile. The Profiles dialog also contains a Delete command, for deleting profiles, and a Maintain Proportions option. When selected, the x/y ratio of the original drawing is preserved. That is, when either the height or the width is changed, the other is automatically updated according to the original x/y ratio.

The Profile Library is automatically retained by the system from session to session. It is stored in the file named “form·Z Extensions.” Recall that this file should be kept in the same folder with your form·Z application.
### 4.11.2 Moving meshes

Areas of meshes can be moved smoothly, using the shape of a preset profile, which is a 2D shape stored in and selected from the Profiles palette, as discussed in the previous subsection. While the profile controls the shape of the “height” of the motion, the shape that bounds the points that are moved is controlled by the shape drawn at the time the movement is applied. It can be linear or circular.

#### Move Mesh

While the Move tool can be used to move points individually, the Move Mesh tool can be used to move a group of points simultaneously and smoothly, following the form of a geometric shape. That shape is derived from the active profile. This tool can be thought of as simulating *clay modeling*, or the movement of a soft surface. Although it can be applied to any object, this tool is most useful when applied to a meshed surface.

There are two basic steps to the mesh movement. The first step is to identify the area of the mesh that is to be moved. The second step is to actually move the mesh by stretching or pushing the profile into the mesh. There are three basic types of mesh movement, determined by the shape used to delineate the area of the mesh that will be affected: the **radial** movement uses a polygonal area, the **linear** movement uses a linear path drawn at the time the move is executed, and the **from path** movement, which is also of a linear character, uses a previously drawn open or closed 2D shape. The type of mesh movement as well as other parameters are selected from the *Move Mesh Options* dialog, shown in Figure 4.11.2.1. It consists of two tabs: **Options** and **Triangulate Options**. The content of the latter is as that of the dialog invoked by the Triangulate tool (see subsection 4.10.5).

![Figure 4.11.2.1: The Move Mesh Options dialog.](image)
**Movement Direction:** These two options control the direction in which the movement is applied.

**Relative To Reference Plane:** This option is the default and produces a mesh move that is relative to the active reference plane. If a **Radial** move is applied and the Perpendicular switch (window tool palette) is on, the movement is perpendicular to the reference plane, which is most frequently the desired result. If the Perpendicular switch is off the move is in a direction parallel to the reference plane. With **Linear** and **From Path** moves, the state of the Perpendicular switch is ignored and the move is always executed in a direction perpendicular to the reference plane.

**Perpendicular To Base Point:** When this option is selected, the program uses as a reference plane the plane that passes through the picked point and is perpendicular to its normal (think of the normal as the line in the middle of the 3D angle of the point). This movement is always perpendicular to the plane (or in the direction of the normal) and ignores the state of the Perpendicular switch.

**Movement Shape:** This group of options control the shape of the mesh movement. When editing, each shape of mesh move (radial, linear, or from path) behaves differently.

**Radial:** This shape is the default and produces mesh moves that are centered around the selected point. The radial move also uses the number entered in the **# Of Sides** field to determine what polygon will be used for the movement. The default is an 8-sided polygon (octagon). The dynamic process of the radial mesh movement is illustrated in Figure 4.11.2.2.

After a profile has been selected in the Profiles palette, the Move Mesh tool has been activated, and **Radial** has been selected in the **Move Mesh Options** dialog, the radial mesh move begins by clicking on a point of the mesh object. The point clicked defines the center of the mesh area to be moved. A rubber banded polygonal shape appears on the screen and follows the motion of the mouse. A second click of the mouse completes the area definition stage and initiates the 3D movement stage. The 2D polygon is replaced by a 3D bell shaped preview form, which is rubber banded and follows the motion of the mouse. This shape is constructed by revolving the active profile around an axis that passes through the center of the polygon. The preview shape is a simplified representation of what the mesh will look like after the movement is applied. Assuming that the **Reference Plane** option is selected for the direction of the move, if the Perpendicular switch is off, the shape will be “stretched” along the reference plane. If it is on, the mesh will move in a direction perpendicular to the plane. A final click of the mouse causes the program to regenerate the mesh object with its points moved.

![Figure 4.11.2.2](image)

**Figure 4.11.2.2:** Radial movement: (a) a point is selected, (b) a polygonal shape is rubber banded, (c) a bell shaped form is rubber banded, and (d) the mesh is regenerated with its points moved.
The preview 3D bell shape can also be manipulated as it is rubber banded by using keys on the keyboard. These operations are illustrated in Figure 4.11.2.3.

- Pressing **option** (Macintosh) or **ctrl+shift** (Windows) while moving the mouse changes the radius of the preview shape.
- Pressing **control** (Macintosh) or **ctrl+alt** (Windows) while moving the mouse changes the rotation or orientation of the preview shape.
- Pressing **option+control** (Macintosh) or **ctrl+shift+alt** (Windows) allows the preview shape to be scaled along the reference plane, independently in each direction.

**Figure 4.11.2.3:** Manipulating the preview shape: (a) Changing the radius. (b) Rotating the preview shape. (c) Scaling the shape.

**Linear:** When this option is selected, a linear shape is applied to the mesh move. The dynamic execution of a linear mesh move is illustrated in Figure 4.11.2.4.

The *linear* mesh movement begins by clicking on a point of the mesh object to be used as the start of the linear movement. A line, starting at that point, is rubber banded and follows the motion of the mouse. A sequence of lines can be drawn in the usual manner. The linear shape is completed by double or triple clicking. A double click creates an open line. A double click on the starting point or a triple click creates a closed shape. Note that the first point has to be on a mesh point since the program needs it to determine where to draw. The other points of the line can be outside the mesh.

**Figure 4.11.2.4:** Linear mesh move: (a) After placing first point on mesh. (b) Line is drawn to second point. (c) Preview shape is rubber banded, and (d) mesh is regenerated with points moved.
As soon as the line is drawn, the system rubber bands a preview shape constructed by sweeping the profile along the line. The **Width** of the active profile as set in the **Profiles** dialog is used for the preview shape. The linear mesh move is always in a direction perpendicular to the reference plane and the Perpendicular switch has no effect. The **option** key on the Macintosh or **ctrl+shift** on Windows can be used to dynamically change the width of the linear preview shape. Moving the mouse up along the perpendicular direction increases the width; moving it down decreases it. The next mouse click completes the operation and modifies the mesh object to reflect the linear movement.

**From Path:** This mesh movement functions identically to the linear mesh movement except that a predrawn 2D shape is used as the path, rather than a dynamically drawn line. With this option, the mesh object is selected first, followed by the selection of the path shape. The rest of the process is as for the linear movement. The from path method is shown in Figure 4.11.2.5.

![Figure 4.11.2.5: Mesh move from path: (a) Mesh and line object are selected, (b) mesh preview is used to stretch the shape, and (c) mesh is re-generated with points moved.](image)

**Movement Interface:** The mesh movement can either be applied directly in one step or it can be executed in a dynamic fashion.

**Edit:** Selection of this option, which is the default, rubber bands the mesh movement and allows you to inspect the form produced before deciding exactly where you want your points to be positioned.

**Generate Directly:** This option applies the mesh motion in one step and the resulting form is presented immediately. If not satisfactory, the operation can be undone and executed again.

**Move Points:** The options in this group affect the behavior of the mesh move in areas of the mesh that are not flat. This includes the cases where one mesh move is executed on top of a previously applied move. These options are illustrated in Figure 4.11.2.6.
Points Under Profile Only: When this option is selected (default) the move only affects points that are actually under the height of the profile, at the time the rubber banding of the preview shape is completed, assuming an upward motion has been applied. The reverse applies for downward motions. Points above (or under) the profile are not affected.

All Points Within Profile: This option moves all points of the mesh that fall within the area covered by the preview shape, in an absolute manner. That is, the points are moved to positions corresponding to the shape of the profile.

Relative To Original Position: This option moves all the points of the mesh that fall within the move area, but in a relative fashion. That is, the height of the corresponding profile point is added to the current height of the mesh point.

![Images of Move Points options](image)

*Figure 4.11.2.6: Move Points options: (a) Mesh and lines to be used. (b) Points Under Profile option, (c) All Points Within Profile option, and (d) Relative To Original Position option.*

Movement Type: The two options in this group determine how the movement will vertically affect the mesh. They are illustrated in Figure 4.11.2.7.

Stretch: When this option is selected (default), the entire profile is proportionally scaled and its proportions are used to stretch the mesh.

Push: When this option is selected the mesh is pushed up as the profile is moved vertically and only the part of the profile that penetrates the mesh is actually used for the movement. This method preserves the proportions of the profile as defined by the Height and Width values in the Profiles dialog. If the mesh is pushed up higher than the height of the profile, the base of the mesh stretches to the base of the profile shape.

*Figure 4.11.2.7: Movement Type options: Progression of heights with (a) Stretch and (b) Push.*
**Move Points Of Front Faces Only:** When this option is selected (which is the default), only points of faces that face the same direction as the normal to the picked point (are on the same side of the input point) are moved. Selection of this option filters out the points of the opposite side of a meshed object, which typically are not intended to move. When this option is off, all the points within the profile shape move, regardless of the orientation of the faces on which they fall.

**Random Point Disturbance:** This option is used to give the mesh a rougher appearance by randomly applying an additional small motion to the points of the mesh. These “disturbances” are perpendicular to the reference plane. If **Apply To Moved Points Only** is selected, only the points affected by the mesh move are disturbed. The whole mesh surface is affected otherwise. The **Min** and **Max** fields specify the range of the disturbances. That is, a number between the minimum and the maximum values is randomly generated for each point of the mesh and is used to move that point.

Mesh movement can be applied to any object, but its results are only meaningful when the area of the surface where it is applied contains relatively dense points. The mesh movements are primarily intended for meshed surfaces and objects, either plain or controlled. That is, the curved surfaces of c-meshes can further be manipulated using the Move Mesh tool. However, the general principles of applying geometric transformations to parametric objects are in effect. That is, as soon as a c-mesh is edited, the points of the moved mesh return to their previous positions. Consequently, whenever a mesh move is appropriate to be applied to a c-mesh, it should be applied at the very end of the mesh sculpting process and the c-mesh should not be regenerated again, or should be turned into a plain mesh before the mesh movement is applied.

**Examples**

A number of mesh move examples are shown in Figure 4.11.2.8. Those on row (a) illustrate single applications of radial moves applied at the center of a meshed surface (the top face of a cuboid). Distinctly different profiles have been used (round, pointed, and flat) to underline the versatility available for “imprinting” different forms onto the surfaces of meshes.

The examples on row (b) illustrate multiple applications of radial moves. In many cases the center of the move was not placed at the center of the mesh surface but rather towards its boundary or even on its boundary. For example, the first (left) example on row (b) was constructed through four radial moves, all centered on corner points of the cuboid. In these cases, part of the polygonal shape that determines the area to be mesh moved is outside the boundaries of the meshed surface, which is perfectly acceptable. The mesh move will affect only the points that are within the polygonal (or linear) shape of the move area.

The examples in the third (c) row illustrate multiple applications of linear mesh movements. Whether the lines for the mesh moves were drawn dynamically or were predrawn is rather insignificant. Either method can be used to obtain the same results. Which **Move Points** options have been used in each case should be obvious. If it is not and you wish to reproduce the examples, try them in different ways and observe the results.
Figure 4.11.2.8: Mesh movements and the profiles used for their execution.
(a) Single application of radial movement.  (b) Multiple applications of radial movement.
(c) Multiple applications of linear movement.
4.11.3 Point disturbances

While the geometry of the points of a mesh can be randomly disturbed when a mesh movement is applied (see previous subsection), points of an object can also be deformed by geometrically disturbing its points. When a disturbance is applied to an entire object, the disturbance parameters are retained as a deformation so that the parameters can be edited and mixed with other deformations. That is, when a disturbance is applied to an object for the first time, the object’s type is changed to a deformed control object. The deformation control retains the list of deformations for the object. When additional deformations are applied to an object, they are added to the end of the deformation list. The deformation list can be edited to add, remove, and re-order the list. The parameters of each deformation can also be numerically edited.

**Disturb**

The Disturb tool changes the appearance of an object by applying a disturbance, or designed movement, to some or all of the points of an object. The disturbance can be applied in either a *random* fashion, or in a mathematically defined pattern referred to as a *wave*. The random method can be used to introduce roughness or irregularity to an object. The wave method can be used to introduce a repetitive undulation or pattern with predictable results. This operation, especially the random disturbance, will often generate many non-planar faces in the object; however, the option to triangulate the disturbed faces is also available (see below).

The point disturbance operation is controlled by the **Disturb Options** dialog (Figure 4.11.3.1), which can be invoked by double clicking on the Disturb tool.

The disturb points operation works at any topological level, and can be used with either the postpick or prepick method. When the postpick method is used, with the Disturb tool you click on the desired entity. The location of the pick point relative to the selected entity may be significant, depending on the options selected in the **Disturb Options** dialog.

With the prepick method, any number of entities can be picked at any topological level using the Pick tool, followed by the selection of the Disturb tool. Clicking anywhere in the project window initiates the disturbance to the points of the picked entities.

![Figure 4.11.3.1: The Disturb Options dialog.](image)
**Model Type:** This group determines the type of model that is created from the deformation operation. Both faceted and smooth objects can be deformed and the type can be changed based on the desired effects of the operation.

**Maintain:** When selected, the result object is the same type as the original object's type. This is the default.

**Facetted:** When a smooth object is selected with the facetted option selected, the smooth object will be converted to a facetted object. It is often useful to use the facetted option during exploration until the desired results are achieved before switching to the smooth option which is more computationally intensive.

**Smooth:** When a facetted object is selected with the this option selected, the object will be converted to a smooth object. This does not work well if the original object is a highly facetted mesh as smooth deformations are computational intensive and work best on objects with lower face counts. Examples are shown in Figure 4.11.3.2.

**Facetted Options:** This group of options applies when facetted objects are deformed. Note that there are no analogous options when deformations are applied to smooth objects.

**Mesh:** When this option is selected, the object is automatically meshed as it is deformed. The new mesh facets are created relative to the deformation axis and the density of the facets is based on the magnitude of the deformation. That is, a deformation which causes a more dramatic change introduces more facets and deformation which causes a more subtle change introduces less facets.

**Density:** This slider allows for overall control over the number of facets created. When the slider is positioned to the right, more facets are created and when it is positioned to the left, less facets are created. Examples are shown in Figure 4.11.4.

**Triangulate:** When this option is on, the non-planar faces of facetted objects are triangulated.

*Figure 4.11.3.2:* Disturbance model type: (a) original faceted object, (b) with the **Maintain** option on, and (c) with the **Smooth** option on.
**Method**: This menu lists the available disturbances, as shown in Figure 4.11.3.3. The deformation selected from the menu is applied when the Disturb Tool is used. Each method has options which are displayed in the area below the menu.

**Disturb Points**: When this option is selected, a random point disturbance is applied to each point of the selected entities by moving it a random distance relative to the active reference plane. The X, Y, and Z options can be used to independently determine if the points will be disturbed in the respective direction, relative to the active reference plane. Each point will be disturbed by adding a random displacement which is greater than Min, and less than Max. The value entered in the Min field must always be less than that entered in the Max field, and vice versa. The Lock option keeps the values in the Min and Max fields identical, as shown in Figure 4.11.3.4.

![Figure 4.11.3.3: The Method pop up menu.](image)

**Disturb Point Normals**: This option disturbs points along their normals, rather than relative to the reference plane. This method produces a disturbance which is less destructive to the form of the original object. The Min and Max fields determine the range of the disturbance that is applied along the point’s normal, as shown in Figure 4.11.3.5.

![Figure 4.11.3.4: Random point disturbances: (a) original mesh. Random point disturbances applied along (b) the X axis, (c) the Y axis (d) the Z axis, and (e) along all three axes.](image)

**Figure 4.11.3.5**: Random disturbance: (a) Cylinder before it is disturbed and after with (b) Disturb Points method and (c) Disturb Point Normals method.
**Linear Wave**: When this option is selected, a linear wave disturbance is applied to the selected entities. This disturbance generates a sine wave that undulates along the X axis in a direction perpendicular to the base reference plane. The distance that a point is moved is determined by its distance from the *reference point*.

**Through Centroid**: This option determines the location of the reference point. When this option is off, which is the default, the point that was used to pick the entities is also used as the reference point. When using the prepick method, the location of the click that activates the command is used as the reference point for all the prepicked entities. This is useful if you wish to apply a uniform wave to multiple entities. When the *Through Centroid* option is selected, the centroid of the object to which the selected points belong is used as the reference point.

**Height**: This field determines the height, or amplitude, of the wave. This is the vertical distance between the highest and lowest points of the wave, measured in a direction perpendicular to the reference point. An example is shown in Figure 4.11.3.6.

**Width**: This field determines the width, or frequency, of the wave. This is the distance of one cycle, or between two peaks, of the wave, as shown in Figure 4.11.3.6.

**Offset**: This field determines the offset of the wave from the reference point. A zero value indicates that the wave should pass through the reference point as shown in Figure 4.11.3.7(a). The offset values is related to the width, or frequency, of the wave. If offset value is equal to 1/4 of the wave width, the peak of the wave will pass through the reference point as shown in Figure 4.11.3.7(b). If the value is 1/2 of the width, the wave will pass through the reference point at the opposite slope to a zero offset value as shown in Figure 4.11.3.7(c). If offset value is equal to 3/4 of the wave width, the valley of the wave will pass through the reference point as shown in Figure 4.11.3.7(d).

![Figure 4.11.3.6: The Height and Width parameters of a wave.](image)

![Figure 4.11.3.7: (a) offset = 0, (b) offset = 1/4 of width, (c) offset = 1/2 of width, and (d) offset = 3/4 of width.](image)
**Circular Wave**: When this option is selected, a circular wave disturbance is applied to the selected entities. This disturbance generates a sine wave that undulates radially from the reference point in a direction perpendicular to the base reference plane. The distance that a point is moved is determined by its distance from the *reference point*. All of the options are as with the **Linear Wave**.

*Figure 4.11.3.8*: Wave disturbances: (a) The original mesh and the pick point. (b) **Linear** wave with **Through Centroid** on. **Circular** wave with **Through Centroid** (c) off and (d) on.

*Figure 4.11.3.9*: **Circular** wave applied to a single planar face with **Model Type** set to **Smooth** and **Through Centroid** (a) off and (b) on.
**Spread**: When this option is selected, the object is spread open by making each face into an individual surface and moving them away from the spread origin point. The spread origin point is the objects origin with optional offset values. The faces are moved along the vector created between the origin point and the center of the face. This operation will cause solid objects to be converted into surfaces. When applied to a smooth object, convex faces are split into multiple faces. For example as a cylindrical face is split in two halves as shown in Figure 4.11.3.10(c).

**Offset**: This group controls the origin of the spread as it offsets in X, Y, and Z relative to the object's axis. The origin of the spread is the point which all of the faces spread away from or to.

**Distance**: This is the distance the faces are moved away from the reference point.

**Base Reference Plane**: This option allows you to specify the reference plane that will be used as the base plane for the operation. The **Object (XY)** option is the default, and uses the reference plane defined by the object's X and Y axes. The **Object (YZ)** and **Object (ZX)** use the plane defined by the object's Y and Z axes or the object's Z and X axes respectively. The **Active Plane** option uses the active reference plane from the active window. The XY, YZ, and ZX options use the corresponding Cartesian reference planes. These options are useful when it is desirable to work in a projection view, and the projection reference plane is not the appropriate reference plane for the operation.

**Plain Object**: When this option is selected a plain object is constructed instead of a controlled deformation object. This option is off by default.

**Edit**: When this option is selected, the **Deformation Edit** dialog is invoked after the graphic creation of the deformation is completed. The **Deformation Edit** dialog is used to add, remove and re-order the deformation list and edit the parameters of each deformation in the list. This is described in the next section.
4.11.4 Deforming objects

Deformations are special types of operations used to reshape the form of a facetted or smooth object using mathematical formulas or geometric algorithms. The deformation operations and parameters are retained so that the parameters can be edited to change the form of the object. That is, when a deformation is applied to an object for the first time, the object's type is changed to a deformed control object. The deformation control retains the list of deformations for the object. When additional deformations are applied to an object, they are added to the end of the deformation list. The deformation list can be edited to add, remove, and re-order the list. The parameters of each deformation can also be numerically edited.

Deform

This tool is used to apply a deformation to an object, whose appearance changes by applying a shear, taper, twist, bulge, radial shear, radial bend, or Bezier bend deformation to it. These types and other options that affect their execution are set in the Deform Options dialog, shown in Figure 4.11.4.1, which is invoked directly from the Deform tool. It consists of three tabs: Options, Display Resolution, and Status Of Objects.

The portion of the object that is affected is determined by the deformation box, which appears as soon as you click on an object with the Deform tool. This box initially bounds the complete object but it can be adjusted to only cover part of the object.

The deformations are based on a reference plane referred to as the base plane. The deformation axis is the line perpendicular to the base plane, through the center of the object. The deformation box is centered around the deformation axis, and extends to the limits of the object. The deformation box can be rotated and its upper and lower limits can be changed; however, it will always enclose the object relative to the base plane.
All of the points of the object that are within the deformation box are affected by the object deformation. Points outside the deformation box are never deformed, but their position in space may change as they follow the parts of the object that were deformed.

To deform an object, first select the object to be deformed. The system will immediately construct and display a deformation box which bounds the complete object. The position and/or size of the deformation box may be modified before the deformation operation is dynamically applied. The dynamic action varies depending on which type of deformation is applied.

When using the postpick method, select the Deform tool and click on the object to be deformed. With the prepick method, using the Pick tool, preselect the objects or groups to be deformed, then select the Deform tool and click anywhere in the project window.

The deformation box is shown around the selected object in the graphics window. If more than one object is preselected, either a deformation box will be generated for each of the picked objects, or a single box will be generated for all the objects, depending on whether the **All Objects At Once** option is selected. When the option is on, a single box is generated as if the objects selected had been joined together as a single object as shown in Figure 4.11.4.2.

At this point, the deformation box can be edited. It can either be rotated around the deformation axis, or the lower and upper limits can be repositioned. To edit the deformation box, press and hold the *option* key (Macintosh) or *ctrl+shift* (Windows), and select an edge of the deformation box. As you position the cursor on the edges of the deformation box, the cursor will change to indicate which operation will be applied when you click on that edge.

### Rotating the deformation box:

This icon indicates rotation. The cursor changes to this icon when you hold down the *option* key (Macintosh) or *ctrl+shift* (Windows), and position the cursor on one of the vertical edges. Clicking on one of these edges begins a dynamic rotation of the deformation box around the deformation axis. The next click completes the dynamic rotation. When the selected object is not symmetric relative to the deformation axis, the deformation box stretches and shrinks as the rotation is applied, such that it encloses all points of the object. Rotating the deformation box essentially rotates the base plane on which the deformation box is based as shown in Figure 4.11.4.3.
Moving the top of the deformation box:

Moving the bottom of the box:

The cursor changes to these icons when you press the option key (Macintosh) or ctrl+shift (Windows) and position the cursor on one of the top or bottom edges of the box parallel to the reference plane. Clicking on one of these edges begins dynamic movement parallel to the deformation axis. The next click completes the move. The lower limit can not be moved above the upper limit, nor can the upper be moved below the lower. Any number of such movements can be applied before the deformation is applied as shown in Figure 4.11.4.4.

Deform from point:

Deform from edge:

The cursor changes to these icons when, without pressing any key, it is positioned on a corner point or an edge, respectively, of either the upper or lower end of the deformation box. Clicking on one of these edges or points will begin the dynamic deformation. The dynamic action for each deformation is different and is described below.

With the exception of the Bulge and Radial Shear operations, selecting an upper versus a lower edge is significant. This selection defines the end of the deformation box that moves during the deformation, while the opposite end remains fixed.

Selecting an edge versus a corner point is significant for the Shear, Taper, Bulge, and Radial Shear operations, which can either work in both the X and Y directions (relative to the base plane) simultaneously, or be restricted to only one of the X, Y directions. When you click on a point, you deform in both X and Y. Clicking on an edge deforms in only one direction. The remaining operations can only work in one direction at a time. Once the desired form is obtained, a final click completes the dynamic action of the deformation box, and the object(s) itself is deformed.
The Deformation Options dialog

The following options in this dialog (Figure 4.11.4.1) determine the specifics of the deformation to be applied:

**Model Type:** This group determines the type of model that is created from the deformation operation. Both facetted and smooth objects can be deformed and each type will result in different effects. Examples are shown in Figure 4.11.4.5.

**Maintain:** When this option is selected, the result object will be the same type as the original object's type. This is the default.

**Facetted:** When this option is on and a smooth object is picked, the smooth object will be converted to a facetted object. It is often useful to use this option for explorations, until the desired results are achieved, before switching to the Smooth option, which is computationally more intensive.

**Smooth:** When this option is on and a facetted object is picked, the object will be converted to a smooth object. This does not work well when the original object has a dense mesh, because smooth deformations are computationally intensive and work best on objects with lower face counts. Examples are shown in Figure 4.11.4.6.

**Facetted Options:** This group applies when facetted objects are deformed. Note that there are no analogous options for smooth objects.

**Mesh:** When this option is on, the object is automatically meshed as it is deformed. The new mesh facets are created relative to the deformation axis and the density of the facets depends on the deformation. That is, deformations that cause more dramatic changes introduce more facets, while those that cause more subtle changes introduces less facets.

**Density:** This slider allows you to control the number of facets created. Right generates more and left less facets. Examples are shown in Figure 4.11.4.7.

*Figure 4.11.4.5:* Deformations applied to a (a) facetted and (b) smooth object.

*Figure 4.11.4.6:* A facetted cuboid is twisted and also converted to a smooth model.
**Triangulate**: When this option is on, the non-planar faces of facetted objects are triangulated.

**Split Faces At Limits**: When this option is selected, the faces of the source object are split along a new edge that corresponds to the limits of the deformation. This is option is necessary when deforming an object with only few large faces. If the source object is of sufficient density of faces, this option may be disabled and the source object will retain the same number of faces after the operation. It is often desirable to use this option when animating deformations as the original objects topology is preserved. This option is on by default.

**The deformation methods**

**Method**: This pop out menu lists the available deformations. The deformation selected from the menu is applied when the Deform Tool is used. Each deformation type has distinct options, which are displayed in the area below the menu only when the respective deformation is selected. These options are discussed below with each type.

On the right of the Method menu, there is a preview box in which the selected deformation is described graphically. The image displayed changes when a new type is selected. The preview box can also be used to change the method selected. Each time you click in the preview box, the selection advances to the next method.

---

*Figure 4.11.4.7:* Meshing a cube that is deformed: (a) The original facetted cube. (b) Bend at 10°, (c) at 90°, (d) at 90° with low density meshing, and (e) at 90° with high density meshing.

*Figure 4.11.4.8:* The deformation Method menu.
**Shear**: When this item is selected, a shear deformation is executed, which can be applied in the X or Y directions, or in both. **Shear** deforms the object by moving the points at the selected end, while keeping the points at the opposite end fixed. As the cursor is moved, the selected edge moves along a plane parallel to the base plane. The points between the two ends are transformed proportionally. When the **Through Center** option is selected, the points at the center remain fixed, and both the upper and lower ends move. That is, the deformation is anchored at the midpoint of the deformation axis, rather than at the upper or lower limit. During dynamic editing, the state of this option can be toggled (inverted) by holding down the **shift** key. Points above or below the deformation box are moved to align with the lower and upper ends, respectively, as shown in Figure 4.11.4.9.

*Figure 4.11.4.9: Shear deformations:*
Shear (a) moving the top end in the X direction, (b) the top end in the Y direction, (c) the lower end in X and Y, (d) the top end in the X direction with **Through Center** option on, and (e) with the top and bottom ends of the deformation box moved, moving the top end in the Y direction.
**Taper**: When this item is selected, a taper deformation is executed, which can be applied in the X or Y direction, or in both simultaneously. The Taper operation works by stretching the points at the selected edge, while keeping the points at the opposite end fixed. The points between the two ends are scaled in proportion to their distances. When the Through Center option is selected, the center stays fixed, and both the upper and lower limits are stretched. As the cursor is moved, the selected edges are stretched along a plane parallel to the base plane. Points below the lower limits and above the upper limits of the deformation box are stretched to stay in alignment with the respective limits as shown in Figure 4.11.4.10.

*Figure 4.11.4.10: Taper deformations: Taper (a) upper limits in X direction, (b) upper limits in Y, (c) lower limits in X and Y, (d) in X with Through Center selected, and (e) upper limits in X and Y, after both the lower and upper limits were moved.*
**Twist**: This operator deforms an object by twisting it around the deformation axis. The twist is applied by twisting the points of the selected edge, while the points at the opposite edge remain fixed. As the mouse is moved around the deformation axis, the shape twists in either a clockwise or counterclockwise direction. Each complete revolution about the deformation axis adds 360° of twist to the object. The dynamic action works much like a rubber band. For example, if the shape is twisted 720° (twice around) clockwise, you have to “unwind” 720° counterclockwise to get back to the original position. When the **Through Center** option is selected, the middle of the object stays fixed, and the opposite end rotates in the opposite direction from the selected end. The points that are above or below the twist are rotated to align with the two ends of the deformation box as shown in Figure 4.11.4.11.

![Twist deformations](image)

*Figure 4.11.4.11: Twist deformations:* Twist (a) upper limits counterclockwise, (b) upper limits clockwise, (c) lower limits counterclockwise, (d) upper limits clockwise with **Through Center** selected, and (e) upper limits counterclockwise after upper and lower limits are moved.

When applied to a smooth object, the twist has an additional option for controlling the continuity between twisted and untwisted sections of the object. The **Alignment** menu offers three options: **Positional (G0)**, **Tangent (G1)**, and **Curvature (G2)**. The default is **Tangent (G1)**. An interpolation function is used to generate a linear, cubic, or quintic polynomial to obtain G0, G1, or G2 continuity. Figure 4.11.4.12 shows examples of these options.

![Smooth Twist Alignment](image)

*Figure 4.11.4.12: Smooth Twist Alignment:* (a) **Positional (G0)**, (b) **Tangent (G1)**, and (c) **Curvature (G2)**.
**Bulge:** This operation deforms an object by moving points toward or away from the deformation axis along a radial profile. The points at the middle of the object are stretched out, while the ends remain fixed. The points in between are scaled based on their distance from the deformation axis and distance from the ends or center of the deformation box. Points directly on the deformation axis are unaffected. This operation can be applied in the X or Y direction, or both. Selection of the upper versus lower edges is not significant, since they react together. As the cursor is moved, the object bulges outward when the cursor moves away from the axis, and inward when it moves toward the axis. When the **Through Center** option is selected, the points at the middle of the object stay fixed, and both the upper and lower limits stretch together. The parts above or below the limits are affected only when the **Through Center** option is on. In this case, these points are moved to match the alignment of the object at the limits, as shown in Figure 4.11.4.13.

*Figure 4.11.4.13: Bulge* deformations: Bulge (a) upper limits outward in X direction, (b) upper limits inward in Y direction, (c) upper limits in X and Y, (d) upper limits in Y with **Through Center** selected, and (e) in X, after upper and lower limits are moved.
Radial Shear: This deformation is similar to the Bulge operation, except that instead of the parts moving symmetrically toward or away from the deformation axis, they all move in the same direction. This operation can be applied in the X or Y direction, or both. Everything else about this tool is identical to the Bulge. Figure 4.11.4.14 shows an example.

Figure 4.11.4.14: Radial Shear deformations: Radial Shear (a) outward in X direction, (b) inward in Y, (c) outward in X and Y, (d) in X with Through Center selected, and (e) in X with upper and lower limits moved.
**Radial Bend**: This operation bends the shape about an axis parallel to the base plane. The selected edge defines the direction of the bend, and the end of the deformation box that is affected. The opposite end stays in its fixed position. As the cursor is rotated around the fixed end of the deformation box, the bending angle increases, and the effective bending radius is reduced. This maintains the length of the deformation axis so that no distortion occurs along this axis. The greater the angle of bending, the smaller the radius becomes. The bending stops at 360°, and, like the **Twist**, to be unwound it must be rotated in the opposite direction. The bending angle is determined as the cursor moves in a plane that is perpendicular to the base plane, and passes through the deformation axis. If the **Through Center** option is selected, the middle of the deformation box is used as the center, and both ends are bent toward the center. The points that are beyond the fixed ends of the deformation box are not affected; however, the points that are beyond an end that moves are rotated to maintain their original relationship to that end, as shown in Figure 4.11.4.15.

![Radial Bend deformed shapes](image)

*Figure 4.11.4.15: Radial Bend* deformations: Radial Bend (a) clockwise in X direction from top edge, (b) clockwise in Y from top edge, (c) counterclockwise from bottom edge, (d) in X with **Through Center** option on, and (e) in X with upper and lower limits moved.
**Bezier Bend**: This operation applies a Bezier deformation to the object. This is the most flexible and least restrictive of the deformation operations. Bezier bending deforms the object by applying a Bezier curve through the moved points of the deformation axis. To achieve this, the deformation axis is divided into three segments, representing four control points of the Bezier curve. The first and last control points are the two end points of the deformation axis. The other two points are placed along the axis between the end points. The default position of these points is an even spacing with each of the points 1/3 the distance along the axis from the previous point. Options are available to place these points at different positions.

The edge of the deformation box selected determines the direction of the editing. All movement is made along a plane perpendicular to the base plane and through the deformation axis. The selected end of the deformation box moves freely with the cursor while the other end remains fixed.

As the cursor is moved along the plane, the corresponding control point of the Bezier axis is moved, and the shape is deformed to follow it. Different effects can be achieved by moving different control points of the Bezier curve (Figure 4.11.4.16). This is achieved by pressing a variety of key commands, after the motion of the mouse begins the deformation, to achieve a variety of results:

- **no key**: The selected point moves on line.
- **shift**: The selected point and the opposite point move together.
- **control** (Macintosh) **ctrl+alt** (Windows): The end points stay fixed, and the middle two points move together.
- **caps lock**: By default, the Bezier Bend operation applies a quadratic Bezier, which touches the midpoint of the deformation axis. A cubic Bezier can also be applied by pressing the **caps lock** key down.

---

*Modeling • Deformations, displacements, and morphing*
Figure 4.11.4.16: Bezier Bend deformations: Bezier (a) movement of bottom end with caps lock, (b) movement of top end with option / ctrl+shift keys, (c) movement with control / ctrl-alt keys and caps lock, (d) movement with command / ctrl keys and caps lock, (e) movement of top end with upper and lower limits moved.

Parallel: This option applies only to the Bezier Bend. It keeps the upper and lower limits of the deformation box parallel to the base plane. The points of the object that are beyond the limits of the deformation box are adjusted to align with the new position of the upper and lower limits (Figure 4.11.4.17).

Figure 4.11.4.17: Bezier Bend deformations with Parallel option on: actions as in the previous Figure produce results as shown.
**Bezier Controls:** The two text fields to determine the initial location of the Bezier control points along the deformation axis. The **Lower** field shows the percentage of the distance along the axis at which the lower control point is located. The **Upper** field is the same for the upper control point. The defaults are 33% and 66%, respectively. Moving these defaults closer together creates sharper bending at the middle of the object, and farther away creates flatter bending at the center. Examples (with control key and caps lock pressed) are shown in Figure 4.11.4.18.

**General options**

**All Objects At Once:** When this option is selected, which is the default, all prepicked objects and groups are deformed simultaneously. When this option is off, each object is deformed individually, with the deformation of one object starting as soon as the previous is completed. When the **Join** option is on, all the objects are joined into a single object.

**Base Reference Plane:** This option allows you to specify the reference plane that will be used as the base plane. The **Object (XY)** option is the default and uses the reference plane defined by the object's X and Y axes. The **Object (YZ)** and **Object (ZX)** use the plane defined by the object's Y and Z axes or the object's Z and X axes respectively. The **Active Plane** option uses the active reference plane from the active window. The **XY**, **YZ**, and **ZX** options use the corresponding Cartesian reference planes. These options are useful when it is desirable to work in a projection view and the projection reference plane is not the appropriate reference plane for the operation.

**Initial Limits:** This option contains two text fields for determining the initial lower and upper limits of the deformation box. The **Lower** field is the percentage of the height of the object at which the lower limit is initially placed. The **Upper** field is the same for the upper limit of the deformation box. The defaults for these are 0% and 100%, respectively, which contains the entire object. A common use of these fields would be to deform just the top half of an object. To do so, you would set the lower limit to 50% and the upper to 100%.

**Plain Object:** When this option is selected a plain object is constructed instead of a controlled deformation object. This option is off by default.

**Edit:** When this option is selected, the **Deformation Edit** dialog is invoked after the graphic creation of the deformation is completed. This dialog is used to add, remove, and re-order the deformation list and edit the parameters of each deformation in the list. This is described in the next section.
Editing a deformation object.

When the Deform tool is executed with the Edit option on in the Deform Options dialog, the Deformation Edit dialog, shown in Figure 4.11.4.19, is invoked. This dialog can also be accessed from the Edit Parameters Tool and the Edit button of the Query Object dialog. This dialog offers the opportunity to preview the deformed object, as well as to experiment with different variations by changing the deformation parameters and adding and removing deformations to the deformations list. This dialog contains a standard preview window on the left, the options found in the Deform Options dialog, and the following additional options:

![Deformation Edit dialog with Parameters and Axis And Box tabs open](image)

Figure 4.11.4.19: The Deformation Edit dialog with (a) the Parameters and (b) Axis And Box tabs open.
**Deformation List:** This area to the right of the preview area lists the deformations that are applied to an object. The list has two columns. The first column identifies the type of deformation. One item in the list is highlighted and referred to as the *active deformation*. Clicking on the name of the deformation in the list makes it the active deformation. Clicking and holding the mouse button down allows you to drag the active deformation to a different position in the list. The second column indicates whether the deformations is enabled. When it is, a check mark is displayed. Clicking in this column toggles the enabled state of the deformation. When a deformation is enabled, it is applied to the object; it is ignored otherwise. This makes it possible to isolate a particular deformation when the operation is applied.

**Add Deformation:** This pop up menu (shown in Figure 4.11.4.20) contains deformations that can be added to the Deformation List. When such a deformation is selected, it is added after the active deformation and becomes the new active deformation.

**Remove:** This button removes the active deformation from the deformation list.

Other than these options, the *Deformation Edit* dialog contains two tabs: **Parameters** and **Axis And Box**.

**Parameters:** This tab displays the parameters of the active deformation. Its contents varies depending on the type of the active deformation.

The parameters displayed are the numeric equivalents to the graphically applied deformations. The numerical options represent more control over deformations that may be applied graphically. Most of the parameters can be edited relative to the deformation box or in their real world dimensional coordinates. These parameters have a slider with a percentage text field to the right, followed by a numeric text field. The slider and text fields are all linked to the same parameter. This allows for percentage sliders to be used when the actual dimensions are less important and experimentation is desired. The dimensional text field can be used when accuracy is desired.

**Box Parameters:** This section contains parameters that affect the deformation box.

**Upper Limit, Lower Limit:** These parameters constrain the area that is affected by the deformation. This is equivalent to setting the upper and lower end of the box graphically.

**Anchor:** This parameter controls where the anchor point lies along the deformation axis. If its value is less or equal to the lower limit or greater or equal to the upper limit, then the anchor has no effect. If the deformation is created with the **Through Center** option on, then the anchor will have a value of 50%.

**Axis Offset:** These parameters control the offset of the deformation axis from the central axis of the deformation box.
Shear and Taper: These deformations are controlled by a scale factor in X and Y directions which in the graphic interface is applied at the top or bottom of the deformation box or at the middle of the box if the through center option is selected. The numeric control allows for specification of values at all three locations. In the Deformation Edit dialog the Through Center position is not restricted to the middle of the box, hence this option is referred to as the Anchor. Figure 4.11.4.22 shows the effect to the Anchor parameter.

![Figure 4.11.4.21: Shear and Taper parameters.](image)

Figure 4.11.4.21: Shear and Taper parameters.

Twist: This deformation is controlled by a twist angle, which, in the graphic interface, is applied at the top or bottom of the deformation box or at the middle, if the Through Center option is selected. The numeric control allows for specification of values at all three locations. As with the Shear and Taper, the Through Center position is not restricted to the middle of the box. Figure 4.11.4.23 shows the Parameters for Twist and Figure 4.11.4.24 shows examples for different values of selected parameters.

![Figure 4.11.4.23: Twist parameters.](image)

Figure 4.11.4.23: Twist parameters.

**Figure 4.11.4.22:** Examples:
- (a) Upper Angle = 0°, Lower Angle = 0°, Anchor Scale = 50%;
- (b) Upper Angle = 90°, Lower Angle = -90°, Anchor Scale = 50%;
- (c) Upper Angle = 90° and Axis Offset X = 50%.

**Figure 4.11.4.24:** Twist examples:
- (a) Upper Angle = 0°, Lower Angle = 0°, Anchor Angle = 45° and Anchor = 50%;
- (b) Upper Angle = 90°, Lower Angle = -90°, Anchor Angle = 0° and Anchor = 25%;
- (c) Upper Angle = 90° and Axis Offset X = 50%.
**Bulge** and **Radial Shear**: These deformations are controlled by a scale factor in the X and Y directions, which, in the graphic interface, is applied at the middle of the deformation box or at the top and bottom, when the **Through Center** option is on. Like the **Shear**, **Taper**, and **Twist**, the anchor replaces the **Through Center** option in the dialog. Figure 4.11.4.25 shows the **Parameters** for **Bulge** and **Radial Shear** and Figure 4.11.4.26 shows examples for different values of selected parameters.

*Figure 4.11.4.25: Bulge and Radial Shear* parameters.

**Radial Bend**: This deformation is controlled by an angle parameter, which, in the graphic interface, is applied at the top or bottom of the deformation box or at the middle, if the **Through Center** option is on. The **Spin** parameter controls in which direction the bend is applied. In the graphic interface this is determined by which edge of the box is selected. For example, if an edge that is parallel to the X axis is selected, the spin value will be 90° or -90°, depending on the direction of the bend (clockwise or counter-clockwise relative to the X axis). Figure 4.11.4.27 shows the **Parameters** for **Radial Bend** and Figure 4.11.4.28 shows examples for different values of selected parameters.

*Figure 4.11.4.27: Radial Bend* parameters.

*Figure 4.11.4.26: Bulge examples with Anchor Scale X = 50% and Anchor = (a) 50%, (b) 0%, and (c) 100%.

*Figure 4.11.4.28: Radial Bend examples with Angle = 90° and Anchor = (a) 0%, (b) 50%, and (c) 100%.**
Bezier Bend: This deformation is controlled by four points (upper, upper anchor, lower anchor, and lower) along the deformation axis, which are re-positioned to deform the object. The graphic interface offers a variety of ways to change the shape, which implicitly define the new control point locations. The Parameters section provides numeric control for the new control point locations and the Box Parameters section provides numeric control for the original positions along the axis. They are both shown in Figure 4.11.4.29.

Disturb Points: This operation is controlled by the Min and Max disturbance distances in the X, Y, and Z directions, which are described in section 4.11.3 and shown in Figure 4.11.3.4. The Seed parameter controls the nature of the randomness of the disturbance. Changing this value yields a different pattern in the disturbance.

Disturb Point Normals: This operation is controlled by the Min and Max disturbance distances along the normal, which are described in section 4.11.3 and shown in Figure 4.11.3.5. The Seed parameter is as with the Disturb Points option.
**Linear Wave**: This operation is controlled by the **Height**, **Width**, and **Offset** parameters which are described in section 4.11.3 and shown in Figures 4.11.3.6 and 4.11.3.7. The **Upper Limit** and **Lower Limit** parameters control the range of the wave. The wave is applied in the area of the object between the upper and lower limits. The upper limit controls how far from the reference point the wave starts. The lower limit controls how far from the reference point the wave stops. The area of the object beyond the upper and lower limits remains unchanged. The default values are large positive and negative numbers respectively that are designed to span to the extents of most objects. Figure 4.11.4.33 shows an example of using the limits of a linear wave. The **Axis Offset** parameters control the reference point of the wave. These values are the offset distance from the reference point to the object's origin relative to the base plane.

![Figure 4.11.4.32: Linear Wave parameters.](image)

![Figure 4.11.4.33: Linear Wave applied to a 32'-0" x 32'-0" object with Upper Limit = 20'-0" and Lower Limit = -10'-0" and Exterior Limit = 30'.](image)

**Circular Wave**: This operation is controlled by the **Height**, **Width**, and **Offset** parameters that are described in section 4.11.3 and shown in Figures 4.11.3.6 and 4.11.3.7. The **Interior Limit** and **Exterior Limit** parameters control the range of the wave. The interior limit controls how far from the reference point, the wave starts. The area of the object between the reference point and the interior limit remains unchanged. The default is 0. The exterior limit controls how far from the reference point, the wave stops. The area of the object beyond the reference point remains unchanged. The default is a large number designed to span to the extents of most objects. Figure 4.11.4.35 shows an example of using the limits of a circular wave. The **Axis Offset** parameters are as with the **Linear Wave**.

![Figure 4.11.4.34: Circular Wave parameters.](image)

![Figure 4.11.4.35: Circular Wave applied to a 32'-0" x 32'-0" mesh object with Interior Limit = 8'-0" and Exterior Limit = 30'-0".](image)
**Spread**: This operation is controlled by the **Offset** and **Distance** parameters that are described in section 4.11.3 and shown in Figure 4.11.3.8. This operation does not have the box parameters (Figure 4.11.4.36).

**Axis And Box**: This tab contains items that define the axis and box for the deformation. The contents of this area are the same for all deformation types. The values displayed correspond to the active deformation.

**Axis**: This group defines the axis of the plane for the deformation. These values are initially based on the settings in the **Base Reference Plane** area of the **Deform Options** dialog. The fields are consistent with the same options in the **Object Transformation** section of the **Query Object** dialog (see section 4.22.1).

**Box**: This group contains the parameters for the deformation box. When **Use Object Extents** is on, the extents of the object, relative to the deformation axis, is used as the deformation box. This is the default unless the deformation was initially created in conjunction with another object, the **All Objects At Once** option is enabled, and the **Join** option disabled. If the deformation is the first deformation in the list, then the deformation box is the extents of the original object. For all other deformations in the list, the deformation box is the extents of the object, after the previous deformations in the list are applied. This prevents changes to the parameters of previous deformations from having an undesirable effect on the deformation. When **Custom** is selected, the deformation box is defined by the values in the **Minimum** and **Maximum** groups.

**Update**: Clicking on this button applies the new parameters and refreshes the preview window. This button is dimmed when the parameters of the preview are up to date.

**Automatic**: When this option is on, the preview is automatically updated every time a parameter is changed. When this option is on, the **Update** button is dimmed.

The remaining options in the dialogs are the same as for the **Deform Options** dialog discussed earlier in this section.
How the deformation tools can be used

The ways in which the deformation tools can be used to reshape the form of an object are unlimited. A few possibilities are illustrated in this section.

Note that only areas of a model which are meshed deform smoothly. Therefore, appropriate meshes should be generated before deformations are applied. In the examples of Figure 4.11.4.38, a meshed cone was generated as a c-mesh, using a large and a very small circle as control lines. The fork in 4.11.4.39 was first created as an extruded object, and then a mesh in only one direction was generated.

**Figure 4.11.4.38:** (a) Original meshed cone. (b) Bulge deformation applied to generate onion dome. (c) Taper operation applied to generate a bullet shape. (d), (e) Bezier Bend applied to form horns.

**Figure 4.11.4.39:** A fork generated from a flat form by a sequence of radial and Bezier bending deformations.
While in the previous two examples it sufficed to generate equally
distributed meshes, by using standard tools of form.Z, this is
not always the case. Some times the deformations work best
if edges are inserted at strategic positions, as illustrated by
the next two examples. To deform the object shown in Figure
4.11.4.40, the object first had to be properly generated by plac-
ing the edges about which the deformations would be applied
at proper positions. The construction process, illustrated in
Figure 4.11.4.40, was as follows:

(a) A rectangular extrusion is created first.
(b) The vertical segments of the object are rounded.
(c) The top and bottom faces of the object are rounded.
(d) Ten lines are drawn on the XY plane. They are strategi-
cally positioned where they will next be “inserted”.
(e) Using the Trim/Split tool, Split With Line and Stitch
selected in the Trim/Split Options dialog, click on the object
and then on the line. The line is imprinted in the object. Repeat
the operation with all the other lines.

After the object was selectively meshed, a number of deforma-
tions were applied, as shown in Figure 4.11.4.41. Similarly, the
telephone handset in Figure 4.11.4.42 first had to be strategically
constructed and then deformed to complete its shape.
### 4.11.5 Bending objects along curves

#### Bend along Curve

The Bend along Curve tool constructs a new controlled object by bending an object along a specified curve. The object to be bent is called the **source object** and the curve along which it is bent is called the **path**. The bending operation is first applied to an axial line of the source object, which is bent according to the path line. Once the axial line is transformed to the shape of the path the source object is modified relative to its axial line.

The axial line that is mapped to a path is referred to as the **bend axis**. A line perpendicular to the bend axis is used to orient the source object relative to the path curve and is called the **alignment axis**. The bend axis and the alignment axis together define the bend reference plane. The alignment axis is aligned to the normal of the curve at the starting point, much like the sweep operation.

Either the prepick or postpick selection method can be used when executing the Bend along Curve tool and there are five variations relative to how the axes of the bending operation are determined. Two of these variations require that the axes are also explicitly selected, in addition to the selection of the source and path objects. How the selections occur for each variation of the operation is discussed in detail below. The variation of the operation and other options are selected from the **Bend Along Curve Options** dialog, shown in Figure 4.11.5.1, that is invoked directly from the tool.

#### Model Type

This pair of options determines the type of the model to be generated, as follows:

**Facetted**: When this option is on, a facetted object will be generated regardless of the object types selected as operands. If either or both of the source and path objects are smooth, they will be converted to facetted before applying bending.

**Smooth**: When this option is on, a smooth object will be generated. If the source object is facetted, it will be converted to smooth before bending it. If the path curve consists of more than one segment or it is a facetted vector line, the result might contain self-intersecting faces.
Reference Axes From: This group of options contains five variations relative to how the bend and alignment axes are determined.

Bounding Box: When this option is on (default), the bend and alignment axes are determined automatically from the bounding box of the source object. The longest dimension of the bounding box determines the bend axis and the next longest the alignment axis. The origin of these axes is at the centroid of the basis of the bounding box. This variation requires the selection of a source object and a path curve, which can be done using either the postpick or the prepick method.

When using the postpick method, with the Bend along Curve tool active and topological level set to Object, you click on the source and then on the path. When using the prepick method, any number of source objects can be selected using the Pick tool, then the Bend along Curve tool is activated and the path is selected. All preselected objects are bent along the path. This variation of the bending operation is shown in Figure 4.11.5.2.

Figure 4.11.5.2: Bending along an arc with the From Bounding Box option on:
(a) a cuboid and (b) an object of revolution.
**Object Axes**: When this option is on, the bend and alignment axes are determined from the local axes of the source object. The Z axis becomes the bend axis and the X axis becomes the alignment axis. Both the postpick and prepick methods can be used and work as for the From Bounding Box variation.

**Active Plane**: When this option is on, the X and Y axes of the active reference plane are used as the bend and alignment axes, respectively. Picking works as for the previous two options. Note that the position of the source object relative to the reference plane axes is significant and will be preserved when the object is bent. An example is shown in Figure 4.11.5.3.

![Figure 4.11.5.3: Bending along an arc with the From Active Plane option on.](image1)

Four cuboids are placed at different distances from the X axis of the reference plane. These distances are preserved when the cuboids are bent.

**Selected Axes**: When this option is on, the bend and alignment axes are explicitly selected, after the source and path are selected. The axes may be Cartesian world axes, axes of a reference plane, or any two segments drawn by the user. The latter case is illustrated by the example in Figure 4.11.5.4. Note the order in which the entities are picked, as shown in the first image.

![Figure 4.11.5.4: Bending along an arc with the From Selected Axes option on.](image2)

The entities are picked by clicking on 1 (source), 2 (path), 3 (bend axis), and 4 (alignment axis).
As illustrated in the example, when using postpicking, with the Bend along Curve tool active you first click on the source, then the path, the bend axis, and last on the alignment axis. When using the prepick selection method, you first use the Pick tool to preselect any number of source objects. Then, with the Bend along Curve tool active, you click on the path, the bend axis, and the alignment axis. All preselected objects are bent along the curve.

**Drawn Axes:** When this option is on, the bend and alignment axes again need to be explicitly specified, which is done by drawing them immediately after the source and the path objects are picked. The process is illustrated in Figure 4.11.5.5.

![Figure 4.11.5.5: Bending along an arc with the From Drawn Axes option on. The entities are picked by clicking on 1 (source), 2 (path), 3, 4, and 5, to draw the bend and alignment axes. Note that point 3 is the apex of the pyramid, which is drawn using Point Snap. Also note how a different alignment axis positions the source object differently along the path curve.](image)

When using the post pick method, with the Bend along Curve tool active, you click on the source and the path, and you then proceed to click three more times, explicitly drawing the bend and alignment axes. The first and second click points determine the bend axis, while the second and third points determine the alignment axis. The three points must not be colinear. All or some of the three points may be points of the source object, which can be facilitated by turning on **Point Snap**, as done in the example of Figure 4.11.5.5. With the prepick method, any number of source objects are preselected with the Pick tool. You then activate the Bend along Curve tool and click on the path and three more points to draw the bend and alignment axes.

Note that the positions of the bend and alignment axes have a major effect on how the source object is placed and aligned along the path curve, when it is bent. Different axes produce distinctly different forms, as was illustrated in Figure 4.11.5.5 and is further illustrated in Figure 4.11.5.6.
Rotation Of Alignment Axis: This field may be used to specify an angle by which the source will be rotated as it is bent along the path curve. A rotation angle of 0 degrees, which is the default, has no effect.

Scaling: This group of options determines the dimensions of the bent object relative to its original size.

Natural: When this option is selected, the source object is either elongated or squeezed along the dimension of bend axis, depending on whether it is longer or shorter than the path curve. An example is shown in Figure 4.11.5.7(b).

Preserve Proportions: When this option is on, the dimension of the source object along its bend axis is again adjusted to fit the length of the path curve and, at the same time, its width is also scaled by the same factor. That is, the whole object is uniformly scaled by a factor that is determined by the fraction of its length over the length of the path curve. An example of this option is shown in Figure 4.11.5.7(c).

Preserve Original Size: When this option is on, the source object is not scaled, but retains its original dimensions.

All the scaling options are affected by the following parameters, which may be used to specify a starting and/or ending point on the path, where the source is placed.

Start: The percentage value entered in this field specifies the point along the path curve where the bent object will start.

End: The percentage value in this field specifies the point along the path where the bent object will end. This option does not apply to Preserve Size and is dimmed when that option is on. The Start and End parameters are illustrated in Figure 4.11.5.8.
**Figure 4.11.5.7:** (a) A source and two path curves. Bending using *Scaling* options: (b) *Natural*, (c) *Preserve Proportions*, and (d) *Preserve Size*.

In all cases *Start* = 0% and *End* = 100%

**Figure 4.11.5.8:** (a) Source and path objects. Bending with options: (b) *Natural*, *Start* =50%, *End* =100%. (c) *Preserve Proportions*, *Start* =50%, *End* =100%.

(d) Repeating Bend 5 times with *Preserve Size* on and *Start* =0%, 25%, 50%, 75%, and 100%.
**Join Adjacent Coplanar Faces**: When this option is on, all the neighboring faces that lie on the same plane are joined. This option is off by default and only applies to facetted objects. An example is shown in Figure 4.11.5.9.

![Figure 4.11.5.9: (a) A source and a path curve. Bending with the Join Adjacent Coplanar Faces option (b) off and (c) on.](image)

**Triangulate Non Planar Faces**: When this option is on, all non planar faces are triangulated. This option is off by default and only applies to facetted objects.

**Split Faces At Path Points**: When this option is selected, the faces of the source object are spilt along a new edge that corresponds to the points of the path object. This is option is necessary when bending an object with a few large faces along a complex path. If the source object is of sufficient density of faces, this option may be disabled and the source object will retain the same number of faces after the operation. It is often desirable to use this option when animating deformations as the original objects topology is preserved. This option is on by default.

**Plain Object, Edit, Adjust To New Parameters**: These options work as for all other objects.
Previewing and editing Bend along Curve objects

When the Bend Along Curve tool is executed with the Edit option on in the Bend Along Curve Options dialog, the Bend Along Curve Edit dialog, shown in Figure 4.11.5.10, is invoked. This dialog offers the opportunity to preview the bent object, as well as to experiment with different variations. This dialog contains a standard preview window on the left, the options found in the Bend Along Curve Options dialog, and the following additional options:

**Update**: Clicking on this button applies the new parameters and refreshes the preview window. This button is dimmed when the parameters of the preview are up to date.

**Automatic**: When this option is on, the preview is automatically updated every time a parameter is changed. When this option is on, the Update button is dimmed.

Objects bent with the Bend along Curve tool can be edited using the Edit Controls tool. When in edit mode, their rotation relative to the alignment axis and the shape of the path curve can be changed.

*Figure 4.11.5.10*: The Bend Along Curve Edit preview dialog.
Examples

**Figure 4.11.5.11:** (a) A source (pencil model) and a non-planar path curve. (b) Bending the pencil along the non planar path, using default options.

**Figure 4.11.5.12:** (a) The sources are a solid beam and a surface. The path is non planar. (b) Bending the beam. (c) Bending the surface object.

**Figure 4.11.5.13:** (a) A solid source with cuts, holes, and a path. The bent object is shown in (b) wire frame and (c) hidden line.
4.11.6 Morphing

The Morph tool transforms the shape of one object, called the source, to the shape of another object, called the destination. The new object has a shape that is partially similar to the former and partially similar to the latter. How close to the source and how close to the destination the new object is depends on a percentage specified by the user either graphically or numerically. The numeric parameter as well as additional options that determine how this tool is executed are set in the Morph Options dialog, shown in Figure 4.11.6.1, which can be invoked directly from the tool.

This tool may be executed in either postpick or prepick mode. When using postpick, with the tool active, you click on the destination object and then on the source. Depending on the mode of execution selected in its dialog, the operation will be executed either dynamically, directly, or through a preview dialog. When using prepick, with the Pick tool you select any number of pairs of objects, where for each pair the first pick is the destination and the second is the source. Then you activate the Morph tool and you click anywhere in the graphics window to execute the operation.

The morphed object is a controlled (parametric) object and can be edited and changed after its initial generation. To edit a morphed object, with the Morph tool active click on it. Or you can edit multiple objects using prepicking. That is, with the Pick tool select as many morphed objects as desired and then with the Morph tool click anywhere in the project window.

The Morph Options dialog contains three tabs, two of which (Display Resolution and Status Of Objects) are as in other dialogs. The content of the Options tab is as follows:

Mode: The options in this group determine the manner in which the tool will be executed.

  Dynamic: When this option is on, as soon as the source object is picked (second click), the morphed object is rubber banded and follows the motion of the mouse. When using postpick, the motion is relative to the two objects and the closer the mouse is to the destination, the more the morphed object resembles the destination object. Likewise for the source object. A third click at the desired position freezes the morphing and completes the operation by generating a new object.
Tracking Speed: This slide bar is only available when Dynamic is on and determines how fast the motion of the mouse will affect the dynamic morphing of the object. Closer to the left end the effect is slower and closer to the right end it is faster.

Preset: When this option is on, the morphed object is generated as soon as the two operands are picked. The shape of the morphed object is affected by a percentage value entered in the Morph Options dialog, as follows:

Morph Percent: This parameter, which is only available when Preset is on, is both a slide bar and a numeric field. It represents how much of the destination object is in the new object. The new object is generated between the source and the destination, at a distance from the source determined by this value. The slide bar ranges from 0% (left) to 100% (right) and its position is automatically reflected in the numeric field. Values can also be typed in the numeric field directly and these can go beyond the range of 0% and 100%. For example 110% generates an object that is 110% the destination object. While such an object is difficult to visualize it can be constructed by allowing the morphing algorithm to continue beyond its target, which may produce some rather intriguing forms.

Note that the ability to morph beyond 0% and 100% is also available in the Dynamic mode, where the mouse can be moved beyond the distances that correspond to the source and destination objects.

Edit: This option is only available when Preset is on. When on, a preview dialog, called Morph Edit, is invoked as soon as the operands are picked. This is shown in Figure 4.11.6.2 and works as all the preview dialogs. It allows you to change the available parameters and preview the results before deciding which one is desirable.

Figure 4.11.6.2: The Morph Edit dialog.
**Placement:** The options in this group determine how objects will be morphed.

**At Source:** When this option is on, the morphed object is generated in the position of the source. If **Dynamic** mode is used, the object changes dynamically while it remains in the same position. If **Preset** is used, the morphed object is generated in one step on top of the source object. This is illustrated in Figure 4.11.6.3.

**Between Source And Destination:** When this option is on, the morphed object is generated in positions between the source and the destination, assuming these are not on top of each other. If **Dynamic** mode is used, the morphed object moves between the source and the destination while its shape changes dynamically. If **Preset** is used, the morphed object is generated in one step in a position between the two operands, which is determined by the same percentage used for the morphing. This is illustrated in Figure 4.11.6.4. If the source and destination are on top of each other and there is no distance between them, the new object is generated in the same position.

**Ignore Orientation:** The objects to be morphed may have the same or different orientations. For example, two extrusions may both be sitting on the XY plane with their Z local axis perpendicular to the XY plane. In such a case they have the same orientation. Alternatively, an object’s Z axis may be perpendicular to the XY plane and the other’s perpendicular to the YZ plane or in some arbitrary direction in 3D space. In such a case they have different orientations.
When the two objects have the same orientation, this option has no effect. When they have
different orientations, there are two ways to morph them. With this option on, the two objects
are morphed in the position they are and their orientations are ignored. With this option off, the
morphing is relative to the orientations and the source object appears to rotate as it morphs
towards the shape of the destination object. The examples in Figures 4.11.6.3 and 4.11.6.4
had the same orientations. An example with different orientations is shown in Figure 4.11.6.5.

**Tips about using morphing**

Two objects may have the same topology and be topologically homomorphic, or they may have
different topologies. The safest way to morph is with objects that have the same topology.
Note, however, that having the same topology does not imply that the two objects look the
same. On the contrary, they may look quite different, as some of the following examples will
illustrate.

When the application of morphing begins, the first thing the program does is to check whether
the two operands have the same or different topologies. If they have the same topology, it
proceeds and matches pairs of points, one from each object, based on the topologies of the
objects. If the operands have different topologies, the program applies geometric heuristics
and matches the points of one object to points of the other object. Typically, one object has
more points than the other or does not have enough points in an area where they are needed
to match points of the other object. In such cases new points are introduced to one or both of
the objects. The result is a hybrid object that has a topology common with the two objects and
can take the form of each of the original objects. This hybrid object is used for the morphing.

Figure 4.11.6.6 illustrates the morphing of
two objects that look quite different but have
the same topology. This is achieved by
manipulating the points of the two objects.
The source looks like a cone and the
destination looks like a cube. They were both
derived from hexagonal extrusions, whose
topology they have. The cone was derived
by moving the points of the top hexagon to its
center. The cube was derived by moving two
of the side edges onto other edges and then
scaling the whole object to the proportions of
a cube. This example illustrates one of the
most basic techniques for morphing between
different shapes.

![Figure 4.11.6.6](image)

**Figure 4.11.6.6:**
Morphing objects with same topologies, but different looks.
(a) Original objects.
Morphed objects at (b) 25%, (c) 50%, and (d) 75%.
Another example of morphing objects with similar topologies but different looks is shown in Figure 4.11.6.7, where a pyramid morphs into a star. The pyramid was actually derived from a star by collapsing some of its points onto other points.

There are more techniques for creating objects that look different while they have the same topology. One is illustrated in Figure 4.11.6.8, where two objects of revolution are generated from two different source shapes that have the same number of points and are revolved the same number of times. This results in identical topologies, while the objects look different.

In contrast to these examples, Figure 4.11.6.9 illustrates the morphing of two objects, a hexagonal extrusion and a icosahedron, that look different and have different topologies. The different topologies are matched and a hybrid object is used for morphing between the two objects. It should be noted that there are varying degrees by which two topologies may be different and with non-matching topologies the morphing process is some times less successful than others. For example, it is much harder to match the topology of an object that contains holes with that of an object that does not. For better results it is recommended that only objects with comparable hole structures are morphed.
Another example where the source looks very different from the destination is shown in Figure 4.11.6.10. Eight cubes morph into a structure, which consists of eight cuboids. Here the individual components are very similar forms (and topologies), but the overall arrangements appear very different.

There are two methods by which multiple objects can be morphed into a similar number of objects. The source objects and the destination objects can be joined into single objects, which are then morphed in the usual way. The objects will be matched in the order they were picked when they were joined.

Another method for morphing multiple objects is to use the prepick selection method. With the Pick tool you click on a source and then on its destination and you repeat this for as many pairs of objects as desired. When you next start the operation by clicking with the Morph tool somewhere in the window, all the pairs of objects are morphed simultaneously.

Figure 4.11.6.10: Morphing multiple objects in one step. Shown in wire frame and hidden line.

Morphing, especially when it is applied to objects with non-matching topologies, has the potential to generate forms that are hard to perceive and that cannot be generated in any other way. This makes it an excellent tool for exploratory design. Whether the derived forms are meaningful or not will depend on the task at hand. Some rather “conservative” examples of morphing non-matching topologies are shown in Figures 4.11.6.11 and 4.11.6.12.
Figure 4.11.6.11: Morphing the layer A into O (different topologies). Shown in wire frame and hidden line.

Figure 4.11.6.12: Morphing a hexagonal extrusion to a torus. Source and destination have (1) same and (2) different orientations. Shown in wire frame and hidden line.
Morphing nurbs surfaces

It is also possible to morph nurbs surfaces and objects. To do so it is required that both the source and the destination are nurbs. In other words, types of objects cannot be mixed. The morphing procedure for nurbs objects is slightly different than that of other objects. It takes advantage of the regularity of the underlying grid of control points. In a first step, the two nurbs objects are changed to have the same number of control points and the same degree. The surface with the lower degree is raised to match the degree of the other surface. Then the number of control points of the surface with less points in length or depth are increased to match the number of control points of the other surface. After this step, both nurbs objects have the same degree and same number of control points.

In a second step, the destination object is temporarily transformed so that its object axes are coincident with those of the source nurbs object, as shown in Figure 4.11.6.13(b). This temporary transformation of the destination object is not shown graphically on the screen, but described here to help in the understanding of the morph process. Note that the object axes of a nurbs object are placed by default at the first control point in length and depth, with the X axis pointing in the length direction and the Y axis pointing in the depth direction. In the example shown, the object axes of the destination object were first edited to be oriented horizontally. Once the destination and source objects are aligned in this way, each control point of the source nurbs object is moved in a straight line towards the corresponding control point of the destination nurbs object by the morph percentage. For example, a 50% morph percentage moves a control point half way between its original position and the position of the control point of the destination object (Figure 4.11.6.13(c)). If the Placement option is set to Between Source And Destination, the source is rotated and translated towards the original position of the destination object by the morph percentage, again using the alignment of the object axes (Figure 4.11.6.13(d)). If the placement option is set to At Source, the source object is only rotated, but not moved (Figure 4.11.6.13(e)).

Figure 4.11.6.13:
Morphing nurbs surfaces.
(a) Source and destination.
(b) Objects are made coincident.
(c) 50% morphing.
(d) Between Source And Destination on.
(e) At Source on.
4.11.7 Image based displacements

The points of meshes can also be moved relative to the gray color intensity of an image. This is called image based displacement and is executed with the following tool.

Displace

This tool can be used to "imprint" an image on the surface of an object. The image is a gray scale image whose shades are used to displace points of the surface geometrically, which may already exist or may be created by the displacement process. This resembles the bump effects of rendering, where an image is used to make a surface appear sculpted. For example, the mortar joints of a brick texture can be made to look three dimensional with a brick bump map. However, there is a major difference between the two techniques. Bump mapping simply adds a depth illusion and does not affect the geometry of a surface, which image-based displacement does.

Using the postpick method, a displacement is applied by clicking on the object to be displaced with the Displacement tool. Note that the postpick method always picks at the Object level, regardless of the topological level currently selected. When using the prepick method, any number of entities at either the Object or Face topological levels may be selected using the Pick tool. Then, with the Displacement tool active, click anywhere in the graphic window. This invokes the Displacement Edit dialog, shown in Figure 4.11.7.1, where the displacement position and parameters are set.

Note, that this dialog is quite similar to the Texture Map Controls dialog (see section 6.3 of the form•Z RenderZone Plus User's Manual). In fact much of the information presented in this section is also included in section 6.3. We repeat it here to make this section complete, mostly for those users who do not have RenderZone Plus.

Objects produced by image-based displacements, to be referred to as displaced objects, are controlled objects. That is, they carry the displacement parameters in their internal representations, and can be subsequently edited and changed by applying the Displacement tool again.
Previewing and positioning displacements

A bit mapped image that is used to displace the geometry of a surface will be referred to as a displacement map. Displacement maps are typically monochromatic and have directions. For example, a displacement map representing a human face has a distinct horizontal and vertical orientation. Positioning such a map upside down makes little sense. As the position of the displacement map is manipulated, it can be viewed in the preview window, located in the upper left area of the Displacement Map Edit dialog (Figure 4.11.7.1). This preview window works in a fashion similar to that of other preview windows in form•Z.

On invoking the Displacement Map Edit dialog, the preview window displays the original object to be displaced and the current position of the displacement coordinate system. Depending on the mapping type currently selected (see below), a flat plane, a cylinder, or a sphere representing the mapping surface is also displayed. These shapes bound the object and have their center at the origin of the coordinate system. As shown in Figure 4.11.7.2, these bounding shapes can be significantly different in size, depending on where the displacement origin is located. They are the smallest when the Object Center option is on, which causes the origin of the bounding shape to be placed at the center of the object.

![Figure 4.11.7.2: The displacement bounding shapes as displayed in the preview window: Mapping origin at (1) center of object and (2) origin of world space. Mapping type (a) flat, (b) cylindrical, and (c) spherical.](image)

Modeling • Deformations, displacements, and morphing
The images in the preview window are color coded:

- The bounding shapes and two of the axes of the displacement coordinate system are shown in the same color used for the reference plane grid lines in the main form•Z windows, which is light green by default.

- The third axis is shown in the color used for the world axes in the main form•Z windows which is red by default. This is the axis about which rotations can be executed, and corresponds to the axis currently selected in the **Axis/Plane** pop up menu (see below).

- The faces used for displacement (see below) are drawn in black. Faces not used for displacement are drawn in the project’s ghost color, which is gray by default.

The check boxes and radio buttons under the preview window determine what is displayed in it, and how the display is refreshed:

**Show Object**: When this option is on (default) the object is drawn. If off, the object is not drawn and its faces cannot be selected.

**Original**: When this option is selected, the original, non-displaced object is shown in the graphic window.

**Displaced**: When this option is selected, the displaced object is shown. If any of the displacement parameters were changed since this option was last checked, the updated version of the object is generated first.

**Update**: When this button is clicked, the updated version of the displacement object is generated and displayed in the graphic window. This button is only active if the **Displaced Object** option is selected and a parameter was changed since the object was last displayed.

**Snap**: When this option is on and you move the mouse after you have initiated a move origin or a rotate axes operation (see below), the mouse snaps to points, midpoints, and segments of the object. You can temporarily cancel the snapping by pressing the **shift** key. Likewise, if **Snap** is off, you can temporarily activate it by pressing the **shift** key.

**Show Mapping Type**: When this option is on (default), the displacement coordinate system and the bounding shape that represents the mapping type are drawn.

**Draw Tiles**: This option produces a wire frame image that includes a display of how the displacement tiles will be mapped onto the faces of the object. Only front facing faces are tiled.
The view in the preview window can be manipulated using tools from the preview tool palette located under the preview window. All these tools are the same as those in other preview windows. In the Displacement Map Edit preview window the Arrow tool (▼) is used to move and rotate the displacement coordinate system, as discussed in more detail below. It is also used to select or deselect faces for displacement.

The Displacement Map Edit dialog offers both numeric and graphic methods for positioning textures onto objects. For both methods, two basic operations are available: moving the origin of the displacement coordinate system, and rotating the displacement coordinate system about one of its axes.

**Numeric input**

The text fields and buttons for setting the position of the displacement coordinate system are located in the upper right corner of the Displacement Map Edit dialog.

**Origin**: The values entered in the X, Y, and Z fields represent the position in the world coordinate system to which the origin of the displacement coordinate system will be moved.

**Object Center**: Clicking on this button moves the origin of the displacement to the center of the object. The coordinates of the center are also displayed in the X, Y, and Z fields. Pressing the option key on the Macintosh or ctrl+shift on Windows changes the label on this button to Center Of Faces. Clicking on this button while these keys are pressed moves the origin of the displacement to the average coordinates of all the faces currently selected for displacement.

**Rotation**: The values entered in the X, Y, and Z fields are in degrees, and represent the angle by which the displacement coordinate system will be rotated relative to each of its axes. These rotations are always executed in the Z, Y, X order.

**Reset**: Clicking on this button cancels all the rotations, and resets the values in the X, Y, and Z fields to 0°. Note that this button does not affect any other operations that may have been applied to the displacement, either numerically or graphically.
Graphic input

When the position of the displacement coordinate system is manipulated through numeric input, the image in the preview window of the Displacement Map Edit dialog is adjusted to reflect those changes. The position of the displacement can also be manipulated graphically, by moving and rotating its coordinate system directly in the preview window.

When the Arrow tool (click here) from the preview tool palette is selected and the cursor is moved inside the preview window, its icon changes depending on which part of the displacement coordinate system or the object it is closest to.

- The cursor changes to this icon when it is on or close to the origin of the displacement coordinate system. It signifies that the origin can be moved. You move it by pressing the mouse button, dragging the cursor to the desired position, and releasing the button.

- The cursor changes to this icon when it is on or close to an axis that can be rotated. Which axes can be rotated depends on the current axis of rotation, which is selected from the Axis/Plane pop up menu located under the preview window. The axes that are not the current axis of rotation can be rotated. For example, if Z/XY is selected, then X and Y can be rotated about the Z axis. In the preview window, the axis of rotation is displayed in red, and the other axes in light green. An axis is rotated about the axis of rotation by pressing the mouse button, dragging it to a new position, and releasing it.

- The cursor changes to this icon when you press the button to execute either a move origin or a rotate axis operation. It returns to its previous icon as soon as the button is released to complete the operation.

- As the mouse cursor is moved, either to move the origin or to rotate an axis, if Snap is on, it will snap to the points, segments, or segment mid-points of the object. If Snap is off, it will move freely, relative to the plane selected in the Axis/Plane pop up menu. For example, if Z/XY is selected, the move will be relative (parallel) to the XY plane. Both states of the Snap can be temporarily reversed by pressing the shift key. When the mouse snaps, the cursor changes to this icon.

A short cut method for positioning the X and Y axes of a displacement map onto the surface of a face is also available. It is applied by pressing option (Macintosh) or ctrl+shift (Windows) when a face is selected to make it active.
Selecting a displacement map

The amount of displacement is determined by the gray scale value of the mapped location of a point in the displacement map. A preview of the currently selected displacement map is shown in the lower left corner of the Displacement Map Edit dialog. When clicking in the preview box, the Displacement Map Options dialog is invoked (Figure 4.11.7.3). Inside this dialog a new displacement map can be loaded or the existing displacement map may be edited.

View...: When you click on this button, the displacement map is displayed at its original size in its own window, as illustrated in Figure 4.11.7.4. Inside this window, the boundaries of the map can be further adjusted by resizing and dragging the red rectangle.

Load...: When you click on this button, a new displacement map can be loaded from an image file. Any bit map based file type can be used for a displacement map.

Default: When this option is selected, the displacement map is reset to the system default map.

Revert: When you click on this button, the displacement map is reset to the state it was at, when the Displacement Map Options dialog was last entered.

Use Alpha Channel: When this option is selected, the gray scale value in the map’s 4th (or alpha) channel is used. Otherwise, the average gray scale value of the map’s red, green, and blue channels are used for displacement.

Invert: When this option is on, the gray scale values of the displacement map are inverted, resulting in an object that is the exact opposite of the displacement generated with this option turned off.
Displacement mapping methods

Displacement maps can be projected onto objects using one of three available methods. These types are selected from the Mapping Type pop up menu located on the middle right portion of the Displacement Map Edit dialog (Figure 4.11.7.5).

**Flat**: This type of mapping uses a flat placement of the map, which can be on any plane in 3D space. By default the map is placed on the XY plane of the Cartesian coordinate system. This position can be subsequently changed using the positioning tools available in the Displacement Map Edit dialog.

A displacement map has its own coordinate system, whose X and Y axes lie on the 3D plane. The map is placed with its lower left point at the origin of the coordinate system. Then it is repeated horizontally and vertically. The map is then projected onto the points of the object in a direction parallel to the Z axis of the coordinate system. In other words, the displacement value of a point results from its intersection with the projection of the map. The point is then moved along the Z axis of the coordinate system by its displacement value.

The flat method of mapping a displacement is illustrated in Figure 4.11.7.6. The map is positioned parallel to the XY plane and it is projected in a direction parallel to the Z axis.
**Cylindrical**: This type maps the displacement using a cylinder which bounds the object. The displacement map is placed on the round surface of the cylinder and is projected onto the points towards its central axis. A point is moved along the direction that is perpendicular to the cylinder’s axis by its displacement value. A cylindrical mapping is illustrated in Figure 4.11.7.7.

![Cylindrical displacement mapping](image1)

**Spherical**: This type maps the displacement using a sphere which bounds the object. The sphere’s origin is at the center and its poles are on the Z axis of the displacement coordinate system. The displacement map is placed on the surface of the sphere starting with its largest diameter, which corresponds to the horizontal direction. As the Y direction of the map converges to the poles of the sphere, the map is scaled as rows are laid out. The spherical map is projected towards the center of the sphere onto the points of the object. A point is moved away from the sphere’s center by its displacement value. Spherical mapping is illustrated in Figure 4.11.7.8.

![Spherical displacement mapping](image2)
Defining the size and amplitude of a displacement

The size of the displacement map as it is projected onto an object is determined by the values in the Size fields, found at the middle right portion of the Displacement Map Edit dialog.

Flip: With this option on, the horizontal and vertical directions of the displacement are reversed.

Horizontal/Vertical Tiling: The values entered in these fields determine the size of the displacement map projected onto an object.

Size: The meaning of the values entered in these fields depends on the type of mapping that is applied, as follows:

- **Flat**: The size values represent lengths expressed in the current unit of measurement (English or metric). The Horizontal Size determines the X and the Vertical Size the Y dimensions of the displacement map.

- **Cylindrical**: The Horizontal Size value represents an angle expressed in degrees. This angle is measured on the XY plane of the displacement coordinate system, and has its tip on the axis of the bounding cylinder. The Vertical Size value represents a length measured along the Z axis of the displacement coordinate system.

- **Spherical**: Both Size values represent angles in degrees. Horizontal Size measures an angle on the XY plane and its tip is on the Z axis. Vertical Size measures an angle on the ZX plane and its tip is on the Y axis.

Infinite: When this option is selected, the displacement map is repeated as many times as necessary to cover the object in the horizontal and vertical directions.

Times: When this option is selected, the displacement map is repeated as many times as indicated in the text field next to the radio button.

Center: When this option is selected, the displacement map is centered in the horizontal and/or vertical direction, relative to its origin. When this option is off, the lower left point of the displacement map is placed on the displacement origin.

Mirror: When this option is selected, the displacement map is mirrored in the horizontal and/or vertical direction.

Lock Size To: This menu contains five entries, allowing you to lock the proportions of the displacement map to a specific value. That is, if the horizontal size is edited, the vertical size is adjusted automatically to maintain the selected proportions, and vice versa.

- None: With this option, the proportions of the displacement map are not adjusted.

- Square Tile: With this option, the horizontal and vertical tile sizes are kept equal.

- Current Proportions: When this option is selected, the horizontal or vertical tile sizes are adjusted to the proportions present at the time this menu item was last selected.
**Displacement Map:** When this option is selected, the horizontal or vertical sizes are adjusted to the proportions of the displacement map image.

**Object Extent:** When this option is selected, the horizontal and vertical tile sizes are automatically adjusted, so that a round (integer) number of copies of the displacement map fit exactly across the object. How many times the displacement map will be repeated is determined by the values entered in the horizontal and vertical **Times** fields. For example, to place a map to fit exactly once on an object, select this option from the menu, enter “1” in both **Times** fields, and turn on the horizontal and vertical **Center** option.

When cylindrical mapping is used, the horizontal size is expressed in degrees while the vertical size is expressed in linear units. The equality of these different units of measurement is calculated as follows: When calculating degrees from length units, the length units are used to subdivide the perimeter of the bounding cylinder, and the end points of one subdivision are projected onto the axis of the cylinder. The angle between the two projection lines is the angle that is considered to be equal to the length unit. When calculating a length from an angle, it is given by the distance between the two points where the sides of the angle intersect the curved surface of the bounding cylinder. These calculations are also used when switching from a mapping whose sizes are expressed as units of length (such as flat), to a mapping whose sizes are expressed in degrees (such as spherical), and vice versa.

The gray scale value in the displacement map indicates the amount of displacement applied to a point, where white indicates maximum displacement and black represents minimum displacement. The distance of **Max** and **Min Displacement** are specified in the respective text fields. If a displacement map is mostly black, with the maximum displacement features represented as white areas and the **Min** value set to 0, the displaced geometry will be extruded from the original object. If a map is mostly gray with the minimum displacement features black and the maximum displacement features white, setting the **Min** value to the negative of the **Max** value generates concave and convex displacements.

**Smoothness:** The value set through this slider bar determines the smoothness of the displacement. If set to 0 the original displacement map is used. Larger smoothness values apply a blur factor to the map softening areas of sharp contrast, resulting in a smoother displacement. However, larger values will also result in a loss of detail. An example of a displacement map applied at different levels of smoothness is shown in Figure 4.11.7.9.

![Figure 4.11.7.9: Applying different levels of Smoothness to a displacement: left to right values are increasingly larger.](image)
Increasing the resolution of the displaced object

In general, objects to which a displacement is applied need to have a sufficient number of points to imprint a reasonably detailed shape. If the initial resolution of an object is too rough, additional faces can be generated by selecting Adaptive Meshing. This option recursively subdivides the original faces as needed to match the details of a displacement. A face is subdivided by splitting it with lines that start at its center and end at the midpoints of its edges. The three available options determine how many times a face is subdivided. Note that this subdivision method is similar to that used by the Q-Subz tool and most of the available options have very similar effects (see subsection 4.10.2).

**Max # Of Subdivisions**: The value entered in this field indicates the maximum number of times a face may be subdivided. Note that, even small values for this parameter may generate a large number of faces. To restrict the number of faces, additional constraints may be specified, by selecting one of the following options.

**Max Segment Length**: Selecting this option causes the subdivision of a face to be terminated, as soon as all the segments of a face derived from the original face are less than the value specified in the text field.

**Threshold**: In areas of a displacement map where the gray values are the same or almost the same, the displaced object will show only a few or no changes. Therefore, it is not necessary to subdivide the faces of such areas. In contrast, in areas where the displacement map contains significant gray scale differences, the object requires a sufficient level of detail. The displacement operation detects the gray scale differences by sampling the color values at the center and each corner point of a face. If the difference between these values is less than the amount indicated by this slider bar, the subdivision of the face is terminated. The lower the Threshold value, the more sensitive the subdivision becomes relative to changes in gray scale.

Note that, under certain conditions, the sampling may produce incorrect results. This happens when the sampled points are found to have the same gray scale value, but in fact different gray scale values exist between the sampled points. The likelihood of this occurring increases when an object consists of a few large faces. Such cases require assistance by the user. That is, it is recommended that you first mesh the object to a finer resolution, using either the Mesh or the Q-Subz tool (see subsections 4.10.1 and 4.10.2).

**Triangulate Mesh**: When this option is on, the resulting mesh is triangulated. This triangulation is different from the triangulation of the Triangulate tool (see section 4.10.5).

**Smooth Mesh**: When this option is on, any new points generated for the mesh are first positioned on the surface of a sphere or a cylinder, if spherical or cylindrical mapping is selected, respectively. This results in a smoother mesh, because the meshed object already has the shape of a sphere or cylinder at the time the displacement is applied. If this option is off, the meshed object retains its original shape, at the time the displacement is applied. This is illustrated in Figure 4.11.7.10: in (a) the Smooth Mesh option is off, while in (b) it is on.
Applying a displacement to individual faces

A displacement map may be applied to complete objects or individual faces. If a displacement is applied to an object for the first time using the prepick method, and one or more faces are preselected, the preview window in the Displacement Map Edit dialog displays only these preselected faces in the highlight color. The other faces are displayed in the ghost color (gray) and are ignored by the displacement operation. The selection of faces to be displaced can be revised (more faces can be selected, or currently selected faces can be deselected) while the Displacement Map Edit dialog is open.

The Arrow tool, located under the preview window, is used to pick additional faces or to deselect previously picked faces, like the Pick tool in the main window. Faces can be selected/deselected with either one or two mouse clicks, depending on whether the Clicking On Edges or Clicking Inside Boundaries option is selected in the Pick Options dialog.

Pressing command + A (Macintosh) or option + shift + A (Windows) selects all the faces. Pressing command + tab (Macintosh) or option + shift + tab (Windows) deselects all faces.

To select faces that may be behind another face, the pick parade method can be used, as discussed in section 4.3.3.

When a face is picked to be added to the displacement, the displacement coordinate system can also be placed on it. This is done by pressing the option key (Macintosh) or ctrl + shift keys (Windows), when selecting the face. The origin of the coordinate system is always placed at the center of the face, the X and Y axes are placed on the surface of the face, and the Z axis is always perpendicular to the face. The orientation of the X and Y axes depends on how the face is picked. When the Click On Edges method is used to select the face, then the orientation of the X axis takes the orientation of the first segment selected. The location of the positive portion of the Y axis is determined by the position of the second segment selected, relative to the first segment selected. When Clicking Inside Boundaries is used, the X axis is parallel to the segment that is closest to the pick point. The positive Y axis is always to the left of the positive X axis. Pressing option or ctrl+shift when selecting faces with the Lasso or Frame has no effect.

Figure 4.11.7.10: Spherical displacement applied with the Smooth Mesh option (a) off and (b) on.
Examples

The success of a displacement operation greatly depends on the structure of the gray scale displacement map. In general it is necessary to avoid areas with very strong contrasts. That is, the more gradual the shade changes in the map are, the smoother the displacement of the geometry of an object will be. In preparing displacement maps, image compositing programs, such as PhotoShop, can be used to produce gradual gray scale transitions, by applying a blur to the map. In addition, the Smoothness parameter may be used to soften a displacement.

The structure and quality of a displaced object is also greatly affected by the mapping type that is used. For example, a map which works well for Flat mapping, may not be appropriate for Spherical mapping, and vice versa.

The following examples illustrate the use of image-based displacement to create a human face and a terrain model. The first example uses a displacement map to extrude the features of a face from a simple square face. Note, that the basic color of the map is black, with the face features represented as gray scale values. This will generate the round forehead and cheek from the flat surface, as shown in Figure 4.11.7.11. The second example (Figure 4.11.7.12) uses a displacement map to generate a face from a sphere. Note, that the displacement map looks quite different. First, the map is twice as wide as it is high to compensate for the fact that the distance around the equator of the sphere is twice as long as the distance from pole to pole. Second, the base color of the map is a medium gray. The features to generate forehead and cheeks are not as strong, since the sphere already has the basic shape of a face.

Displacement maps can be used to generate a variety of objects whose surfaces are sculpted, such as terrain models. For the latter, the displacement map may actually be scanned from a real map, where the heights of the terrain are color coded. In a paint program, the color can be converted, so that the highest areas in the map are white and valleys are dark. An example of a terrain model created using the Displacement tool is shown in Figure 4.11.7.13.
Figure 4.11.7.11: Using Displacement to create a human face on a flat surface.

Figure 4.11.7.12: Using Displacement to create a human head from a spherical object.

Figure 4.11.7.13: Using Displacement to create a terrain model on a flat surface.
4.12 Rounding, beveling, blending, and draft angles

This section discusses the following six tools:

- Plain Round
- Control Round
- Stitch Round
- Blend
- Fillet
- Draft Angle

As the names of the top three tools above imply, form•Z offers three types of rounding: **plain** or **direct** rounding, **controlled** rounding, and **stitch** rounding. All types of rounding can be applied to both **facetted** and **smooth** objects. Also the rounding itself can be facetted or smooth, depending on the type of object that is rounded and options selected in respective dialogs.

When you use plain rounding, you can only apply one type of operation at a time (round point, edge, or point and edge) and you can only use uniform rounding parameters. However, plain rounding allows you to apply multiple layers of rounding. Each subsequent rounding layer can apply a different rounding operation and/or different rounding parameters. With proper planning to select parameters that fit, multilayer rounding offers enormous possibilities.

In contrast to plain rounding, which is always applied on top of what is already there and does not reevaluate previously applied rounding, controlled rounding recalculates all the rounding from scratch each time it is applied. This is possible because controlled rounding remembers what rounding parameters were assigned and reuses them, which is not the case with plain rounding. In controlled rounding, the rounding parameters may be assigned to entities one at a time, which makes it possible to use different parameters and to mix them in the same object. Since controlled rounding always recalculates the rounding from scratch, it is able to deal with cases where edges with different radii meet. Note however that, for facetted rounding, only uniform resolution can be applied. This is the number of points used in the rounding arcs. If assignments with different resolutions are made, the highest number will be used uniformly throughout the object.

**Facetted stitch** rounding is applied to closed sequences of segments that meet at angles of no less than 100°. **Smooth stitch** rounding is applied to open or closed sequences of curved edges that meet at angles of no less than 180°. That is, edges that are tangent to each other.

Blending creates smooth transitions between open edges of surfaces, while Fillet creates a round fillet at the line of intersection of two solid objects or surfaces. They both produce smooth surfaces, which can also be facetted through conversion operations.

The last tool applies to faces of solid objects a slight tapering known as **draft angle**. These angles are useful for removing physical objects from their mold when injection molding is used to manufacture objects from their computer models.
4.12.1 Rounding objects directly

Plain Round

This tool is used to execute one of the available types of direct rounding, which is selected from the Plain Rounding Options dialog (Figure 4.12.1.1), invoked directly from the Plain Round tool. Both faceted and smooth objects can be rounded with this tool.

Figure 4.12.1.1: (a) The Plain Rounding Options dialog with the Circular options shown. (b) The Elliptical and (c) the Bevel options.
Facetted rounding

**Facetted Rounding Of Facetted Objects:** When this option is on, faceted objects will be rounded with facetted curves, according to the parameters set in the box of this option, which includes the three available types of facetted rounding, **Vertex**, **Edge**, and **Edges & Vertices** (see examples in Figure 4.12.1.2). When this option is off, smooth rounding will be applied to a selected object, regardless of whether it is facetted or smooth. Note that facetted rounding can not be applied to a smooth object.

Plain rounding is executed using either the postpick or the prepick method. With the postpick method, select the type of rounding in the **Plain Rounding Options** dialog and activate the Plain Round tool, select the appropriate topological level (Point, Segment, Outline, Face, or Object), then pick the entity you wish to round. The operation is executed immediately. Plain rounding is always applied to the object itself, rather than to a copy of the object, and the **Status Of Object** options do not apply.

While the postpick method allows you to round one entity at a time, the prepick method allows you to preselect any number of entities at different topological levels, to be rounded together.

**Rounding Type:** The type of plain facetted rounding is selected from this group.

**Vertex:** The round point operation can be applied at the Point, Segment, Face, Outline, or Object level and rounds the points that are included in the respective entity. That is, at the Point level it rounds a single point, while at the Object level it rounds all the points of the object. Only points that are completely convex or concave can be rounded. Mixed points (partly convex and partly concave) can not be rounded by themselves. This restriction is illustrated in Figure 4.12.1.3.

**High Density:** Corner points can be rounded at two levels of resolution. When this option is selected, more faces will be generated for the rounding of individual points. Note that this option only affects **Vertex** rounding. It is dimmed and inactive for the other types of rounding. This is illustrated in Figure 4.12.1.4.
**Edge:** The round edge operation can be applied to points, individual segments, outlines, faces, or objects, using the postpick method. Examples are shown in Figure 4.12.1.5. When applying edge rounding to a point, all the segments that converge to (share) that point are rounded. Note that the result produced by edge rounding a point is substantially different from point rounding when applied to the same point.

![Figure 4.12.1.5: Rounding (a) the edges converging to a point, (b) individual edges, (c) a face, and (d) a complete cube. Rounding (e) a face and (f) a complete pyramid.](image)

The prepick method can also be used. With the Pick tool, preselect any number of entities, which may be at different topological levels, select the Plain Round tool, and click anywhere in the graphics window. The system will collect all the segments that have been picked for rounding, and will construct rounding solutions as appropriate. The rounding at points where edges meet will vary depending on how many edges converge to them. Open ends of segments work best when they are three-segment points. When they are points to which more than three segments converge, then the rounding also converges to these points.

**Edges & Vertices:** Simultaneous edge and point rounding rounds the points only when all the edges that converge to that point are also rounded (there is no rounding solution unless this condition holds). Conversely, mixed (partly convex and partly concave) points, all edges of which are rounded, will also be rounded even if Edges & Vertices is not selected. In these cases, there is no way the rounded edges can meet properly unless the point they contain is also rounded. These conditions are illustrated in Figure 4.12.1.6.

![Figure 4.12.1.6: (a) Edge and (b) Edges & Vertices rounding: Point (1) is concave and is rounded both times. Only two edges of point (2) are rounded and the point itself is rounded in neither case. Point (3) is not rounded in (a), and is in (b).](image)

Simultaneous edge and point rounding can be applied at all the topological levels (except Group) using either the postpick or the prepick method. When the prepick method is used, entities at different topological levels can be selected, but they should all be on a single object.
**Rounding Method:** The two options in this group determine how the numeric value entered in the **Rounding Size** field will be interpreted. When **Use Radius** is selected, the value will be used as a true radius. When **Use Distance** is selected, the value will be used to calculate an offset from the edge. The radius is then calculated from this position. For 90° angles the two options produce identical results. For angles wider than 90° the actual radius will be larger than the value entered. The reverse is true for angles less than 90°.

**# Of Rounding Points:** The value entered in this numeric field is the rounding resolution: how many points or edges will be used for the rounding. This number must be an even number and cannot be less than 2.

**Bevel:** When this option is selected, beveling will be applied rather than rounding. Beveling is affected by the **Rounding Method** options but not by the **# Of Rounding Points** parameter.

**Triangulate:** When this option is selected, all non planar faces that were produced by the rounding will be triangulated.

**Adjust Radius To Fit Angle:** The rounding parameters assigned to the entities of an object are first checked by the system to make certain that there is enough space on the face or edge to fit them. When they do not fit, an error message is issued. The system may also be requested to try to adjust the sizes of the rounding parameters by selecting this option. The radius/distance values assigned to the entities will be adjusted up to a maximum of 25% (one quarter) of the original values. If, after all the proper adjustments have been made, the rounding parameters still fail to fit, an error message is issued. The process is of a global character. That is, the system adjusts all the parameters until a fit is achieved.
Figure 4.12.1.7 illustrates how plain rounding can be applied in one or multiple layers to achieve a variety of results.

A single layer of rounding is applied to the examples of row 1, using the postpick method: (a) Corner Point, (b) Edge, and (c) Edges & Points rounding is applied at the Object level. Note that in 1(a) the three mixed points have not been rounded since there is no point rounding solution for these cases.

A single layer of rounding has also been applied to the examples of row 2, however, the prepick method is used to select multiple entities: (a) Four segments are preselected and edge rounding is applied. (b) Seven segments not all on the same face are preselected and edge rounding is applied (edges & points rounding would have produced the same results). (c) Two faces (top and front) are preselected and edges & points rounding is applied. Note that only two points are rounded, since they are the only ones with all their edges rounded.

Multiple layers of rounding are applied to the examples of row 3: (a) 6’ Beveling is first applied to the four upper front segments. Then 1’ edge rounding is applied to six more segments. Both times the segments were preselected. (b) 6’ edge rounding is applied to all the vertical segments, after preselecting them. Then the top face is rounded using radius of 2’ and finally the bottom face is beveled again using radius 2’. (c) 6’ edges & points rounding is applied to the concave point. Then the resulting segments around the perimeter of the cavity are preselected and 2’ edge rounding is applied.
Smooth rounding

Smooth Options: The options in this group take effect when Facetted Rounding Of Facetted Objects is off, which results in smooth rounding to be applied to both facetted and smooth objects. There are options for two types of smooth rounding: Edge and Vertex. There are no options to select specific to combined Edge and Vertex rounding, even though the program will automatically round both edges and points, when a complete object is rounded. This is illustrated in Figure 4.12.1.8(e). Compare this to the example in Figure 4.12.1.8(f), where, even though all the edges of the object have been rounded, the points have not. This is done by rounding the edges individually, one or a few at a time.

Plain smooth rounding can be applied using either the prepick or postpick selection method. It can be applied to points, edges, outlines, faces, objects, groups, or holes, as follows:

- When applied to a point (vertex), the point and all the edges that converge to it are rounded (Figure 4.12.1.8(a)).
- When applied to an edge, the edge is rounded (Figure 4.12.1.8(b)). Note that, for smooth rounding, edges on surfaces with free ends may also be rounded, while for facetted rounding only edges of solids can be rounded.
- When applied to an outline (Figure 4.12.1.8(c)), face (Figure 4.12.1.8(d)), or hole, all the edges of that entity are rounded.
- When applied to an object, all the edges and the points of the object are rounded (Figure 4.12.1.8(e)).
- When applied to a group, all the edges and points of the objects in the group are rounded.

Edge Rounding: The options in this group set the parameters for rounding edges.

Edge Type: One of three items can be selected from this pop up menu: Circular, Elliptical, or Bevel. Each selection causes a different set of options to be displayed, as follows:
Circular: When this item is selected, circular rounding is applied to the edges. There are two methods for defining the size of the arc that will be used for the rounding.

Use Distance: When this method is selected, the value entered in its field is the length of the chord of the arc cross section.

Use Radius: When this method is selected, the values entered for radii will be used to derive the arc.

Start, End: The values entered in these fields are used to derive the arcs for the rounding. Different values can be entered in each of these fields, resulting in a rounding of an edge that starts with one radius and ends with another. There is a lock option between Start and End. When on, both ends of the rounded edge are locked to the same value.

Because all edges of the form-Z objects are pairs of reversely coincident segments, which of their end points is the Start and which is their End is relative to which of their faces one views them from. Consequently, for the purpose of rounding, the user determines which point is Start and which End by where he/she clicks to select the edge. The end point that is closest to the click point is the Start and the other end point is the End.

Weights: When this option is on, the rounding of an edge is not a full arc but its curvature may be reduced by pulling its sides closer to the straight line according to the values entered in its fields, as illustrated in Figure 4.12.1.9. These fields may contain independent values or they may be locked to the same value. The two fields correspond to the left and right ends of the rounding arc respectively.

Elliptical: When this item is selected, the rounding arc is elliptical rather than circular and its exact shape is determined by the following parameters:

Major Radius, Minor Radius: The values entered in these fields determine the shape of the rounding ellipsis, which can be independently set for the Start and End of an edge.

Rotation Angle: The elliptical rounding can also be rotated according to the values in these fields. When no rotation is desired, 0° is entered in these fields (default). The angle is between the normal to the reference face and the major axis of the ellipse.

Reference Face: This is the face relative to which the major and minor radii of the elliptical fillet are positioned. There are two choices: Left and Right.
Bevel: When this item is selected, a flat bevel is generated at the selected edge, rather than a round fillet. The parameters of the bevel are as follows:

**Left Distance, Right Distance:** These values specify the distance of the bevel break point from the vertex, separately for the left and right sides of the vertex and separately for the start and end points of an edge.

Bulge: When this option is on, a bulge is added to a bevel, which is not flat anymore. Two separate values specify the bulge for the start and end points of an edge. These values can range from 0 to 100.

**Vertex Rounding:** The parameters in this group of options control whether individual vertices will be rounded and how.

None: When this option is on, no rounding is applied to a vertex.

Circular: When this option is on, vertices are rounded. Note that this is the only option for rounding and, for example, elliptical rounding does not apply to vertices.

Bulge: This slider bar controls the level of bulge that is applied to the rounding of a vertex. It increases from left to right.

Setback: The number entered in this field represents how far a vertex will be rounded. Examples for three different values are shown in Figure 4.12.1.10.

*Figure 4.12.1.10:* Rounding vertices with **Setback** set to (a) 2', (b) 4', and (c) 6'.
4.12.2 Controlled rounding

Control Round

This tool is used to execute rounding different than the plain or direct rounding. The difference is that controlled rounding applies all the rounding operations that have been assigned to an object simultaneously, rather than one or a few at a time, which is the way of plain rounding. Whether the resulting rounded object will be faceted or smooth is determined by the type of the original object and by options in the Controlled Rounding Options dialog (Figure 4.12.2.1), invoked directly from the Control Round tool. It contains three tabs: Options, Display Resolution, and Status Of Objects. The content of the latter two are discussed in sections 4.1.1 and 4.5, respectively.

Controlled rounding can be applied to both faceted and smooth types of models. With smooth objects, the rounding is also smooth, always. With faceted objects the following options are available:

Facetted Objects: There are two options, when applying rounding to faceted objects.

  Keep Facetted: When this option is selected, the rounding that is applied is also faceted, which makes the whole object faceted.

  Make Smooth: When this option is on, smooth type of rounding is applied to an object, which makes the complete object smooth. As with all smooth objects, a smooth rounded object can subsequently be changed to a faceted object, using the Convert tool.

Also note that, when controlled rounding is applied, the resulting rounded object becomes a parametric object and subsequent applications of the Control Round tool can change its rounding parameters and re-execute the rounding.
Controlled rounding is applied when, with the Control Round tool active, you click on an object (postpick method). The prepick method can also be used to preselect an object, however, it produces the same result as the postpick method. Controlled rounding is executed through a **Round Edit** dialog, which appears as soon as you click on the object that will be rounded.

There are two variations of the **Round Edit** dialog, one for **Facetted** and one for **Smooth** rounding. They are shown in Figures 4.12.2.2 and 4.12.2.4, respectively. The former is invoked when you click on a facetted object and **Keep Facetted** is selected in the **Controlled Rounding Options** dialog. The latter is invoked when you click on a facetted object and **Make Smooth** is on, or you click on a smooth object. In other words, the former dialog contains the options for facetted rounding, while the latter contains the settings for smooth rounding. In all cases, the object on which you click with the Control Round tool may be rounded for the first time or it may have already been rounded before. In the latter case the object has preserved the previously assigned rounding parameters, which can be changed once in the rounding preview environment.

**Facetted rounding**

The **Facetted Round Edit** dialog consists of a preview window where both the original object (called **control** object) and the rounded object are displayed. The entities (points and segments) to which rounding parameters have been assigned are highlighted. Outside the preview there are options and command buttons that allow you to manipulate the preview display, as well as to assign, clear, or edit rounding parameters.

- **Use Radius**, **Use Distance**, **Adjust Radius To Fit Angle**, **Bevel**, **Triangulate**, **High Density**, **Radius**, and **# Of Points** are as for plain rounding.

- **Delay Rounding Size Checks**: Typically, with controlled facetted rounding, size checks and adjustments are made at the time rounding parameters are assigned, except when this option is selected, in which case the checks and adjustments are applied at the time the preview or the final object is generated.
Whether the rounding size check is performed one assignment at a time or is delayed until preview, may make a difference with regards to whether a parameter will be found to fit (and will thus be accepted) or not. This apparent discrepancy may arise when rounding polygonal faces, as is illustrated in Figure 4.12.2.3. In (a), where all but two segments of the top face were rounded, the largest radius possible is about half of what we were able to use in (b), where all the segments of the top face were rounded.

In controlled rounding, where the rounding parameters are assigned one segment at a time, if the size test is executed after each assignment, we shall never be able to assign a higher radius to all the segments of the top face. If on the other hand we delay the size test until all the assignments are completed, this becomes possible.

**Round, Clear, Edit:** Rounding parameters are assigned or cleared graphically by clicking the mouse on a point or a segment. What action is taken is determined by selection of one of these options. **Round** assigns the current rounding parameters to a point or a segment. **Clear** clears a point or a segment from its rounding parameters. **Edit** can be used to both read what rounding parameters are currently assigned to an entity and to change them. When it is selected and you click on an entity, its current parameters appear in the Radius and # Of Points fields. Either of them can be changed. The new value will be assigned to the entity as soon as the next action is taken.

**Preview:** This group of options determines what is displayed in the preview area.

- **Show Original Object, Show Round Object:** When these options are on, the original object and/or the rounded object are displayed.

- **Update:** When you click on this button the rounding parameters of your object will be reevaluated and the new version of your object will be displayed. If Automatic is on, then your object will be reevaluated every time you apply a rounding operation.

- **Quick Preview:** When this option is on, your rounding preview will be displayed at a resolution of 2 (# Of Points). The object will still be constructed with the actual resolution you have selected when you exit the rounding preview environment.

The icons in the tool palette under the preview window can be used to manipulate the shown view. They work as in all preview dialogs (see subsection 2.1.6).

**Plain Object:** If this option is off (default) when you exit the Round Edit dialog, a controlled object is produced. Such an object can be edited again and its rounding parameters can be further manipulated. If this option is on, a plain object is created. This may be useful in order to economize memory, after you have finalized your rounding decisions, or for applying multilayer rounding, as discussed in the previous section.
Smooth rounding

The Smooth Round Edit dialog also consists of a preview window where the original object and rounded object are displayed. Outside the preview there are options and command buttons for manipulating the preview display and for assigning rounding parameters.

Figure 4.12.2.4: The Smooth Round Edit dialog.

The parameters in the Edge Rounding and Vertex Rounding boxes are the same as those in the Smooth section of the Plain Rounding Options dialog. Consequently, they do not need to be discussed here. The options in the Preview box work as for the Facetted Round Edit dialog.
Entities (edges and vertices) can be selected in the preview window individually or in groups, as follows:

- To select a single edge, with the arrow icon active, you click on it and then turn on the type of rounding that needs to be applied to it (Circular, Elliptical, or Bevel).

- To select a group of edges, you click on the individual edges one at a time. When done picking all the edges you need, you turn on the type of rounding you wish to apply. This not only assigns the rounding parameters to the edges you selected, but it also constructs a group of the edges you just selected. To confirm this, click in the preview window, away from any selectable entity, which will deselect the previously selected edges. Then click on any of the edges you previously selected. The whole group is highlighted. Deselect this group again and continue constructing more groups.

- To add more edges to a previously constructed group, select the group and then click on the edge you wish to add. You can continue adding as many new edges as you desire. When done, deselect the group to complete the grouping. If the edge you are adding to a group is already a member of another group, you need to *shift-click* on it, which will remove it from the other group and will add it to the highlighted group.

- To remove an edge from a group, while the group is active, shift click on the edge you want removed.

- To dismantle a group completely, select the group and turn on the *None* option from the dialog.

When using different rounding sizes for the *Start* and *End* of an edge, the direction of the edge is significant. You control this direction by where you click to select an edge. The point closest to the pick point becomes the *Start* and the direction of the edge is graphically indicated with a red arrow. You can reverse the direction after its initial setting by clicking on the arrow.
4.12.3 Stitch rounding

For facetted objects, a **stitch** is a continuous and closed sequence of straight segments, which meet at angles of no less than 100°. For smooth objects, a stitch is a continuous sequence of curves that meet at 180°. Smooth stitches can be closed or open.

This tool is used to round stitches according to parameters selected from the **Stitch Rounding Options** dialog (Figure 4.12.3.1), invoked directly from the Stitch Round tool. It contains two tabs: **Options** and **Display Resolution**. The **Options** tab contains sections for both facetted and smooth rounding, labeled **Facetted Options** and **Smooth Options**, respectively. Facetted stitch rounding is applied when the selected object is facetted. By the same token, smooth stitch rounding is applied when the selected object is smooth.

The Stitch Round tool can be executed using either the postpick or the prepick method. Stitch rounding requires picking at the Segment level. When prepick is used and a number of segments that belong to the same stitch are preselected, only one will be used to detect the stitch, and the others will be ignored.

You select a stitch by selecting only one of its segments. The system will automatically detect the complete sequence of segments that constitute a stitch, provided such a sequence can be detected.

As with plain rounding, stitch rounding works on the original object.
Facetted stitch rounding

Facetted stitch rounding is applied when a stitch on a facetted object is selected. It is not restricted by the sizes of the segments in a stitch and is less sensitive to the lack of planarity.

**Distance:** The distance from the stitch where the rounding will start and end is entered in this numeric field.

**# Of Rounding Points:** The number in this field represents the resolution of the rounding.

Note that these two parameters are the same as those used for the direct rounding of stitches that result from Trim & Stitch operations, but they are independent from them.

Facetted stitches are rounded by first creating three control lines, and then using them to generate a mesh. The control line in the middle is derived directly from the stitch line. The control lines at the two ends of the mesh are created to the “left” and “right” of the stitch at the **Distance** set in the **Stitch Round Options** dialog, and in such a manner that they lie exactly on the surfaces on the two sides of the stitch line. Note that this type of rounding is distinctly different from the plain and controlled rounding, which fit arcs to the edges of neighboring faces. Consequently, stitch rounding is generally more tolerant to non-planar surfaces and can handle non-planar sequences of segments. However, it will not handle stitches with steep turns, since such cases do not have proper rounding solutions.

![Diagram](image)

*Figure 4.12.3.2:* (a) An object, five stitches of which are rounded after picking segments where shown. (b) The resulting stitches after the application of five rounding operations.

Stitch rounding is applied in a “depth” direction only, that is between the three control lines. The resolution of the rounding is controlled through the **# Of Rounding Points** option in the **Stitch Round Options** dialog. Given that the end control lines are required to fit precisely on the surfaces on each side of the stitch, they are not rounded, since rounding them would distort the original shapes. Consequently, whenever smoother surfaces are required along the length of a stitch, they should be created before the stitch is derived, and certainly before the stitch is rounded. Examples of faceted stitch rounding are shown in Figure 4.12.3.2.
Smooth stitch rounding

Smooth stitch rounding is applied when a stitch on a smooth object is selected. Contrary to the facetted stitch, which is required to be closed, a smooth stitch can be open or closed. It may consist of a single curved segment or a sequence of segments. In the latter case, neighboring segments need to be tangent (or meet at 180° angles). Note that this is stricter than the facetted stitches, where the segments can meet at up to 100° angles.

The different types of smooth stitch rounding (Circular, Elliptical, or Bevel) and their parameters are selected from the Stitch Rounding Options dialog. They are as for plain rounding. Examples of rounding closed and open smooth stitches are shown in Figures 4.12.3.3 and 4.12.3.4, respectfully.

![Figure 4.12.3.3: Rounding closed smooth stitches:](image-a)
(a) With the Stitch Round tool click where shown.
(b) The resulting rounding (3 circular and 2 bevel).

![Figure 4.12.3.4: Rounding open smooth stitches:](image-b)
(a) With the Stitch Round tool click where shown.
(b) The resulting rounding (1 circular and 1 bevel).
4.12.4 Rounding cases to watch

The Use Radius versus Use Distance options

Assume that the cuboid and cylinder shown in Figure 4.12.4.1(a) have been unioned resulting in the object shown in 4.12.4.1(b). We shall round the vector line (outline) where the cylinder intersects the cuboid.

• Set the topological level to Outline.
• Double click on the Plain Round tool to invoke the Plain Round Options dialog.
• Select: Edge, Use Radius, and for Radius and # Of Rounding Points use the defaults: 4' and 4.
• With the Plain Round tool active, select the section outline. The result should be as in 4.12.4.1(c-e).
• Undo, invoke the Plain Round Options dialog again and select Use Distance.
• Execute the rounding operation as above. The result should be as in 4.12.4.1(f-h).

Observe the difference between the Use Radius (e) and Use Distance options (h). The former applies an arc which is larger and deeper for angles less than 90° than it is for angles greater than 90°. For this option the radius of the arc is exactly the same for all angles. Use Distance applies variable arcs whose radii depend on the local angle of the rounding. The sizes of the arcs are about the same, regardless of the local angle. Note that these distinctions affect both facetted and smooth rounding.

Rounding facetted objects with non planar faces

We can continue a facetted version of the previous example and also round the top end of the cylinder, as shown in Figure 4.12.4.2. You repeat the rounding operation as above, except that this time, the face at the top end of the cylinder is selected. A warning about the object having non planar faces is issued. If you click on OK the top face will be properly rounded, since the non planar faces of the object are in the area previously rounded. In general, when the warning is issued, you can still complete the rounding operation. However, if you are rounding segments that belong to non planar faces, the results are not guaranteed.

Figure 4.12.4.1: The Use Radius versus Use Distance options.

Figure 4.12.4.2: Rounding in spite of the non planar faces warning.
Rounding a faceted stitch with the fillet fitting method

A *stitch* is a continuous and closed sequence of segments, typically resulting from a Trim & Stitch or a Boolean operation. Stitches may also be found in objects that are created directly or through other derivative operations. The term is used for sequences of segments that are not outlines. However, outlines can also be rounded as stitches. This distinction is illustrated in Figure 4.12.4.3.

When rounding stitches using plain rounding (the fillet fitting method), they can be selected (picked) by selecting their individual segments. The order in which the segments are picked is not significant. The group of selected segments should form a complete and single sequence. If the selected segments consist of more than one sequence, they will not be recognized as a stitch.

- Generate two cylinders as shown in 4.12.4.4(a). The vertical (large) cylinder has a radius of 32' and the horizontal (small) cylinder has a radius of 4'. The alignment of the cylinders is significant. Use snapping and position the top and bottom segments of the small cylinder on the front segment of the large cylinder, as shown in the front view detail of Figure 4.12.4.4(c).
- Difference the small from the large cylinder. The result should be as shown in 4.12.4.4(b). The vector line at which the two cylinders intersect is a *true stitch* and not an outline. It can only be selected through its individual segments, one segment at a time, or by frame picking, as shown in 4.12.4.4(d).
- Preselect the stitch (with topological level at Segment).
- Invoke the **Plain Round Options** dialog and select **Edge**. For the numeric parameters use the defaults (4' and 4).
- With the Plain Round tool active click anywhere in the graphics window. The result should be as in 4.12.4.4(e) and (f).

Lining up the segments is necessary to be able to apply reasonably large radii. Figures 4.12.4.4(g) and (h) illustrate a case where the segments of the cylinders do not line up. As shown in Figure 4.12.4.4 (i) and (j), 6" was the largest radius we were able to apply to the rounding of the stitch.

An even worse case would be when the corresponding segments appear to be lined up but are not. This results in one or more very small segments which would make it virtually impossible to apply fillet rounding. Such cases should be avoided.

Figure 4.12.4.3:
(a) A stitch and (b) an outline.

Figure 4.12.4.4:
Rounding stitches.
Applying variable radii to faceted rounding

Different size radii can be applied to different segments only when using controlled rounding. It can be done as follows:

• Generate a faceted cylinder as shown in 4.12.4.5(a). It has a radius of 32’. If you generate a cylinder of a different size you will also have to adjust the radii suggested below.

• Invoke the Controlled Rounding Options dialog and select Keep Facetted.

• With topological level at Object and the Control Round tool active click on the cylinder. This takes you into the Facetted Round Edit dialog.

• Select Delay Rounding Size Checks and enter 7’ in the Radius field. Then click on two opposite segments of the top face of the cylinder. They should be highlighted as shown in 4.12.4.5(b).

• Change the Radius field to 6’, then click on segments on the left and right of each of the previously highlighted segments. You now have a total of six segments highlighted.

• Change the Radius field to 5’, then 4’, then 3’, then 2’, and finally 1’. Each time you change (reduce) the number in the Radius field select four more segments adjacent to the previous selection. With the last value (1’) you will only have two segments left to select (Figure 4.12.4.5(c)).

• After all the segments have been assigned a radius, click on Preview to review what you have. If the preview is satisfactory, click on OK, which should produce the rounded object shown in 4.12.4.5(d) and (e).

If for some reason the result is not as shown, you should go back and correct the assignments of the radii. The logic is to start with a high radius value, sequentially change to a small radius, and then return to the high value. For best results, you should assign rounding to all the segments. After you complete the shown example, feel free to experiment with additional variations.
Facetted rounding cases to watch

As already discussed, there will be cases for which no proper facetted rounding solution exists. The most severe of such cases will be denied by the system. For example, as was illustrated in Figure 4.12.1.7(1a), partially convex and partially concave points will not be rounded, when point rounding is applied. In other cases, the system will apply what is geometrically possible and present a solution that is not acceptable. This is intended to offer feedback to the user, who, once the improper conditions are recognized, may find a way to adjust the rounding. Two such cases are illustrated in Figure 4.12.4.6, with suggested solutions.

In 1(a), the front middle face has edges whose angles are in opposite directions. When these edges are rounded, they pull each other where they meet, resulting in a flat rounding structure. There is no proper rounding solution for this case except as shown in 1(b). The points should also be rounded, which is only possible when the adjacent segments are also rounded.

In 2(a), a very large radius is applied to one of the edges, and significantly smaller radii (1/10 of the first) are applied to the other two edges of a 3-edge point. Note that this type of assignment is only possible with controlled rounding. This again is a case for which no proper rounding solution exists, except as shown in 2(b). Two layer plain rounding is applied. The vertical edge is rounded first using the large radius, then the top front edges are rounded using the smaller radius.

There are also cases where the results are correct but may surprise you and may at first glance appear incorrect. Two such cases are illustrated in Figure 4.12.4.7. The two edges of the concave point in (a) have angles in opposite directions, thus where they meet they merge into a straight line. In (b), a single side edge of a pyramid is rounded, which by necessity converges to the apex of the pyramid. To also round the apex, all the side edges should be rounded.

Figure 4.12.4.6:
(a) Cases with no proper rounding solution, except as in (b).

Figure 4.12.4.7:
Two cases that may appear incorrect, but are actually correct.
4.12.5 Blending

The Blend tool is used to create smooth transitions or blends between two edges, two outlines, or two faces. These entities can be from different objects or the same object and their shapes must either both be closed or open. Note that this tool cannot handle faces with holes, which are ignored when such faces are selected. The Blend tool may result in an independent smooth object that can be a solid, or the blend surface may be stitched to the original objects, which is controlled by options in the Blend Options dialog (Figure 4.12.5.1) that is invoked from the tool.

Either the prepick or postpick method can be used. With the postpick method, the Blend tool is selected first and then two edges, two outlines, or two faces. With the prepick method, any number of edges, outlines, or faces are picked first, followed by the selection of the Blend tool and a mouse click anywhere in the graphics window. The Blend tool handles the selected entities in pairs, in the order they were picked. Examples are shown in Figures 4.12.5.2 and 4.12.5.3.

While for the Blend tool you can select segments, outlines, or faces, the blend operation is always applied to faces. When these faces have open edges, then they can be uniquely picked through these edges. This is illustrated in Figure 4.12.5.2. In (a) the side faces of the cylinders are blended, but because the tops and bottoms of these cylinders are open, the side faces can be picked by selecting open edges, as shown. Similarly, the two squares in (b) can be selected through their edges, as shown.

Were the cylinders in the examples solids, the faces to be blended would have to be selected as faces, or otherwise the results would be unpredictable. In addition, when selecting the faces with two clicks on edges, the order of the clicks is significant and dictates the direction in which the selected faces will be blended. This is illustrated in Figure 4.12.5.3.

Figure 4.12.5.1: The Blend Options dialog.

Figure 4.12.5.2: Blend examples between faces with open edges (picked through edges).
Figure 4.12.5.3: Blending two solid cylinders. The results vary depending on which faces are picked and how they are picked.
**Start Parameters, End Parameters**: These two groups determine the blending parameters at the start (entity selected first) and the end (entity selected second) of the blend.

**Automatic**: When this option is on, the bulge is automatically calculated by the system.

**Custom**: When this option is on, the bulge is determined by the values entered by the user.

**Bulge**: This slider determines the size of the bulge. A value of 0 results in no bulging and a sharp transition. The higher the bulge, the more distinct the effect on the blend. A value of 100 is recommended as the maximum. However, larger values can also be entered to generate rather extreme results, if desirable. Examples are shown in Figure 4.12.5.4.

**Reverse**: By default, this option is off which results in a blending direction that is away from the blending edge. If this option is on, the blend is constructed towards the inside of the blend edge. Examples are shown in Figure 4.12.5.5.

*Figure 4.12.5.4: Custom blending with Bulge = :* (a) 0% at both Start and End; (b) 50% at both Start and End; (c) 10% at Start and 60% at End; (d) 100% at both Start and End.

*Figure 4.12.5.5: Blending with Reverse: * (a) off at both Start and End, (b) on at Start and off at End, (c) off at Start and on at End, and (d) on at both Start and End.
Make Solid: When this option is selected and the shapes to be blended are closed, faces are constructed at the start and the end of the blend to make a solid object.

Stitch Start, Stitch End: When this option is on and the corresponding edge or outline is an open edge of a surface object, then the blend is stitched to the edge making it a single object.

Preview: When this option is selected the Blend Edit preview dialog, shown in Figure 4.12.5.6 is invoked to allow interactive editing of the blend. This dialog has a standard preview window on the left and the options from the Blend Options dialog, with the following additional options.

Update: Clicking on this button applies the new blend parameters and refreshes the preview window. This button is dimmed when the parameters of the preview are up to date.

Automatic: When this option is on, the preview is automatically updated every time a parameter is changed. When this option is on, the Update button is dimmed.

Figure 4.12.5.6: The Blend Edit preview dialog.
4.12.6 Filleting

The Fillet tool is used to create fillets or rounded edges between two faces of existing objects. The faces can be disjoint or they can be intersecting. They can be from the same or different objects. The result of the operation is a new object with a fillet that fuses the objects between the selected faces.

The Fillet tool works at a **Basic** and an **Advanced** level. The former creates a fillet with an arc cross section that is constructed at a preset radius and can work on all faces. The latter offers more control over the fillet cross section. However, it is restricted to faces that have an extended line of intersection. Consequently, advanced filleting can not be applied to objects such as spheres that do not overlap. These and other options are selected from the **Fillet Options** dialog (Figure 4.12.6.1) that is invoked directly from the tool.

Either the prepick or postpick method can be used with this tool. With the postpick method, the Fillet tool is selected first and then two faces are picked. With the prepick method, any number of faces are picked first, followed by the selection of the Fillet tool and a mouse click anywhere in the graphics window. The Fillet tool will handle the selected entities in pairs, in the order they were picked. Note again that the Fillet tool always works at the Face level. While with the prepick method the Topological Level modifier should be at Face, with the postpick method entities will be picked at the Face level, even if the Topological Level is set to a level other than Face (but not to Segment). As an exception, surface objects may be selected by picking their open edges (with Topological Level set to Segment) because such edges uniquely identify the desired face.

**Fillet Type:** This pair of radio buttons determines the direction of the fillet relative to the selected faces. The two options are **Concave** and **Convex**. Examples are shown in Figure 4.12.6.2. Note that, typically, the **Concave/Convex** options do not both yield a result and if the correct option is not selected the operation will fail. Which option is appropriate depends on the directions of the selected faces. If both normals point outwards (relative to the edge that will be filleted), a convex fillet needs to be generated. If they point inwards, a concave fillet should be generated. If one normal points in and the other out, the operation will fail and neither a convex nor a concave fillet can be generated. It is recommended that the normals of the surfaces be displayed to be able to check their direction, before the fillet operation is executed.
Figure 4.12.6.2: Original faces (top row) and after applying the Fillet tool (lower row).

*Fillet Type: Convex* from (a) disjoint and (b) intersecting faces.

*Fillet Type: Concave* from (c) disjoint and (d) intersecting faces.

Note how in Figure 4.12.6.2 the directions of the normals match each other and how they are consistent with the type of fillet (concave or convex) that is generated. Also note how the faces are picked, which is shown in 4.12.6.2(a). In this particular example we could have also picked the faces through one of their open edges, such as 1 and 3, in 4.12.6.2(a).

Figure 4.12.6.3: Fillet tool applied to (a) intersecting faces and (b) non-intersecting faces.

**Basic:** When this option is on, a simple arc fillet is applied between the two selected shapes.

**Fillet Radius:** The value entered in this field determines the size of the arc of the fillet. Note that the size of the fillet should be appropriate (not too small and not too large) relative to the pair of faces that will be filleted, or otherwise the operation will fail.
**Figure 4.12.6.4**  Fillet examples between a sphere and a nurbz surface:
(a) original objects, (b) concave direction, and (c) convex direction.

**Advanced**: When this is selected the advanced fillet method is applied.

**Circular, Elliptical, Bevel**: These options determine the cross section of the fillet. They are the same as the edge options found in the smooth plain rounding and smooth control rounding (see section 4.12.2).

Note that, unlike the **Basic** fillet, the pick order and pick locations are significant for the **Advanced** fillet. For best results the faces should be selected using the **On Edges** option in the **Pick Options** dialog to identify the fillet faces. The first face selected is used as the orientation face for the **Elliptical** and **Bevel** cross sections. The first edge selected is used as the orientation edge for variable radius/distance fillets. For closed face edges, the click point is used as the start for variable radius/distance fillets.

**Figure 4.12.6.5**: **Advanced** filleting:
(a) original objects, (b) **Circular**, (c) **Elliptical**, and (d) **Bevel**.
**Preview**: When this option is selected the *Fillet Edit* preview dialog is invoked to allow interactive editing of the Fillet. This dialog has a standard preview window on the left and the options from the *Fillet Options* dialog with the following additional items.

**Update**: Clicking on this button applies the new blend parameters and refreshes the preview window. This item is dimmed when the preview is up to date with the parameters.

**Automatic**: When this option is selected, the preview is automatically updated, when a parameter is changed. When this option is selected the *Update* button is dimmed.

*Figure 4.12.6.6*: The *Fillet Edit* preview dialog.
4.12.7 Applying draft angles

The slight tapering that may be applied to faces of solid objects is called a **draft angle**. Draft angles are necessary when physical objects are manufactured through injection molding; they facilitate the removal of the objects from their mold, which otherwise may be wedged in the concavity of the mold.

![Draft Angle](image)

This tool is used for applying one or more draft angles to an object, using either the prepick or postpick selection method.

When using the postpick method, with the Draft Angle tool you click on the object on which the draft angles will be generated. When using the prepick method, you use the Pick tool to preselect any number of entities, then with the Draft Angle tool active you click anywhere in the graphics window to execute the operation. The Draft Angle tool can be applied at the Face, Object, or Group topological levels. Any other level causes the program to post an error message. Groups of objects are handled as individual objects.

The draft angle operation is applied to an object itself and no copy is produced. If a copy is desired, you should make a copy of the object before applying the Draft Angle tool.

The draft angle tapering is produced by rotating a face by a specified angle around an axis, defined by the intersection line of a face’s plane and a taper plane, as illustrated in Figure 4.12.7.1. This taper plane can be any of the orthogonal planes or an arbitrary reference plane, which may be used at the position they are or their position may be adjusted to the maximum or minimum extents of an object (or the group of selected faces). These and other available options can be selected from the **Draft Angle Options** dialog (Figure 4.12.7.2), which is invoked directly from the tool.

**Base Reference Plane**: This group of options allows you to select the plane that will be used as the base for the calculation of the draft angle.

![Draft Angle Options](image)

**Figure 4.12.7.1**: Calculating the axis of the draft angle.

**Figure 4.12.7.2**: The Draft Angle Options dialog.
**Active Plane**: Selection of this option results in the current reference plane being used for the draft angle. Note that this can be any of the Cartesian planes or an arbitrary plane, if one is defined and is currently the active plane.

**XY, YZ, ZX**: Selecting one of these options results in the respective Cartesian plane being used as the base plane for the draft angle.

**Position Of Base Plane**: This group of options determines whether the selected base plane will be used at its current position or whether it will be placed to a position calculated relative to the extents of the object (or group of faces) to which the draft angles are applied.

**At Current Position**: When this option is selected, the base reference plane remains at its current position. This option is illustrated in Figure 4.12.7.3.

**At Minimum Of Object/Faces**: When this option is selected, the base plane is moved in a direction perpendicular to it, until it touches the point of the object that has the lowest coordinate value in that direction. If the operation is applied to a group of faces (rather than the complete object) then the point closest to the base plane is a point of the group of faces. This option is illustrated in Figure 4.12.7.4.

**At Maximum Of Object/Faces**: This option is as the previous, except that the base plane is moved until it touches the point on the object (or group of faces) that has the greatest coordinate value in the direction perpendicular to the plane. This option is illustrated in Figure 4.12.7.5.

**Inclination Angle**: The value entered in this text field indicates the angle by which faces are tapered.

**Use Faces Within n Of Plane Normal**: When this option is selected, the draft angle is applied only to faces which have an inclination of less than $n^\circ$ relative to the normal (perpendicular) of the base plane, where $n$ is the value entered in the field of this option.

**Use All Faces**: When this option is selected, the draft angle is applied to all the faces that were selected.
4.13 Smooth splines and c-curves

This section discusses the following tools that generate and manipulate smooth curves (including NURBS, splines, and tangents) that may be derived from previously drawn control lines or can be drawn directly:

- Curve
- Reconstruct Curve
- Attach Curve
- Blend Curve
- Merge Curve
- Extend Curve
- Break Curve
- C-Curve
- Formula Curve

The first tool generates a variety of smooth curves, namely NURBS, splines, and tangent curves from previously drawn vector lines, which become its control lines. Note that the curves generated by this tool are to a certain extent redundant with the directly drawn splines (discussed in section 4.2.5), however, their creation process is different, which, from a practical point of view, makes them easier to use for certain application areas. Once generated, all the smooth curves can be edited the same way.

The second tool offers the ability to reconstruct a curve after its initial creation. This includes the ability to change both degrees and control points.

The next three tools offer different methods for joining curves together, either by a simple attachment or through a blending process. The next to the last tool offers the opposite, which is the ability to break a curve.

The next to the last tool offers the ability to generate a special type of curves, called control curves (c-curves). These are to a large extent based on standard NURBS and spline structures, but they also deviate in a few ways as they aim at accommodating some practical needs that are not covered by the pure NURBS. As a result, the c-curves are intended to be used within form-Z only, because their data structures are unknown to other programs and can not be easily exported. They can be exported only after they are converted to NURBS, which some times can only be done by altering their original shape.

The last tool generates curves from preset and prestored formulas or from formulas provided by a user.
4.13.1 Types of smooth curves

The new curve types are: NURBS curves, tangent curves, and spline curves. Each curve type has its own unique character and, depending on what shape needs to be drawn, you may choose one or the other. This selection is done in the **Curve Options** dialog (Figure 4.13.2.1), which is invoked from the tool and is discussed in more detail later in this section.

**NURBS curves**

A NURBS curve is defined through a set of control points, a weight factor for each control point, and a degree. Note that a NURBS curve does not pass through the control points, except for the first and last points of open curves.

The **weight** factor at a control point allows the curve to be pushed farther away from the control point or closer to it. If the weights of all the control points are the same, the curve is interpolated evenly between the control points.

The **degree** of a NURBS curve determines how flat the curve is. The higher the degree the flatter the curve. For all practical purposes, curves of degree 2 and 3 should be used. Degrees higher than 3 may significantly increase processing time. Some modeling applications are actually limited to NURBS curves of degree 3 and lower. When exporting to these applications, NURBS curves of a higher degree will not be compatible. NURBS curves of degree 1 are technically possible, but yield the same shape as the vector line selected. Therefore, **form-Z** does not offer NURBS curves of degree 1. Examples of NURBS curves are shown in Figure 4.13.1.1.

![Figure 4.13.1.1: NURBS curves.](image-url)
Tangent curves

Unlike NURBS curves, a tangent curve passes through the points of the vector line from which it is derived. Internally, a tangent curve is represented as a NURBS curve of degree 2 with equal weights, which makes it compatible for exporting. The NURBS control points of the tangent curve are calculated in such a way that the curve passes exactly through the vector line points. When a tangent curve is edited with the Edit Controls tool, the NURBS control points are not visible, but the tangent points are. This is discussed in more detail in the next section. Tangent curves also maintain a curvature parameter for each tangent point. This parameter determines how sharp the curve bends when it passes through the tangent point. Examples of Tangent curves are shown in Figure 4.13.1.2.

Spline curves

The spline curves generated with the Curve tool are actually the same as those which can be drawn directly with the Sketch Spline and Cubic B-Spline tools, except that the input points are taken from the selected vector line. The spline curves generated with the Curve tool may also create curves with multiple segments. This is analogous to the broken Bezier option of the c-curves. While the shape of the broken Bezier can only be represented as a single NURBS curve with sharp corners, a spline curve with multiple segments represents several, independent NURBS curves joined together. This representation is preferable for many modeling applications and also facilitates export compatibility. Examples of spline curves derived from vector lines are shown in Figure 4.13.1.3.
4.13.2 Creating and editing curves

Creating smooth curves

The Curve tool generates a smooth NURBS, tangent, or spline curve from an open or closed vector line. To create a curve, with topological level at Object and the Curve tool active, click on an open or closed vector line. The points of the vector line become the control points of the smooth curve, which is drawn on the screen. You can also prepick several vector lines and click in the graphic window to derive several spline curves at the same time. The options for creating smooth curves are located in the Curve Options dialog (Figure 4.13.2.1), which is invoked from the tool.

Figure 4.13.2.1: The Curve Options dialog.

The Curve Options dialog contains three tabs. The first, Options, contains all the options regarding the shape of the curve. The second tab contains the standard Display Resolution options, common to all smooth objects. The third is the standard Status Of Objects options.
The Options tab

Model Type: The two options in this group determine whether the curve will be generated as a Facetted or Smooth type of object.

Nurbs Curve: When this option is on, a NURBS curve is created from the selected vector line.

  Degree: The degree of the NURBS curve is entered in this field. The degree must be equal to or greater than 2 and less than 20. The number of points in the vector line used for the generation of the NURBS must be at least degree+1.

Tangent Curve: When this option is on, a tangent curve is created from the vector line picked.

  Curvature: This slide parameter determines how sharply the curve bends at each tangent point. The higher the value the larger the bend. When first created, the curvature parameter is the same at each tangent point. This can later be changed with the Edit Controls tool.

Spline Curve: When this option is on, a spline curve is created from the selected vector line. Three different variations of splines can be generated, as follows:

  Quadratic: When this option is on, the spline curve is of degree 2. This is the same spline as created directly with the Sketch Spline tool.

  Cubic: When this option is on, the spline curve is of degree 3.

  Through Points: When this option is on, the spline curve passes through the points of the selected vector line. This is similar to the tangent curves, except that there is no control over the curvature. This type of spline curve can also be drawn directly with the Cubic B-Spline tool.

  Break Into Segments: When this option is on, the spline curve is broken into a number of curves joined together at their end points. The Points Per Segment parameter determines how many points are used per curve.

Edit: When this option is on, the Nurbz Curve Edit dialog (Figure 4.13.2.3) will also be invoked when the Curve tool is applied.
The Display Resolution tab

This tab is found in many dialogs. For surfaces and solids, its options determine how densely an object is facetted. For curves, they determine how densely they are segmented. This tab now displays different options for curves, as shown in Figure 4.13.2.4.

Simple: This slide rule offers a quick method for setting the density of the segments of a curve, as for the equivalent tab for the surfaces.

Detailed: When this is on, the display resolution is set using one of the options in this group.

# Of Segments: With this option on, the resolution is specified through the number of segments of a curve, entered in its field.

If the curve is an arc, the number of segments applies to a full circle; an arc will use a proportional number of points, based on the relative size of the arc. For example, if 48 is entered and the arc is 180 degrees, the arc will have 24 segments. Splines use the number specified in the field.

Max Normal Deviation: With this option on, the value entered in its field specifies the maximum angle between two segments of a curve. The smaller this angle, the larger the number of segments in a curve. This option makes the segmentation of an object curvature sensitive. If a curve has many bends, more points are generated. If a curve is flat, less points are generated.

Max Edge Length: When this option is selected, segments with dimensions no larger than the number entered in this field are generated.
Editing smooth curves

The shape and parameters of a curve can be changed with the Edit Controls tool, which is described in more detail in section 4.3.7. Only information relating to the curves generated by the Curve tool is highlighted here.

To edit a NURBS, tangent, or spline curve, with the Edit Controls tool active click on the curve. This places the curve in edit mode and displays its controls. The Edit Controls Options palette also appears on the screen, as shown in Figure 4.13.2.5. This palette controls the editing actions when clicking on control points and lines.

For a NURBS curve, its control segments, points, and weights are displayed. The latter appear as sliders at each control point, as shown in Figure 4.13.2.6.

- To move a control point or a control segment, with the Move icon selected in the Edit Controls Options palette, click on the entity to activate it, than click on it again and move it.
- To change a weight, with the Move icon selected, click on its slider to activate it, then click on its slider point to move it, which alters the weight of the curve at that control point. When more than one weight slider is selected, all the selected weights are set to the same value. When pressing the option (MacOS) or Ctrl+shift (Windows) key while changing the weight, all weights of the curve are set to the same value.
- To insert a control point, with the Insert Point icon selected, click on the curve. This will insert a control point close to the clicked point without changing the shape of the curve. You can also click on the line between two control points to insert a new control point at that location. This will slightly alter the shape of the curve.
- To delete a control point, with the Delete icon selected, click on the point. Keep in mind that a NURBS curve must have at least degree + 1 control points. Thus, if a NURBS curve of degree 2 has 3 control points, no more points can be deleted.
For a tangent curve, its tangency points and sliders that represent curvature at each tangency point are displayed, as shown in Figure 4.13.2.7.

- To **move a tangency point**, with the Move ( últimos) icon selected, click on the point to activate it, then click on it again to move it.
- To **change the curvature** of a tangent curve, with the Move ( últimos) icon on, click on its slider to activate it, then click on the slider point to move it. When more than one tangency slider is selected, all the selected tangency values are set to the same value. When pressing the **option** (MacOS) or **Ctl+shift** (Windows) key while changing the tangency, the tangency of all of the curve are set to the same value.
- To **insert a tangency point**, with the Insert Point ( últimos) icon selected, click on the curve. This will always slightly change the shape of the curve.
- You **delete a tangency point** as for the NURBS. Tangent curves need at least three tangency points, as they are internally represented as NURBS of degree 2.

The spline curve controls and editing are discussed in section 4.2.9.
4.13.3 Reconstructing curves

**Reconstruct Curve**

This new tool is used to change the degree and the number of control points of a curve while maintaining the shape as much as possible. Note that changing the degree and or control points may also change the type of the curve. For example splines are always second or third degree. If you use this tool to change the degree to 5, the new curve will transform to a NURBS curve. This is not really significant except that the controls of the curve may look different when it is edited.

Either the prepick or postpick method can be used with this tool. With the postpick method, with the Reconstruct Curve tool active click on the curve that will be reconstructed. With the prepick method, any number of curves are picked first, then the Reconstruct Curve tool is activated and the mouse is clicked anywhere in the graphics window.

The changes to be made to the reconstructed curve are set in the **Reconstruct Curve Options** dialog, shown in Figure 4.13.3.1, that is invoked directly from the tool.

**Change Degree**: When this option is on, the degree of the curve will be changed.

**New Degree**: The new degree for the curve is entered in this text field.

**Keep Shape**: When this option is on, the curve shape is maintained when increasing the degree by adding more control points. If this option is off, the degree of the curve is changed without changing the number of control points and hence the shape may not be maintained. The curve shape cannot always be maintained when lowering the degree, as often control points need to be removed. An example is shown in Figure 4.13.3.2.

![Figure 4.13.3.1: The Reconstruct Curve Options dialog.](image)

![Figure 4.13.3.2: Changing degree from 2 to 5: (a) original curve; result with Keep Shape (b) on and (c) off.](image)
Change Control Points: When this option is on, the number of control points of the curve will be changed, using one of a few methods available. Examples of this option are shown in Figures 4.13.3.3 and 4.13.3.4.

Relative To Curvature: When this option is on, the curve is reconstructed in a way such that the control points are distributed relative to the curvature of the curve. The reconstructed curve will have more control points in the areas of high curvature and less in flatter regions.

Max Normal Deviation: The value entered in this text field, which is in degrees, determines the density of the control point distribution along the curve. Higher values produce less number of control points and lower values produce a high number of control points.

Number Of Control Points: When this option is selected, the curve is reconstructed with the exact number of control points entered in the text field to the right.

Figure 4.13.3.3: Change Control Points with the Relative To Curvature option: (a) original curve and results with Max Normal Deviation set to (b) 70°, and (c) 15°.

Figure 4.13.3.4: Change Control Points using the Number Of Control Points option: (a) original curve and results with (b) 5 and (c) 30 control points.
**Preview:** When this option is on, which is the default, the *Curve Reconstruct* preview dialog, shown in Figure 4.13.3.5, is invoked as soon as the object to be reconstructed has been picked. This dialog contains a preview window on the left and options under it and to the right. In this dialog, you can change the parameters and observe the effect of the changes.

![Figure 4.13.3.5: The Curve Reconstruct preview dialog.](image)

**Options:** This tab contains the same options found in the *Reconstruct Curve Options* dialog, discussed previously in this section.

**Display Resolution:** This tab shows the standard *Display Resolution* settings for the curve as described in section 4.13.2. Examples are shown in Figure 4.13.3.6.
**Show Original**: When this option is selected, the original curve is shown in the preview window. This option is off by default.

**Show New**: When this option is selected, the reconstructed curve is shown in the preview window. This option is on by default. Note that, when first opening the dialog, the new reconstructed curve is identical to the original curve.

These two options can be used to show the difference between the original and new, if the shape of the curve changed due to the new parameters entered by the user.

**Show Controls**: When this option is selected, the controls for the original or new curve are also drawn.

**Show Deviation**: When this option is selected, the deviation of the new curve from the original curve is calculated. It is shown as a series of lines. Each line is drawn from a point on the original curve to the closest point on the new curve. The collection of lines indicates where the deviation of the new curve is largest.

**Maximum Deviation**: This field shows the length of the longest deviation line.

**# Of Samples Per Control Point**: The value entered in this field determines how many deviation lines are calculated and drawn. The value is multiplied by the # of control points. For example, a curve with 10 control points will have \( 5 \times 10 = 50 \) deviation lines calculated, with the default value 5.

**Reconstruct**: When selected, this button executes the reconstruction using the current options and updates the preview to show the new curve.

**Curve Info**: This group contains non editable alphanumeric fields. They show the current object type, degree, and number of control points of the curve.
Figure 4.13.3.6: A nurbs curve of degree 3 with (a) 20 control points before reconstruction, (b) after reconstructing it to 10 control points, and (c) showing the deviation lines.
4.13.4 Attaching curves

Attach Curve

This operation attaches the curve picked first to the curve picked second. There are different alignment options, which are selected from the Attach Curve Options dialog, shown in Figure 4.13.4.1, that is invoked from this tool.

Either the prepick or postpick method can be used with this tool. With the postpick method, with the Attach Curve tool active you click on the two curves that will be attached. With the prepick method, any number of curves are picked first, followed by the selection of the Attach Curve tool and a mouse click anywhere in the graphics window. The Attach Curve tool will attach the selected curves in pairs.

Stretch Curve: This option aligns only the picked edge of the first curve to the picked edge of the second curve, according to the following alignment method.

Keep Opposite End: This option is used to protect the opposite edge to the edge being attached. When this option is on, extra control points will be inserted to ensure that the opposite edge is not affected by this operation.

Move Curve: This option transforms and aligns the first curve to the second curve by aligning their respective normals. This option is illustrated in Figure 4.13.4.3.

Alignment: This menu offers four different methods for aligning the stretched or moved curves. These options represent increasing levels of smoothness between the curves. The alignment is performed by making adjustments to some of the control points to achieve the desired results as follows:

Positional (C0): The last end control point from one curve is repositioned to the end control point of the second curve. For an example, see Figure 4.13.4.2 (a).

Tangents (G1): In addition to the adjustment for Positional (C0), the control point previous to the end control point of the first curve is repositioned such that it is colinear with the end control point of the second curve and its next control point. For an example, see Figure 4.13.4.2 (b).
Equal Tangents (C1): In addition to the adjustment for Tangents (G1), the same control point is adjusted to be at the same distance from the end control point of the second curve as the next control point is from the end control point of the first curve. In NURBS mathematical terminology, the curves are said to have the same first derivatives. For an example, see Figure 4.13.4.2 (c).

Curvature (C2): In addition to the adjustment for Equal Tangents (C1), the second control point from the end is repositioned to form a concave relationship with the other control points. In NURBS mathematical terminology, the curves are said to have the same first and second derivatives. For an example, see Figure 4.13.4.2 (d).

Merge: When this option is on, in addition to being aligned, the two curves are merged into one curve.

Figure 4.13.4.2: Stretch Curve from p1 to p2 with Alignment of (a) Positional (C0), (b) Tangents (G1), (c) Equal Tangents (C1), and (d) Curvature (C2).

Figure 4.13.4.3: Move Curve (a) from p1 to p2 and (b) from p3 to p4.
4.13.5 Blending curves

This operation creates a blend curve between two different curves or between two edges of the same curve.

Either the prepick or postpick method can be used with this tool. With the postpick method, with the Blend Curve tool active, you pick the two curves to be blended. With the prepick method, any number of curves are picked first, then the Blend Curve tool is activated and the mouse is clicked anywhere in the graphics window. The Blend Curve tool will blend the selected curves in pairs.

The Blend Curve tool is affected by a few options, which are selected from the Blend Curve Options dialog (Figure 4.13.5.1), invoked directly from the tool.

**Alignment, Merge**: These options are as for the Attach Curve tool.

**Degree**: This group determines the degree for the depth (V) direction of the new curve.

**From Curve**: When this option is on, the degree of the new curve is the same as the degree of the picked curves. If the picked curves have different degrees, the highest degree is used.

**New Degree**: When this option is on, the degree is determined by the value in the text field to the right of the option. The default is 5.

Examples of blending curves are shown in Figure 4.13.5.2.

Figure 4.13.5.1: The Blend Curve Options dialog.

Figure 4.13.5.2: Blending curve from \( p_1 \) to \( p_2 \) with **Alignment**: (a) Positional (C0), (b) Tangents (G1), (c) Equal Tangents (C1), and (d) Curvature (C2).
4.13.6 Merging curves

**Merge Curve**

This operation merges two curves that touch at a point into a single new curve. The operation automatically recognizes the touching points. The operation is refused when no touching points are found. If two pairs of touching points are found, the new curve will be derived by merging both pairs of points resulting in a closed shape. The merge options are selected from the **Merge Nurbz Options** dialog, shown in Figure 4.13.6.1, that is invoked directly from the tool.

Either the prepick or postpick method can be used with this tool. With the postpick method, the Merge Curve tool is activated first and then a curve is picked. With the prepick method, any number of curves are picked first, the Merge Curve tool is activated next, and then the mouse is clicked anywhere in the graphics window. Examples are shown in Figure 4.13.6.2.

**Align Curves**: This option aligns the first curve to the second curve at the merged point(s), according to the following alignment method.

**Alignment**: This menu offers four different methods for aligning the merged curves. These options are as for the Attach Curve tool. Note that the **Positional** alignment does not change the shape of the curves. Examples are shown in Figure 4.13.6.3.
**Smooth Curves**: This option smooths the merged point by manipulating the knots at the point rather than the adjacent control points as with the **Align Curves** option.

**Smoothness**: This slider controls the amount of smoothing applied with 0% representing no smoothing (hard edge) to 100% which is completely smooth. Note that the location of the adjacent points to the merged point affects how smooth the results will be. If the adjacent control points are very close, the smoothing effect will be minimal even at the 100% smooth setting. Examples are shown in Figure 4.13.6.4.
4.13.7 Extending curves

Extend Curve

This operation extends a curve either to a clicked point or a predetermined distance. The extension is applied at the end of the curve nearest to the click point that selects the curve. The method of the extension and other options are selected from the Extend Curve Options dialog, shown in Figure 4.13.7.1, that is invoked directly from the tool.

Through Point: When this option is on, the curve is extended to the click point either by constructing additional spline points or by constructing an arc.

With the prepick method, the curve(s) to be extended are picked first, then the Extend Curve tool is activated, the extended curve is shown interactively extended to the cursor location. A final click ends the interactive drawing and completes the operation (p2). When multiple curves are prepicked, they are all extended to the same point.

When the postpick method is used, with the Extend Curve tool active you click on the curve to pick it (p1) and the extended curve is shown interactively extended to the cursor location. A final click ends the interactive drawing and completes the operation (p2). The operation is illustrated in Figure 4.13.7.2.

Construct Spline: With this option on, the curve is extended by constructing a spline from the end of the curve through the click point.

Construct Arc: With this option, the curve is extended by constructing an arc from the end of the curve through the click point.

By Distance: When this option is on, the curve is extended by a preset distance or through interactive input.
**Preset**: When this option is selected, the curve is extended the specified distance. When the distance entered is positive, the curve is extended along the tangent at the end of the curve. When the distance entered is negative, the curve is trimmed back the specified distance from the selected end of the curve. With the postpick method, with the Extend Curve tool active you pick the curve. With the prepick method, you pick any number of curves, you then activate the Extend Curve tool and you click anywhere in the window. Examples are shown in Figure 4.13.7.3.

**Interactive**: When this option is selected, the curve is interactively extended along the tangent at the end of the curve. With the postpick method, with the Extend Curve tool active you pick the curve. With the prepick method, you pick any number of curves, you then activate the Extend Curve tool and you click anywhere in the window. The extended curve is then interactively shown as the mouse is moved. The extension length is calculated from the distance the mouse is moved from the click point. A final click completes the extension and ends the interactive input.

**Merge**: When this option is on, the curve extension is merged with the original curve. With **Construct Spline** and **By Distance**, the extension is merged to form a single curve. With **Construct Arc**, if the original curve is a spline, the arc is added to it to form a composite curve. If the original curve is not a spline, the arc is constructed as a NURBS curve and merged to the original curve. The **Merge** option has no effect when using the **Preset** option with a negative number.

![Figure 4.13.7.3](image)

**Figure 4.13.7.3**: Extending curves using **Preset**: (a) Original, (b) positive, and (c) negative distance.
4.13.8 Breaking curves

**Break Curve**

This operation breaks a curve at the click point. The resulting curves will have the same degree as the original curve. Breaking a closed shape results in an open shape at the split point. This tool has no options and invokes no dialog.

Either the prepick or post pick method can be used with this tool. With the postpick method, the Break Curve tool is selected first and then a curve is picked. With the prepick method, any number of curves are picked first, followed by the selection of the Break Curve tool and a mouse click anywhere in the graphics window. Note that the location of the click point to prepick the curve(s) is significant as it determines the break point(s).

![Figure 4.13.8.1: Breaking (1) an open and (2) a closed curve. (a) Original curves, (b) the click points, and (c) the broken curves (where one piece of the curve has been moved.)](image)
4.13.9 Curved lines

A variety of curve lines or splines can be generated from previously drawn open or closed vector lines that are used as controlled lines. These splines can be created either as parametric or plain entities.

The C-Curve tool allows you to generate curved lines from straight control lines. These are called controlled curves (or c-curves) and are distinguished from plain curves (or simply curves) in that they are parametric entities and their internal representation carries the controls that generated them. This allows you to go back and edit them to reshape their geometry or to change their parametric values. The C-Curve tool is also used to edit c-curves after their initial generation.

To derive a c-curve, with topological level set to Object and the C-Curve tool active you click on a vector line. To edit a c-curve after its initial generation, with the C-Curve tool active you click on the c-curve. In both cases, if Edit is on in the C-Curve Options dialog, the program enters c-curve edit mode, in which you can interactively manipulate the shape of the c-curve. The C-Curve Options dialog (Figure 4.13.9.1) can be invoked from the C-Curve tool in the usual manner by double clicking on the tool icon. It contains a variety of options that affect how the C-Curve tool will be executed and the parametric values that will be applied.

Construct Plain Curve: The C-Curve tool will generate a c-curve unless this option is on, which will result in a plain curve. By default this is off. A plain curve can be edited upon creation (if the Edit option is selected). It cannot be edited and its shape cannot be changed after its initial generation, other than by using the general geometric transformations that apply to all objects.

Adjust To New Parameters: When a curve is generated for the first time, the system uses the current settings of the C-Curve Options dialog. When the C-Curve tool is subsequently used to regenerate the c-curve (with or without editing) the system will preserve the original parameters of the curve if this option is off. If it is on (which is the default), the curve will be regenerated using the current parameters in the dialog.

Figure 4.13.9.1: The C-Curve Options dialog.
Construct Directly: When this option is on and a c-curve is generated for the first time or is regenerated, it is constructed directly, rather than entering edit mode. Note that regenerating a curve directly without adjusting to new parameters has no effect.

Edit: When this option is on, which is the default, as soon as the C-Curve operation is executed, the system enters the c-curve edit mode, under which you can manipulate the geometry (shape) of your curve. You recognize that you are in c-curve edit mode by that the tool palette is grayed out and becomes inactive. All the window tools remain active and you can use them as you manipulate the shape of your curve. You terminate the c-curve edit mode by double clicking the mouse anywhere in the graphics window.

Typically, under the edit mode, all the controls of a curve will be shown and will be rubber banded, so that you may manipulate them. The curve itself is also rubber banded for the lower but not for the upper level curves. For the lower end curves, you can also select the Hide Controls option which causes the system to show the curve only. You can still execute quite a few editing operations, even with the controls hidden. Hiding the controls usually implies that you wish to shape your curve by simple visual inspection of the effects of your manipulations.

Snap: Moving one or more control points on top of another (making them coincident) has major implications for the curve generation process. Such points typically break the continuity of a curve and result in straight lines or corner points. When this option is on, positioning of points on other points becomes easier. The snapping tolerance is the same with that used for object snaps and is set in the Object Snaps dialog, accessible from any of the snaps in the window tool bar.

The curves

form•Z offers a complete range of methods for the generation of curves, aimed at addressing the needs of any user. The desired method can be selected from the C-Curve Options dialog. All the methods are illustrated in Figure 4.13.9.2, for both closed and open curves. The curve types are organized in two groups or levels. At the upper level are the NURBS (Non Uniform Rational B-Spline), the B-Spline, and the Bezier. At the lower level are cubic and quadratic Bezier, and three variations of tangent curves. In general, the higher the level of a curve the more control a user has over the shaping of a curve, and the more computationally expensive (slower) the generation process is. The quick curves in the lower group can be used in a more intuitive manner to create curved lines mostly through visual inspection. The quick curves are similar to those offered by painting/drawing programs currently on the market and are expected to be the most frequently used. Consequently, Quick Quadratic has been made the default. The remainder of this subsection summarizes the mathematical features of the curves. You may or may not read it. You can still use the curves without that information and by simply experiencing how they behave and are affected by the different options.
All curves are derived from control lines that are drawn first and play a major role in determining the final shape of a curve. A control line is a connected sequence of points which can be open or closed. In form•Z, the control lines are surface objects and are not required to be planar. Different mathematical formulas (curve types) may be applied to the same control line to generate different curves. The upper level curves may execute for different degrees. A degree is the high exponent in the formula that generates a curve. The higher the degree the more computation is required. The degree is set in the Degree field of the C-Curve Options dialog. The dialog also contains the # Of Points field which determines how many points a curve will be generated with. The degrees of the lower level curves are fixed to 2nd and 3rd. Their density is controlled by the value entered in the Smooth Interval field.

Figure 4.13.9.2: (1) Third degree NURBS with variable weights. (2) Third degree Spline. (3) Bezier. (4) Broken Bezier (2nd degree). (5) Continuous Bezier (2nd Degree). (6), (7) Third and second degree Quick curve. (8), (9), (10) Shallow, normal, and deep tangent.
Upper level curves:

- **NURBS** (Non Uniform Rational B-Spline): The NURBS is a mathematical construction which, while not pure from a strictly mathematical point of view, offers the utmost flexibility and control when generating curves. It also allows what is known as “local” control. That is, the shape of a portion of a curve can be changed without affecting the other parts of the curve, a typical side effect with many mathematical formulas.

The NURBS can be of any degree and uses two sets of controls: the **weights** and the **knots**. The weights are positive numeric values and there is one weight for each point in the control line. The weights are interpreted relative to each other and their practical significance is that they determine how close to the control point the local apex of a curve is placed. The knots are positive numeric values which affect the distribution and local density of points in a curve. There are always more knots than control points. How many more depends on the degree of the curve \((\text{number of knots required} = \text{number of control points} + \text{degree} + 1)\).

- **B-Splines**: A B-Spline is a NURBS with equal weights. It can be of any degree and it is generated on the basis of control points and knots.

- **Bezier**: A Bezier is a B-Spline with equal knots (in addition to the equal weights). It can be of any degree, however, there is a required correlation between its degree and the number of control points \((\text{degree} = \text{number of control points} - 1)\). By definition, a Bezier always starts and ends at the endpoints of a control line. Consequently, the Bezier does not work well with closed control lines, whenever a continuous curve is desired. It can be useful when a break point is desirable. Bezier curves will appear continuous when the first and the last segments of a closed control line are colinear.

Given the required correlation between the degree and the number of control points, when a Bezier is generated in form•Z, the current degree value is ignored and the program automatically calculates a degree after counting the control points. The next two variations of the Bezier (broken and continuous) do take the current degree value into consideration.

- **Broken Bezier**: This is a composite Bezier consisting of a sequence of Bezier curves. Depending on the degree currently selected, the control line is broken to parts the size of which (number of points) is determined by the degree and a Bezier curve is generated for each part. If the last portion of the control line has less than the required number of points, the last point is repeated the required number of times. The broken Bezier is characterized by break (corner) points where the parts meet, which is frequently quite useful for the generation of certain types of shapes (see examples in Figure 4.13.9.7).

- **Continuous Bezier**: This is a variation of the broken Bezier which breaks the complete control line to portions that start and end on midpoints of segments (except for the first and last segments of an open control line). This results in a continuously smooth curve.
Lower level curves:

- **Cubic**: This is a 3rd degree continuous Bezier. That is, the complete control line is subdivided to portions that start and end at the midpoint of every other segment and a Bezier is generated on the basis of the three resulting segments (or four points). While the initial generation of this curve, by default, places the endpoints of the control portions on midpoints of segments, when the curve is edited these points can be moved anywhere between the ends of the segment on which they are located. For open control lines, this curve begins and ends on the first and last point, respectively.

- **Quadratic**: This is a 2nd degree continuous Bezier. The control line is subdivided into portions that start and end on midpoints of consecutive segments and a Bezier is generated for each segment. Everything else is as with the quick cubic curves.

- **Shallow, Normal, and Deep Tangent**: These curves are quick quadratic curves (2nd degree continuous Bezier) that are generated on the tangent vector of the original control line. That is, given a control line, a tangent vector line is generated first and is then used as the control line for the generation of a 2nd degree continuous Bezier. The tangent vector consists of segments which are perpendicular to the bisectors of the angles at the control points and of segments parallel to the segments of the original control line. The distance at which the parallel lines are generated controls the curvature and is the base for the distinction between the three variations included in form-Z: shallow, normal, and deep.

While all the other types produce curves that are tangent to all or some of the segments of the control line, the tangents produce curves which pass through the control points. This makes them useful in application areas where required. For example, for the rounding of contour vectors.

**Curve edit mode**

The system enters the curve edit mode when the **Edit** option is selected in the C-Curve Options dialog and the C-Curve operator is executed either to generate a curve for the first time (by picking a control line) or to regenerate a curve (by picking a previously generated c-curve). Assuming that the **Hide Controls** option is not selected, all the controls that apply to the type of curve being generated are shown in rubber banded mode, together with the curve itself. The curve is shown in the color assigned to it. The highlight color (default red) is used for the control line and some of the other controls.

Of the upper level curves, the NURBS is the only curve where all the controls are present, as illustrated in Figure 4.13.9.3. The B-Spline appears with knots but no weights, and the Bezier appears with neither knots nor weights. All types of upper level curves appear with their complete control lines.
The weights are shown as lines drawn at the control points in directions that coincide with the bisectors of the angles of the points. The length of these lines represent the magnitude of the weights. The end of the weight line can be moved to increase or decrease its length, which changes the magnitude of the corresponding weight value.

The common knots are shown as filled diamonds. A larger filled diamond is used for the last knot of an open control line to indicate that this knot cannot be moved. Larger white diamonds are used for knots in the neighborhood of the closing point of closed lines to indicate that these knots, while they can be moved, they do not move independently. When moved they also cause some other (white) knots to be moved.

Of the lower level curves, the Quick curves are shown with their control lines, their control points (white diamonds), and their knots (filled diamonds), as illustrated in Figure 4.13.9.4. Only those segments of the control lines that contain knots are shown. That is, every other control segment of the cubic and all the segments of the quadratic is shown. The knots of the quick curves are the points where the parts to which the control line is broken meet. That is, the knots of the quick curves are entities different from the knots of the upper level curves. The tangent curves are shown with their complete control lines and points.

Editing operations are applied by moving the mouse cursor onto a control, clicking, repositioning the mouse cursor, and clicking again. As the mouse is moved from one control to another, the cursor changes to indicate what is currently selected. What operation is applied is determined by the icon of the mouse at the time the mouse is clicked. In most instances keys may be pressed to switch to another cursor and hence to another operation. The available operations, the corresponding cursors, and the type of curve to which they apply are summarized in the table of Figure 4.13.9.5.
<table>
<thead>
<tr>
<th>Operation</th>
<th>NURBS</th>
<th>B-Spline</th>
<th>Bezier</th>
<th>Quick</th>
<th>Tangent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move point</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Move segment</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Move through curve</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Move knot</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move c-curve</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Change weight</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R-Move point</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>R-Move segment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>R-Move curve</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Insert on point</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Insert on segment</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Delete point</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Change point type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

*Figure 4.13.9.5:* Summary of the curve edit operations, their cursor icons, and the types of curves to which they apply.

To **move a point** the mouse is clicked when the cursor is close to a control point and the cursor shows ☐. After the first click, the affected controls are rubber banded and follow the motion of the mouse, which is relative to the currently active reference plane, as with the geometric transformations. The Perpendicular Switch (window palette) may be turned on to move the point in a direction perpendicular to the reference plane. When the point is at the desired position, the mouse is clicked again to complete the operation. With the upper level curves only the controls are rubber banded during the motion of the mouse and the curve is regenerated after the second click. With the lower end curves, both the controls and the curve are rubber banded and the curve is regenerated continuously. This difference in the editing behavior is due to different computational demands of the two levels of curves. The move point operation applies to all types of curves.

To **move a segment** the mouse is clicked when the cursor is close to a control segment and the cursor shows ☐. Everything else is the same as the move point operation.
The **move curve** operation can be applied when the mouse cursor is close to the curve and its icon is \(^{\text{m}}\). This operation is only available with the quick and tangent curves and is particularly useful when the **Hide Controls** option is selected.

To **move a knot** the mouse is clicked when the cursor is close to a knot and the cursor shows \(^{\text{m}}\). This operation applies to the NURBS, B-Splines, and Quick curves. However, the behavior of the knots of the latter is somehow different than that of the former two. The movement of knots is always constrained to the control lines.

The knots of the upper level curves can move anywhere between their previous and next knots. They are not restricted by the points of the control lines and they may move from one segment to another. The last knot of an open control line (which always coincides with the last control point) cannot be moved. The white knots of closed control lines (the knots in the vicinity of the closing point of the control line) can be moved but their motion also affects other white knots. The knots of the quick curves can only be moved between the end points of the segments on which they are and cannot cross over to other segments. For open quick curves, the first and the last knots (which coincide with the first and last control points) cannot be moved.

To **move an entire control line**, the shift key is pressed when the cursor is close to any part of the line, which causes it to change to \(^{\text{M}}\). Click, move the cursor, and click again. The entire c-curve and all of its controls move.

The **change weight** operation essentially consists of moving the endpoint of the weight line to make it longer or shorter. It is executed by placing the cursor on the endpoint of the weight line (which causes the cursor icon to change to \(^{\text{a}}\)) and clicking the mouse. This motion is constrained by the direction of the bisector of the angle of the point to which the weight line is attached. This operation can only be applied to a NURBS.

**Rotational move** operations (r-move) can be applied to points, segments, or curves of quick curves only. They can be executed by pressing the control key (Macintosh) or ctrl+alt (Windows), while placing the cursor close to a point, segment, or curve, and the cursor icon shows \(^{\text{a}}\), \(^{\text{b}}\), or \(^{\text{c}}\), respectively. When r-moving a point, it rotates around the knot next to it and this motion also affects the point at the other end of the segment on which the knot lies. The point can also be moved in the direction of its segment without rotating, or the two motions may be combined.

To **insert a point on a point** option (Macintosh) or ctrl+shift (Windows) are pressed and the mouse is clicked when the mouse is close to a control point and the cursor shows \(^{\text{d}}\). To **insert a point on a segment** the option key is pressed and the mouse is clicked when the cursor is close to a control segment and the cursor shows \(^{\text{e}}\). To **delete a point** the shift key is pressed and the mouse is clicked when the cursor is close to a control point and the cursor shows \(^{\text{f}}\). The insertion and deletion operations can be applied to all types of curves.
Inserting a point on a point or on a segment, or moving one or more points onto another point affects the continuity of a curve and produces corners or straight line segments. These manipulations can be executed on all types of curves. For the quick curves a special operation is also available for changing the type of a point. It is executed by pressing the command (Macintosh) or ctrl (Windows) key, positioning the mouse on the knot point whose type is to be changed, and clicking the mouse while the cursor icon shows ☒. This is the only curve editing operation that is executed with a single mouse click, which causes the Point Type dialog, shown in Figure 4.13.9.6, to appear.

The dialog contains the four types of points available and the one that corresponds to the point selected is shown checked. To change the type of point the desired type is selected and the OK button is clicked.

A Corner point corresponds to three coincident points and produces a corner break. The Flat-Curve and Curve-Flat points correspond to two coincident points and produce a straight line to their left and right, respectively. A Curve point corresponds to a single point and produces a curve on both sides. Straight line portions are produced between a pair of a curve-flat and a flat-curve points, or a pair of corner points, or some other combination of these types.

Examples

The same control lines produce significantly different results when different types of curves are generated and/or different parameters are used. These differences are illustrated by the examples in this section.

In Figure 4.13.9.7, the same control lines are used with different curves and different degrees to produce different “daisy” shapes. In (a) a broken Bezier of degree 3 and in (b) of degree 6 was used. The shape in (c) was generated by a 2nd degree continuous Bezier and a normal tangent was used for (d).
The example in Figure 4.13.9.8 shows the effects of different degrees on the same type of B-Spline, when applied to the same control line. In general, the higher the degree the further away from the control line the curve is.

Figure 4.13.9.9 illustrates the effects of different weights. A NURBS has been generated eight times on copies of the same control line, each time with different weight values for the two points marked A (increasing top to bottom). The weight of B remained constant. The higher the weight value the closer to the corresponding control point the curve is pulled.

Curved lines can be generated anywhere in 3D space and they do not need to be planar. The control line used in Figure 4.13.9.10 was first drawn on a flat plane and its points were then moved. Copies were then used to generate a quick quadratic and a 4th degree NURBS.

The control line in Figure 4.13.9.11 was drawn directly in 3D by snapping to the segments of a cuboid. Then the control line was used to create a quadratic, resulting in a spiral shape.

In both cases the option **Allow Intersecting Lines** was turned on in the **Vector Line Options** dialog (see section 4.2.4). With this option off these lines would have not been allowed by the system.
4.13.10 Formula curve

The Formula Curve tool constructs a parametric curve that is defined by a mathematical formula.

Formula Curve

Formula curves are generated on the basis of mathematical formulas that are either pre defined or are entered by the user. This tool is useful when a curve needs to follow specific parameters. For example, a helix can be defined using a formula.

Formulas for curves are specified by three parametric functions: \(X(u), Y(u),\) and \(Z(u),\) where \(u\) is defined in the domain \([U_{\text{min}} < u < U_{\text{max}}]\). These functions must map one dimensional parameter space \((u)\) to three dimensional object space \((x, y, z)\). The minimum and maximum values specify edge boundaries in parameter space.

A function is unbounded at its vertical asymptotes. The Formula Curve tool requires that the lower and upper limits of the domain not be the function’s vertical asymptotes. For instance, function \(T=\tan(u)\) has vertical asymptotes at values \(\pi/2, \pi, 3\pi/2,\) and so on.

The options that affect the generation of formula curves are selected from the Formula Curve Options dialog, shown in Figure 4.13.10.1, that is invoked directly from the tool. This dialog consists of two tabs: Options and Display Resolution. The first tab contains the options that affect the generation of the formula curves. The second tab contains the standard parameters that control how the wires are displayed on the screen.

Parametric Functions: The values entered in this group of fields specify the three parametric functions \(X(u), Y(u),\) and \(Z(u).\) The user may either choose from a pre-defined set of functions or type in new functions.
**Domain**: The values in these fields determine the domain for parameter $u$.

**Umin, Umax**: The values entered in these fields specify the lower and upper limits for $u$.

**Unit Size (Scale)**: This value determines the actual size of the curve to be created. The formula curve is initially created in the normalized unit space and is then scaled by this value.

**Pre Defined**: This pop up menu contains the names of preset mathematical formulas, namely, **Helix**, **Spiral**, **Sine**, and **Cosine**. Selecting one of these items enters the respective formula in the $X(u)$, $Y(u)$, and $Z(u)$ fields. The user can then modify these fields or enter a new function. Whatever is in these field will be executed next time the Formula Curve tool is applied.

**Recent**: This popup menu stores up to 10 most recently executed formulas, from where they can be selected again to be re-executed.

![Figure 4.13.10.2](image)

**Figure 4.13.10.2**: Examples of formula curves using the predefined functions: (a) **Helix**, (b) **Spiral**, (c) **Sine**, and (d) **Cosine**.

![Figure 4.13.10.3](image)

**Figure 4.13.10.3**: Example of formula curve using a custom function: $X(u)=u\cdot\cos(u)$, $Y(u)=u\cdot\sin(u)$, $Z(u)=2\cdot\cos(5u)\cdot2^{-u}$, and domain $U_{\text{min}}=0$, $U_{\text{max}}=5$. (a) Top view, (b) front view, and (c) 3D view.
Previewing and editing formula curves

When the Formula Curve tool is executed with the Preview option on in the Formula Curve Options dialog, the Formula Curve Edit dialog, shown in Figure 4.13.10.4, is invoked. This dialog makes it possible to preview a formula curve and to experiment with different variations. This dialog contains a standard preview window on the left, the options found in the Formula Curve Options dialog, and a few additional options, as follows:

**Update**: Clicking on this button applies the new parameters and refreshes the preview window. This button is dimmed when the parameters of the preview are up to date.

**Automatic**: When this option is on, the preview is automatically updated every time a parameter is changed. When this option is on, the Update button is dimmed.

![Figure 4.13.10.4: The Formula Curve Edit dialog.](image)
4.14 Nurbz objects and c-meshes

In form-Z, an object is called a *nurbz* when its surfaces are NURBS surfaces. There are a number of tools generating and manipulating such surfaces, which are discussed in this section. They are:

- Nurbz
- Nurbz Cross Sections
- Reconstruct Nurbz
- Extract Curve
- Attach Nurbz
- Blend Nurbz
- Merge Nurbz
- Extend Nurbz
- Split Nurbz
- Untrim Nurbz
- Curve On Nurbz
- C-Mesh
- Formula Surface
4.14.1 Creating nurbz objects

The Nurbz Options dialog, shown in Figure 4.14.1.1, which is invoked directly from this tool, has been rearranged and it now consists of three tabs: Options, Display Resolution, and Status Of Objects. The first tab contains the options that affect the generation of nurbz objects. The second tab contains the common parameters that dictate how the faceted version of a nurbz is displayed on the screen. The third tab contains the options that determine how the operands picked for the operation are treated.

Surface Construction: This group of items determines how the surface is constructed.

By Loose Lofting: When this method is selected, nurbz surfaces are generated from any number of open or closed control lines that are connected (lofted) in a given order (Figure 4.14.1.2). When this method is used, the length degree must be greater than 0 and equal or less than the number of segments in the control lines. The depth degree must again be greater than 0 and less than the number of the control lines selected. The default degree is 3 for both directions. Note that a degree of 1 generates straight segments and no curvature.

When creating a nurbz object by lofting lines, either the prepick or postpick selection method can be used. How many control lines are picked depends on the depth degree set in the Nurbz Options dialog. This number is depth degree + 1. While this is exactly the number of control lines that are picked with the postpick method, it is the minimum number required when using the prepick method.

When using the prepick method, with the Pick tool select degree+1 or more lines to be lofted for the generation of the nurbz object, in the order they will be used. Next with the Nurbz tool click anywhere in the window. With the postpick method, with the Nurbz tool select exactly degree+1 lines. As soon as you select the last line, the nurbz object is generated and, depending on the options currently set in the Wire Frame Options dialog, its iso lines may be displayed together with its facets or by themselves. Examples of nurbz objects derived from closed and open control lines are shown in Figure 4.14.1.2.
By Tight Lofting: When this method is selected, nurbs surfaces are generated from any number of control lines that are all closed or all open. These lines are picked as for the Loose Lofting variation. The resulting surface passes through all the control lines. Other options are the same as for loose lofting. Either prepick or postpick can be used, as with the By Loose Lofting method. For examples, see Figure 4.14.1.3.

By Boundary Curves: When this method is selected, a nurbs surface is constructed to fill the area between two, three, or four open curves that delineate the boundary of a surface. The boundary curves can be any type of spline, c-curve, or even simple vector lines. Examples are shown in Figures 4.14.1.4, 4.14.1.5, and 4.14.1.6.
Figure 4.14.1.5: A nurbz surface created **By Boundary Curves** and (a) the four splines shown. (b) The resulting wavy nurbz surface.

Figure 4.14.1.6: Modeling a nurbz handle **By Boundary Curves**: (a) four splines (b) generate a nurbz surface, (c) which is next copy-mirrored to complete the handle.
For the **By Boundary Curves** method, adjacent curves can touch end to end, but this is not an absolute requirement. If the curves intersect, the surface corner is established at the curve intersection. If the curves have multiple intersections, the portion of the curve where the pick point lies is used as the surface boundary. If the curves do not intersect or touch, the corner of the surface is positioned at the average of the curve endpoints. Note that, for curves with very few points, this can have a significant effect on the shape of the boundary of the surface. Figure 4.14.1.7 illustrates different combinations of intersecting and non-intersecting curves. The most predictable results are achieved when using boundary curves that are touching or intersecting and are convex in shape.

When using the **By Boundary Curves** method, either the prepick or postpick method can be used. With the prepick method, use the Pick tool to select the desired two, three, or four curves, select the Nurbz tool, and click anywhere in the window. The surface is created right away. With the postpick method only four curves can be selected. Select the Nurbz tool and, with it, click on the four curves. The surface is created right after the fourth click. Note that the curves do not need to be picked in any particular order and that the directions of the curves do not affect the resulting surface.

**By U/V Curves**: When this method is selected, a nurbz surface is constructed from a network of curves. The curves do not have to intersect each other. All curves must be open. Examples are shown in Figure 4.14.1.8.

**Accuracy**: This slider is only available to the **By U/V Curves** method. As this method interpolates all the curves in the network, it has a tendency to produce surfaces with a huge number of control points. Lowering the accuracy reduces the number of control points. However, when it does, the surface may not pass through all the curves.

---

**Figure 4.14.1.7**: Nurbz **By Boundary Curves** from (a) intersecting, (b) non-intersecting, and (c) mixed intersecting and non-intersecting curves.

**Figure 4.14.1.8**: Nurbz created **By U/V Curves** and 3x3 open control lines.
**By Points**: When this method is on, a nurbz surface is created from a grid of points. The surface passes through all the points of the grid. For examples, see Figure 4.14.1.9.

**Number In Length (U), Number In Depth (V)**: Determine the number of curves in the length/depth direction for the **By U/V Curves** option and the number of points in the length/depth direction for the **By Points** method. These options are dimmed when other construction methods are active.

**Adjust Direction**: Inconsistent directions of the control lines used by the **Lofting** operations result in twisted nurbz surfaces. Turning this option on corrects the inconsistent directions and eliminates the twist, as shown in Figure 4.14.1.10. This option is dimmed when operations other than the two **Lofting** operations are selected.

**Degree**: This group contains options for controlling the degree of the new surface.

**From Source**: The degree of the nurbz surface generated will be that of the curves used to create it. If the curves are of various degrees, the highest degree will be used. This option is dimmed when the **By Points** construction method is used.

**Length (U)**: These parameters determine the U degree of the nurbz surface generated. This item is dimmed if the **From Source** option is selected and the construction method is not **By Points**.

**Depth (U)**: These parameters determine the V degree of the nurbz surface generated. This item is dimmed if the **From Source** option is selected and the construction method is **By Boundary Curves** or **By U/V Curves**.

**Closed In Length (U)**: The nurbz surface generated will be closed in the U direction.

**Closed In Depth (V)**: The nurbz surface generated will be closed in the V direction. An example is shown in Figure 4.14.1.11.

---

*Figure 4.14.1.9*: Nurbz constructed **By Points**:
(a) 10x13 points and (b) the resulting nurbz.

*Figure 4.14.1.10*: Nurbz **By Tight Lofting** with **Adjust Direction** (a) off and (b) on.

*Figure 4.14.1.11*: Nurbz objects with the **Closed In Depth (U)** option (a) off and (b) on. Shown in (1) wire frame and (2) hidden line.
**Caps:** This group determines the capping of a nurbs surface derived from closed control lines. This group is dimmed when the construction method is **By Boundary Curves** or **By U/U Curves**.

**Start, End:** When this option is on, the start/end of the surface is capped. The text field to the right determines the extent of the cap. A cap of 100% will close the start/end of the surface completely and a cap of 0% will leave it completely open. Examples are shown in Figure 4.14.1.12.

**Rounding:** This group of options determines whether the ends of a nurbs derived from closed control lines may be rounded. This group is dimmed when the construction method is **By Boundary Curves** or **By U/U Curves**.

**Start, End:** When these options are on, the start/end of the nurbs is rounded. The text fields next to the options determine the extent of the rounding. A rounding of 100% closes the start/end of the object completely. A rounding of 0% leaves it completely open. Examples are shown in Figure 4.14.1.13.

Figure 4.14.1.14 shows examples for both **Cap** and **Rounding**.
**Maintain Shape:** When applying the Nurbz tool to an existing nurbz and the degree parameters are increased (which would normally affect the shape of the object), selection of this option preserves its shape. However, to do this, it needs to generate new control points that fit the existing shape. When this option is off, the previous control points are maintained, but the shape of the object changes. It is not possible to maintain both the same shape and the same control points. This option is illustrated in Figure 4.14.1.15.

**Edit:** When this option is on and a new nurbz is created, a Nurbz By ... Edit dialog, such as that shown in Figure 4.14.1.16 is invoked. The title of the dialog reflects the specific operation used to generate a nurbz. When the Nurbz tool is used to edit a nurbz after its initial creation, the Nurbz Edit dialog, shown in Figure 4.14.1.17, is invoked. Both dialogs work as all the edit dialogs and allow you to preview a shape as its parameters are changed.

**Adjust To New Parameters:** When this option is on, and the Nurbz tool is clicked on a previously generated nurbz object, it is regenerated according to the parameters currently set in the dialog. This option can be used with or without the Edit option. Note however, that both of them cannot be off at the same time.
Nurbz from analytic primitives

The Nurbz tool can be used to convert an analytic primitive to a nurbs object. To do so, with the Nurbz tool active, click on the object. It is converted to a nurbs, but it maintains its shape.

When generating a nurbs from an analytic primitive, formZ automatically decides what degrees will be used for length and depth, to maintain the shape of the primitive. In other words the Degree parameters in the Nurbz Options dialogs are ignored. However, different degrees can be applied after the initial generation, by editing the nurbs object.

The Edit option in the Nurbz Options dialog is taken into consideration. If on, the preview dialog is invoked as soon as you click on an analytic primitive. If off, the object is converted immediately without any previewing. How the set of analytic primitives converts to nurbs objects is shown in Figure 4.14.1.18.

Nurbz objects from derivative objects

The Nurbz tool can be used to convert objects of revolution, sweep objects, skins, and helixes to nurbs objects. To do so, with the Nurbz tool active, click on the object. If the object already uses its nurbs option, the shape is maintained. Otherwise a nurbs shape is generated from the faces which will smooth the original object. If a faceted sweep object is converted to nurbs, the Length (U) and Depth (U) degrees are applied. For a faceted object of revolution or helix, only Length (U) is used.
4.14.2 Editing nurbz

After their initial generation, nurbz objects can be edited. Editing can be of two different types: geometric editing, which moves the control points and reshapes the object, and parametric editing, which changes the parameters of the nurbz object. Each of these editing types can be applied with more than one tool, as follows:

- Parametric editing can be applied using the Nurbz tool or the Query tool. Both invoke the Nurbz Edit dialog (Figure 4.14.1.17), as discussed in the previous section, from which the nurbz parameters can be changed. Editing can also occur at the end of the generation process, when the Edit option is on in the Nurbz Options dialog. The latter occurs through the Nurbz By ... Edit dialog that is invoked.

- Geometric editing can be applied using the Edit Controls or the Edit Surface tools, as discussed in the remainder of this section.

![Edit Controls](image)

Clicking on a nurbz object with the Edit Controls tool sets the object in controls edit mode and shows the controls. These are the control lines and points that were used for the initial generation of the nurbz object or those automatically generated by the program when an analytic primitive or a derivative object is transformed into a nurbz object (Figure 4.14.2.1).

When in controls edit mode, clicking on a control point while control (MacOS) or ctrl+alt (Windows) is pressed picks the control as well as its related controls. Clicking on a point, the adjacent control points are also picked. Clicking on an edge, the sequence of edges in the same direction are also picked.

The Edit Controls Options palette appears on the screen as shown in Figure 4.14.2.2. This palette controls the editing actions when clicking on control points and lines. It contains two sections: The Action section determines the operation that is applied when clicking on selected control points and lines. The type of mapping that will be applied to interpret the graphic input is determined in this section. The Selection section offers the ability to switch the designation of control points.

![Figure 4.14.2.1](image)

The controls of nurbz objects for (a) surfaces created from control lines and (c) nurbz derived from a primitive.

![Figure 4.14.2.2](image)

The nurbz Edit Control Options palette.
When selecting one of the icons in the **Action** group, the respective operation is applied. It can be a geometric (move, rotate, or scale) or a topological (insert, delete) transformation. Note that the execution of the geometric transformations varies. Move can be executed using either the click-and-drag or the click-and-click method. The others can only be executed using click-and-click.

**Geometric transformations**

- **Move**

  With this tool active, when using the click-and-drag method, click on a control and move the mouse while you keep its button pressed. When using the click-and-click method, click on a selected control, release the button, move it to a desired position, and click again.

- **Rotate**

  With this tool active, when using click-and-click, the only method available, click on a selected control (1) to pick it, click to establish the center of the rotation (2), click to start the rotation (3), and click to establish the angle of rotation (4), as shown in Figure 4.14.2.3.

- **Scale**

- **Uniform Scale**

  With one of these tools active, click on a selected control (1), click to establish the basis of the scaling operation (2), click to start the scaling (3), and click to establish the scaling factor (4), as shown in Figure 4.14.2.4, where four preselected controls are scaled with a single Uniform Scale operation.

*Figure 4.14.2.3:* Rotation applied to controls of (a) a nurbz object and (b) the result.

*Figure 4.14.2.4:* Uniform scale applied to controls of (a) a nurbz object and (b) the result.
Inserting and deleting control points while editing a nurbs surface

As with nurbs curves it is possible to insert and delete control points while using the Edit Controls tool on a nurbs or curve on nurbs object.

(a) **Click on a control segment:**
At the click point a new set of control points is inserted. If the segment clicked runs in the horizontal (U) direction, the new control points are inserted in the vertical (V) direction and vice versa. This is illustrated in Figure 4.14.2.5.

(b) **Click on a control point:**
This inserts two sets of new control points, running through the clicked point, in the U and V directions. This will usually result in sharp creases intersecting at the clicked point, as in Figure 4.14.2.6. These creases are called *surface discontinuities*, which means that the surface is not smooth in that area. There are a number of modeling operations that will fail when nurbs surfaces with discontinuities are selected. Thus the insertion of control points on top of each other should be applied with caution.

(c) **Click on the surface:**
Clicking away from any controls inserts a new set of control points in the V direction without changing the shape, which methods (a) and (b) above do. Pressing `command+ctrl` (Mac) or `ctrl` (Windows) with the click, makes the insertions in the U direction. The control points may not be close to the clicked point. See Figure 4.14.2.7 for an example.
Delete

(a) **Click on a control segment:**
This operation deletes all the control points that connect to the clicked segment and run in the same direction, as illustrated in Figure 4.14.2.8.

(b) **Click on a control point:**
This operation deletes the control points in both the U and V direction that are on the lines going through the clicked point. This is illustrated in Figure 4.14.2.9.

In the **Action** group of the Edit Control Options palette, under the tool icons, there are three options that affect the how the mouse movement is interpreted, as follows:

**Reference Plane:** When this option is selected, the cursor is mapped to the reference plane during interactive editing, or perpendicular to it, if the Perpendicular switch (Window tool) is on.

**Local Normal:** When this option is selected, the cursor is mapped to the normal of the control that is nearest to where the mouse is clicked. During editing, the direction of the normal is displayed on the screen as a little arrow.

**Average Normal:** When this option is selected, the cursor is mapped to the average normal of the selected controls. When only one control point is picked, the effect of this item is the same with the previous.

**Selection:** The items in the two menus of this group display the type of a control point you click on, while in control edit mode, and they can also be used to set the type of a control point of a nurbz surface.

**Smooth U, Smooth V:** Each of these menus contains three items:

- **On:** This item indicates that a point is smooth.
- **Off:** Indicates that a point is non-smooth.
- *****: This string of asterisks is used when multiple points are picked and they are of different types; that is some are smooth and some are non-smooth. Switching this item to either **On** or **Off** and clicking again sets all the selected points to that condition.
When this operation changes a non-smooth point to a smooth, the geometry of its neighboring segments is adjusted to make the segments colinear, if they are not already. This produces an immediately visible result. If, on the other hand, a smooth point is switched to non-smooth, there is no immediately visible result, but the point will behave differently when it is moved, as shown in Figure 4.14.2.10. When moving a non-smooth point, it moves independently of its neighboring points (a). Moving a smooth point also moves its neighboring segments, which behave as if they were connected at the moving point (b).

**Figure 4.14.2.10:** Moving control points: (a) non-smooth and (b) smooth.

---

### Edit Surface

Clicking on a nurbs object with the Edit Surface tool highlights the object and sets it in surface edit mode. No controls of the object are displayed, however, an arrow is drawn starting at the click point and pointing in the direction of the surface’s normal, if the **Surface Normal** option is on. This option can be selected from the Edit Surface Options palette, which appears as soon as a nurbs surface is clicked with the Edit Surface tool. The palette is shown in Figure 4.14.2.11.

The only geometric transformation that can be applied in nurbs surface edit mode is move and a click-and-drag method is used exclusively. This is intended to be a solely “what you see is what you get” operation that is applied without the interference of any control graphics. However, there are two options for controlling the motion of the mouse: **Reference Plane** and **Surface Normal**. These are as for the controls edit mode.

In v. 4.0, the Edit Surface operation starts by clicking once on the object, which invokes the Edit Surface Options palette. A second click starts the actual surface editing. Between the first and second clicks the selection of **Reference Plane** or **Surface Normal** can be changed, which affects the direction of the motion that is applied. Note that this is different from previous versions of the program where a single click both started the operation and the surface editing.
4.14.3 Operations that preserve or generate NURBS controls

A number of operations, when applied to nurbz objects or spline surfaces, preserve their NURBS controls, as follows:

- Trim tool: Nurbz objects can be trimmed using the Trim tool, without losing their status as NURBS surfaces. After they are trimmed they can be edited with the Edit Controls tool and their controls are maintained together with the trimming information of the surface. The controls and parameters are preserved automatically and no special option needs to be turned on. However, only when the trimming process results in a nurbz surface that consists of a single face can the controls be preserved. For example trimming a nurbz surface with a line will maintain the NURBS controls. In contrast, trimming and stitching a nurbz with another nurbz causes them to lose their NURBS controls and converts them to a plain object. Examples of trimmed nurbz objects are shown in Figures 4.14.3.1, 4.14.3.2, and 4.14.3.3.

- Parallel tool: Using this tool to derive a Single Parallel (Surface) from a nurbz surface preserves the NURBS parameters of the original nurbz surface.

- 2D Surface tool: When using this derivative tool to generate an object with the Each Selected Entity option and a face that is a spline surface is picked, a nurbz surface is created with all its NURBS controls preserved.

- Edit Controls tool: This tool will automatically turn a single face plain object, whose underlying surface is a spline, into a nurbz object.

- Importing SAT and IGES files: Any single face object, which is a spline surface, is imported as a nurbz object.

Figure 4.14.3.1:
(a) A nurbz object (b) with its control lines.
(c) The object after trimming it with some lines.
(d) The object after moving some control points.
Figure 4.14.3.2:
(a) A nuriz surface (b) is trimmed with a pattern and (c) is then deformed.
Shown (1) with its editable controls and (2) without them.

Figure 4.14.3.3:
(a) A nuriz surface (b) is trimmed with a streamer pattern and (c) is then twisted.
Shown (1) with its editable controls and (2) without them.
4.14.4 Deriving cross sections and more nurbz

The tool discussed here can be used to create cross section curves from an ordered set of profile curves. In addition, it can optionally be used to create a nurbz surface by lofting the newly derived cross sections. The cross section curves are constructed by threading points along each of the profile curves in the order they were selected. A minimum of three profile curves must be selected. You can think of the profiles picked for this tool as V curves and the cross sections it derives as U curves.

![Nurbz Cross Sections Options dialog](image)

Either the prepick or post pick method can be used with this tool. With the postpick method, the Nurbz Cross Sections tool is activated first and then three profile curves are selected. With the prepick method, any number profile curves are picked first, followed by the selection of the Nurbz Cross Sections icon and a mouse click anywhere in the graphics window.

Note that the order in which the profiles are selected is significant, because it determines the order in which points will be threaded. Different orders result in different shapes, as illustrated in Figure 4.14.4.2.

This tool is affected by options selected from the **Nurbz Cross Sections Options** dialog, shown in Figure 4.14.4.1, which is invoked directly from the tool. It consists of three tabs: **Options**, **Display Resolution**, and **Status of Objects**. The latter two tabs are as in other dialogs. The **Options** tab contains options specific to this tool.

**Number Of Curves**: This option determines the number of cross sectional curves that will be generated or used to create the nurbz surface. The default is 10.

![Figure 4.14.4.2: The order in which the profiles are picked affects the shape of the cross sections.](image)
Place Along Curve: The cross sectional curves are evenly placed along the profile curves as shown in Figure 4.14.4.3. With this method the points along each of the profile curves are determined by dividing the length of the curve by the number of desired cross sections. Note that the cross sections will generally not be planar, whenever more than 3 profiles are used.

![Figure 4.14.4.3: Nurbz cage using Place Along Curve: (a) Original curves, (b) resulting cross section curves, and (c) nurbz surface.](image)

Use Reference Plane: The cross sectional curves are generated along the intersection of planes parallel to the reference plane with the profile curves, as shown in Figure 4.14.4.4. With this method the points along each of the profile curves are determined by intersecting a parallel plane with the curve. The cross sections will always be planar. The sections are generated in the range where all profiles intersect planes parallel to the reference plane.

![Figure 4.14.4.4: Use Reference Plane option: (a) original profiles, (b) resulting cross sections, and (c) nurbz surface.](image)

Note that for this option to work, the position and direction of the profiles should be close to perpendicular to the reference plane. Otherwise it may not be possible to derive meaningful sections of the selected profiles.

Create Surface: When this option is selected, a nurbz surface is created. When this option is off, which is the default, the result is a set of cross section curves. The surface is constructed using the By Tight Lofting construction method discussed in section 4.14.1 of this Addendum. Note that if a different nurbz creation method is desired then the Nurbz tool can be applied directly to the cross sections.

Degree, Caps, and Rounding: These options are as with the Nurbz tool (see section 4.14.1).
4.14.5 Reconstructing nurbz surfaces

Reconstruct Nurbz

This tool is used to change the degree and the number of control points of a nurbz surface while maintaining the shape as much as possible. The desired changes are set in the Reconstruct Nurbz Options dialog (Figure 4.14.5.1) that is invoked from this tool.

**Change Degree:** When this option is selected, the degree of the surface will be changed.

**Length (U), Depth (V):** When these options are on, the degree in the length (U) or depth (V) direction will be changed to the value entered in its field.

**Keep Shape:** When this option is on, the surface shape is maintained when increasing the degree, by adding more control points. If this option is off, the degree of the surface is changed without changing the number of control points and hence the shape may not be maintained. The surface shape cannot always be maintained when lowering the degree, as often control points need to be removed. This option is illustrated in Figure 4.14.5.2.

![Reconstruct Nurbz Options dialog](image)

*Figure 4.14.5.1:* The Reconstruct Nurbz Options dialog.

![Change Degree from 2 to 5:](image)

*Figure 4.14.5.2:* Change Degree from 2 to 5:
(a) Original surface and after the change with **Keep Shape** (b) on and (c) off.
**Change Control Points:** When this option is on, the number of control points will be changed.

**Relative To Curvature:** When this option is selected, the surface is reconstructed in such a way that control points are distributed relative to the curvature of the surface. The reconstructed surface will have more control points in the areas of high curvature and less in flatter regions. This option is illustrated in Figure 4.14.5.3.

![Figure 4.14.5.3: Changing control points using Relative To Curvature: (a) Original, (b) 70° and (c) 15°.](image)

**Max Normal Deviation:** The value entered in this field, which specifies degrees, determines the density of the control point distribution along the surface. Higher values produce less number of control points and lower values produce high number of control points.

**Number Of Control Points:** When this option is on, the surface is reconstructed by setting the number of its control points to the exact number specified in its \( U \) and \( V \) fields.

**Length (U), Depth (V):** When these options are on, the control points in the length (U) or depth (V) direction will be reconstructed according to the values entered in their fields. These options are illustrated in Figure 4.14.5.4.

![Figure 4.14.5.4: Changing control points using Number Of Control Points: (a) Original surface, (b) \( U = 5 \), \( U = 5 \), and (c) \( U = 20 \), \( U = 20 \).](image)
Swapping Length (U) And Depth (V): When this option is on, the U and V control points are swapped, changing the U and V directions of the surface. Note that this is the last step of the reconstruction process. Consequently, all other settings in this dialog are applied before the U/V swapping.

Minimize Control Extents: When this option is on and the tool is applied to a trimmed Nurbz, the control points are trimmed back to the extents of the surface. For examples, see Figure 4.14.5.5.

Preview: When on, which is the default, the Surface Reconstruct preview dialog (Figure 4.14.5.6) is invoked, when picking the object to be reconstructed. This dialog contains a preview window on the left and options under it and on the right, organized under two tabs, as follows.

Options: This tab contains all of the options found in the Reconstruct Nurbz Options dialog.

Display Resolution: These options are as described in section 4.1.1.

Show Original: When this option is selected, the original Nurbz surface is shown in the preview window. This option is off by default.

Show New: When this option is selected, the reconstructed Nurbz surface is shown in the preview window. This option is on by default. Note, that when first opening the dialog, the new, reconstructed surface is identical to the original surface.

These two options can be used to show the difference between the original and new, if the shape of the surface changed due to new parameters entered by the user.

Show Controls: When this option is selected, the controls for the original or new surface are drawn.

Show Deviation: When this option is selected, the deviation of the new surface from the original surface is calculated. It is shown as a series of lines. Each line is drawn from a point on the original surface to the closest point on the new surface. The collection of lines indicates where the deviation of the new surface is largest.

Maximum Deviation: This field shows the length of the longest deviation line.

# Of Samples Control Point: The value entered in this field determines how many deviation lines are calculated and drawn in the u and v direction of the surface. The value is multiplied by the # of control points in each direction. For example, a surface with 10 control points in u and 20 control points in v will have: (2 * 10) * (2 * 20) = 800 deviation lines calculated, with the default value 2.
**Reconstruct:** When you click on this button, the reconstruction is executed and previewed in the dialog.

**Surface Info:** This is a group with information fields displaying the current **Degree** and number of **Control Points**, in both the U and V directions.

*Figure 4.14.5.6:* The **Surface Reconstruct** dialog showing a nurbz surface of degree 3 with (a) 20 x 20 control points before reconstruction, (b) after reconstructing is to 5 x 5 controls points, and (c) showing the deviation lines.
4.14.6 Extracting curves

**Extract Curve**

This tool is used to produce curves by extracting them along the U or V directions of nurbs surfaces. The resulting curve will be a spline for second and third degree surfaces or a NURBS curve for higher degree surfaces. It invokes the Extract Curve Options dialog, shown in Figure 4.14.6.1.

**At Click Point**: When this option is selected, a curve is extracted at the point on the surface where the mouse is clicked.

**Length (U), Depth (V)**: When these options are on, a curve is generated along the length (U) or depth (V) or both directions of the surface at the click point on the surface (Figure 4.14.6.2).

**At Interval**: When this option is on, a set of curves is extracted at intervals along the surface.

**Length (U), Depth (V)**: When these options are on, curves are generated along the length (U) or depth (V) or both directions of the surface, across the extents of the surface. The number of curves is determined by the text fields to the right. For examples see Figure 4.14.6.3.

---

**Figure 4.14.6.1**: The Extract Curve Options dialog.

**Figure 4.14.6.2**: Extract At Click Point in Length (U) and Depth (V): (1) Open and (2) closed shapes. (a) Original shape and (b) resulting curves.

**Figure 4.14.6.3**: Extract At Interval: (1) Open and (2) closed shapes. (a) Original shape and (b) resulting curves.
4.14.7 Attaching nurbz surfaces

Attach Nurbz

This tool attaches the nurbz surface picked first to the surface picked second. There are different alignment options, which are selected from the Attach Nurbz Options dialog (Figure 4.14.7.1), that is invoked from the tool.

Stretch Surface: This option aligns only the picked edge of the first surface to the picked edge of the second surface.

Keep Opposite Edge: When this option is selected, the opposite edge of the surface is preserved. This is done by inserting additional control points in the surface when there are not enough present to avoid disturbances to the opposite edge.

Move Surface: This option aligns the first surface to the second surface by aligning their respective normals. That is, the whole first surface is moved as the picked edge is aligned.

Alignment: This menu offers four different methods for aligning the stretched or moved curves. These options represent increasing levels of smoothness between the curves. The alignment is performed by making adjustments to some of the control points to achieve the desired results as follows:

Positional (C0): The last end control point from one curve is repositioned to the end control point of the second curve. For an example, see Figure 4.13.4.2 (a).

Tangents (G1): In addition to the adjustment for Positional (C0), the control point previous to the end control point of the first curve is repositioned such that it is colinear with the end control point of the second curve and its next control point. For an example, see Figure 4.13.4.2 (b).

Equal Tangents (C1): In addition to the adjustment for Tangents (G1), the same control point is adjusted to be at the same distance from the end control point of the second curve as the next control point is from end control point of the second curve. In NURBS mathematical terminology, the curves are said to have the same first derivatives. For an example, see Figure 4.13.4.2 (c).

Curvature (C2): In addition to the adjustment for Equal Tangents (C1), the second control point from the end is repositioned to form a concave relationship with the other control points. In NURBS mathematical terminology, the curves are said to have the same first and second derivatives. For an example, see Figure 4.13.4.2 (d).

Merge: When this option is on, in addition to being aligned, the two surfaces are merged into one surface.
Figure 4.14.7.2: Attach with **Stretch Surface** on: (1) Wire Frame and (2) Shaded Render. (a) Original surface; (b) **Positional (C0)**, (c) **Tangents (G1)**, (d) **Equal Tangents (C1)**, and (e) **Curvature (C2) Alignment**.

Figure 4.14.7.3: Attach using **Move Surface**: (a) Original surfaces. Attaching (b) \( p_1 \) to \( p_2 \), (c) \( p_1 \) to \( p_3 \), and (d) \( p_1 \) to \( p_4 \).
4.14.8 Blending nurbz surfaces

Blend Nurbz

This tool creates a blend surface between two different surfaces or between two edges of the same surface. It invokes the Blend Nurbz Options dialog shown in Figure 4.14.8.1.

Alignment, Merge: These options are as for the Attach Nurbz tool.

Degree: This group determines the degree for the depth (V) direction of the new surface.

From Surface: When selected, the depth (V) degree of the new surface is the same as the source surfaces degree. If the source surfaces have different degrees, the higher degree is used.

New Degree: When selected the depth (V) degree is determined by the value in the text field to the right. The default is 5.

Figure 4.14.8.1: The Blend Nurbz Options dialog.

Figure 4.14.8.2: Blend Nurbz: (a) Original object; (b) Positional (C0), (c) Tangent (G1), (d) Equal Tangents (C1), and (e) Curvature (C2) Alignment.
4.14.9 Merging nurbz surfaces

Merge Nurzbz

This operation merges two nurbz surfaces that have a matched edge into a single new nurbz surface. The operation automatically recognizes the matched edge(s). Two edges are considered to be matched when they have the same number of closely positioned control points, are of the same degree, and have the same number of knots. The operation is refused when no matched edges are found. If two pairs of matched edges are found, the new surface will be derived by merging both pairs of edges. The merge options are selected from the Merge Nurbz Options dialog, shown in Figure 4.14.9.1, that is invoked directly from the tool.

Either the prepick or postpick method can be used with this tool. With the postpick method, the Merge Nurzbz tool is activated first and then a surface is picked. With the prepick method, any number of surfaces are picked first, the Merge Nurzbz tool is activated next, and then the mouse is clicked anywhere in the graphics window. Examples are shown in Figure 4.14.9.2.

Figure 4.14.9.1: The Merge Nurzbz Options dialog.

Figure 4.14.9.2: Merging (1) open and (2) closed nurbz surfaces: (a) Original and (b) resulting surfaces.
Align Surfaces: This option aligns the first surface to the second surface along the merged edge(s), according to the following alignment method.

Alignment: This menu offers four different methods for aligning the merged surfaces. These options are as for the Attach Nurbz tool. Note that the Positional (C0) alignment does not change the shape of the surfaces. Examples are shown in Figure 4.14.9.3.

Make Symmetric: The option makes the control points along the seam symmetric. This is accomplished by repositioning the adjacent control points to the merged edge such that they are symmetric about a plane as determined by the Use Edge and Use Reference Plane options. This option is most useful when merging surfaces where one is a mirrored copy of the other. This option is only available when the Equal Tangents (C1) option is selected from the Alignment menu. Examples are shown in Figure 4.14.9.4.

Use Edge: When this option is selected, the Symmetry plane is derived from the plane defined by the merged edge. If the edge is non-planar, then the best matching plane is calculated. If the edge is linear (all the control points are collinear) the operation is refused.

Use Reference Plane: When this option is selected, the current reference plane is used as the Symmetry plane is derived from the plane defined by the merged edge. If the edge is non-planar, then the best.
**Smooth Surfaces**: This option smooths the merged edge by manipulating the knots at the edge rather than the adjacent control points as with the *Align Surfaces* option.

**Smoothness**: This slider controls the amount of smoothing applied with 0% representing no smoothing (hard edge) to 100% which is completely smooth. Note that the location of the adjacent points to the edge affects how smooth the results will be. If the adjacent control points are very close, the smoothing effect will be minimal even at the 100% smooth setting. Examples are shown in Figure 4.14.9.5.
4.14.10 Extending nurbz surfaces

**Extend Nurbz**

This tool extends a nurbz surface from a picked edge by a preset distance or to a curve. This is performed by adding additional surface at the picked edge. The old and the new portions may optionally be merged. The tool invokes the *Extend Nurbz Options* dialog, shown in Figure 4.14.10.1.

**Extend At Distance:** When this option is on, the surface is extended at a selected edge by a given distance. Either the prepick or postpick method can be used with this option. With the postpick method, the Extend Nurbz tool is selected first and then an edge of a nurbz surface is picked. With the prepick method, any number of edges of a nurbz surfaces are picked first, followed by the selection of the Extend Nurbz tool and a mouse click anywhere in the graphics window.

**Distance:** This is the distance that the surface will be extended by.

**Extend Direction:** This group of options determines the direction of the extension relative to the selected edge.

**Surface Edge:** When this option is on, the extension is constructed by constructing a new parallel curve along the normals to the surface edge, as shown in Figure 4.14.10.2. This method yields the most accurate result, however, it will not work well on surfaces with few control points. The distance is the distance along the surface between the original edge and the new extension edge.

*Figure 4.14.10.1:* The *Extend Nurbz Options* dialog.

*Figure 4.14.10.2:* Extending a nurbz surface using *Surface Edge.*
**Control Points:** When this option is selected, the extension is constructed by establishing new control points along the vector formed by the previous control point to the edge. Examples are shown in Figure 4.14.10.3. The distance is the distance between the control points at the edge and the new control point at the extension.

**Reference Plane:** When this option is selected, the extension is constructed by establishing new control points along a direction perpendicular to the reference plane. Examples are shown in Figure 4.14.10.4. The distance is the same as with **Control Points**.

*Figure 4.14.10.3:* Extending a nurbz surface using **Control Points**.

*Figure 4.14.10.4:* Extending a nurbz surface using **Reference Plane**.
**Extend To Curve:** When this option is on, the surface is extended at a selected edge to an existing curve. Either the prepick or postpick method can be used with this option. With the postpick method, the Extend Nurbz tool is selected first and then an edge of a nurbz surface is picked followed by the selection of the curve to extend to. With the prepick method, an edge of a nurbz surface is picked followed by the selection of the curve to extend to, followed by the selection of the Extend Nurbz tool and a mouse click anywhere in the graphics window. Multiple pairs of edges and curves can be prepicked and the Extend Nurbz tool will handle them in the order selected. Examples are shown in Figure 4.14.10.5.

![Figure 4.14.10.5: Extending a nurbz surface using Extend To Curve.](image)

**Alignment, Merge:** These options are as for the Attach Nurbz tool (see section 4.14.7). Alignment examples are shown in Figure 4.14.10.6.

![Figure 4.14.10.6: Nurbz surface extension Alignment:](image)

(a) **Positional (C0)**, (b) **Tangents (G1)**, (c) **Equal Tangents (C1)**, and (d) **Curvature (C2)**.
4.14.11 Splitting nurbz surfaces

This tool splits a surface in either U or V direction or in both U and V simultaneously, at the click point or at intervals. The resulting surfaces will have the same degree as the original surface in both U and V. Note that splitting a nurbz surface that is closed in the U direction has the effect of opening the surface at the split point.

Either the prepick or postpick method can be used with this tool. With the postpick method, the Split Nurbz tool is selected first and then a nurbz surface is picked. With the prepick method, any number of surfaces are picked first, followed by the selection of the Split Nurbz tool and a mouse click anywhere in the graphics window.

**At Click Point**: When this option is selected, a surface is split at the point on the surface where the mouse is clicked.

**Length (U), Depth (V)**: When these options are on, the surface is split along the length (U) or depth (V) or both directions of the surface at the click point on the surface. Examples are shown in Figure 4.14.11.2.

**At Interval**: When this option is on, a set of splits are applied at intervals along the surface.

**Length (U), Depth (V)**: When these options are on, the surface is split along the length (U) or depth (V) or both directions of the surface, across the extents of the surface. The number of splits is determined by the text fields to the right. For examples see Figure 4.14.11.3.
Figure 4.14.11.2: Splitting nurbs surfaces at click point.
(1) Open surface split in \textbf{Length (U)} and \textbf{Depth (U)}. (2) Closed surface split in \textbf{Length (U)}.
(a) The original surface, (b) after the split, (c) after split pieces moved.

Figure 4.14.11.3: Splitting nurbs surfaces at intervals.
(1) Open surface split in \textbf{Length (U)} and \textbf{Depth (U)}. (2) Closed surface split in \textbf{Length (U)}.
(a) The original surface, (b) after the split, (c) after split pieces moved.
4.14.12 Untrimming nurbz surfaces

**Untrim Nurbz**

This tool removes trimming curves from a trimmed nurbz. If all of the trimming curves are removed, the object becomes an untrimmed nurbz.

Either the prepick or postpick method can be used with this tool. With the postpick method, the Untrim tool is selected first and then an outline or an segment of a trimmed nurbz surface is picked. With the prepick method, any number of outlines or segments of a single or multiple trimmed nurbz surface are picked first, followed by the selection of the Untrim tool and a mouse click anywhere in the graphics window. When selecting a segment, the outline to which the segment belongs will be removed.

This tool invokes the **Untrim Nurbz Options** dialog, shown in Figure 4.14.12.1.

**Outside Boundary**: This group of options controls how the outer boundary is handled when it is selected.

- **Remove**: With this option, the outer boundary is removed making any untrimmed curves outer or boundary curves.

- **Restore Control Extents**: When this option is used, the trimmed outer boundary is removed and the boundary is restored to the limits of the nurbz control points. These options are illustrated in Figure 4.14.12.2.

- **Untrim All**: When this option is on, all trimming curves are removed from the trimmed nurbz regardless of the outline or edge that was selected.

![Untrim Nurbz Options dialog](image)

*Figure 4.14.12.1: The Untrim Nurbz Options dialog.*

![Untrimming a surface](image)

*Figure 4.14.12.2: Untrimming a surface. (a) The original trimmed surface; (b) after untrimming one hole, (c) after untrimming the boundary with the **Remove** option, (d) untrimming boundary with **Restore Control Extents**, and (e) **Untrim All**.*
4.14.13 Creating curves on nurbz surfaces

**Curve on Nurbz**

This operation creates a new object by combining a curve and a surface. Alternatively, it can be used to add curves to existing trimmed nurbz or a curve on nurbz objects. The curves are created by sketching on a nurbz object or by mapping an existing curve onto a nurbz object. The source surface can be a nurbz, trimmed, nurbz or curve on nurbz object. The resulting object can be either a curve on nurbz object or a trimmed nurbz.

Either the prepick or postpick method can be used with this tool. With the postpick method, the Curve on Nurbz tool is activated first and then the surface is selected. With the prepick method, the surface object is picked first with the Pick tool, followed by the selection of the Curve on Nurbz tool. The next click on the surface starts the sketching of the curve, using the selected drawing mode. The drawing modes work the same as the corresponding tool from the Lines, Splines and Arcs group of tools. As the mouse is moved and points are drawn, the points are mapped onto the surface of the nurbz. Interactive input stops when the mouse moves outside the boundary of the nurbz surface.

This tool invokes the **Curve On Nurbz Options** dialog, shown in Figure 4.14.13.1.

**Sketch on Surface**: When this option is selected the curve is created by sketching on the surface using one of the drawing modes identified by the icons in this group. When sketching on a surface to create a trimmed nurbz, the whole underlying surface is temporarily displayed. This allows you to see the full extent of the surface, even when it was previously trimmed, and help determine where on the surface the mouse can be clicked. Note that, to create a hole, the curve must be drawn inside the area of the surface. When drawing a boundary that will enclose a surface, the curve needs to be drawn in an empty (trimmed) area of the surface. A boundary curve can also be drawn inside the area of a surface when the surface is completely untrimmed.

![Figure 4.14.13.1: The Curve On Nurbz Options dialog.](image)

![Figure 4.14.13.2: Sketch on Surface examples.](image)
Map Existing Curve: When this option is selected, one or more existing curves are mapped onto the nurbz surface. Either the prepick or postpick method can be used with this option. With the postpick method, the Curve on Nurbz tool is selected first and then two objects are selected. If the Object Type option is set to Trimmed Nurbz (see below), one of the objects needs to be a Nurbz or Trimmed Nurbz object. If the Object Type option is set to Curve On Nurbz, one of the objects must be a Nurbz or Curve on Nurbz object. You may also select any object which can be converted to a nurbz object, such as a sphere or cylinder. The second object must be a curve, which can be an arc, circle, ellipse, spline or composite curve or any curve which can be converted to a spline. It can also be a plain object which is an open or closed wire or a plain faceted surface object with a single face. If the Object Type option is set to Trimmed Nurbz, the curve must be closed.

With the prepick method, any number of curves and one surface are picked first. The order in which they are picked does not matter. Then the Curve on Nurbz tool is selected and the mouse is clicked anywhere on the graphic window.

When the operation is executed, the selected curves are transformed from the 3D world space into the parameter space of the surface according to the following rules:

- The x axis of the current reference plane represents the U direction of the surface. The y axis of the current reference plane represents the V direction of the surface.
- The curves must all be planar and must be contained in the current reference plane. The best way to execute this tool is to draw the curves on the XY plane using the world X axis as the reference for the U direction and the world Y axis for the V direction of the surface.
- The curves are scaled so that their extent fits within the parameter space of the surface. The value entered in the Max Pspace Size field of the Curve On Nurbz Options dialog adds an additional scale factor to the parameter space curves.
- When the curves are first mapped, they are centered in the surface’s parameter space.
- When the Object Type option is set to Trimmed Nurbz, and the selected surface is already a Trimmed Nurbz object with existing parameter space curves, the new curves are fitted in the largest empty area of the parameter space, but no larger than the Max Pspace Size value. This avoids intersections of the new curves with the existing curves.
- When the Object Type option is set to Trimmed Nurbz, the selected curves may not intersect each other, may not self intersect and cannot have duplicate points.
- The selected nurbz surface may not have any discontinuities (sharp creases in the middle of the surface) or poles. A pole is where all the control points at the end of a nurbz surface collapse to a single point.

Max Pspace Size: The value entered in this field determines how large the selected curves are when mapped into the parameter space of the surface. At 100%, the bounding box of the curves stretch the entire parameter space. At a smaller value, the parameter space curves are scaled accordingly. Examples of parameter space curves mapped at different scales are shown in Figure 4.14.13.3.
Object Type: The options in this group determine the type of the object that will be generated, as follows:

Curve On Nurbz: When this option is selected, the picked curves are mapped onto the surface and become 3D curves. The surface itself disappears, but manifests itself through the shape of the curves mapped onto it. Examples of curves mapped onto nurbz surfaces are shown in Figure 4.14.13.4.

Trimmed Nurbz: When this option is on, the picked curves are mapped onto the surface as trimming curves. The two options under this (Curves Enclose Surface and Curves Enclose Holes) determine whether the mapped curve will enclose the surface or a hole. For examples see Figure 4.14.13.5.

Edit: When this option is selected, the Edit In Parameter Space dialog is invoked after selecting the curves and the surface. This dialog shows the parameter space curves in a 2D preview window and the resulting 3D objects in a 3D window. This dialog is described in more detail in section 4.3.7.
**Editing curves on nurbz and trimmed nurbz**

Curves on nurbz and trimmed nurbz objects can be edited in 3D space and in 2D space. To edit the 3D controls of the surface, the Edit Controls tool can be used in the usual manner. For curves on nurbz objects, the controls drawn are the controls of the original nurbz surface. Changing the controls modifies the shape of the surface and the curve mapped on the surface adjusts its shape to always follow the surface. This is illustrated in Figure 4.14.13.6.

To edit the controls of the trimming curves of a trimmed nurbz object or the controls of the curves of a curve on nurbz object, the Edit In Parameter Space tool is used. This tool invokes the **Edit In Parameter Space** dialog and is described in more detail in section 4.3.7.

Curve on nurbz objects can also be used very effectively as the input curves for other derivative operations. For example, a curve on nurbz object may be used as the path for an axial sweep. When editing the sweep object with the Edit Controls tool, the path object’s controls are the control points of the curve on nurbz object’s nurbz surface. Changing these control points will cause the sweep object to always follow the shape of that nurbz surface. This process is illustrated in Figure 4.14.13.7.

**Figure 4.14.13.6:** Editing the controls of the surface of a curve on nurbz object.

**Figure 4.14.13.7:** Using a curve on nurbz object as the path for an axial sweep: (a) the curve on nurbz is created, (b) the axial sweep is created using the curve on nurbz as the path, and (c) the axial sweep is edited with the Edit Controls tool.
4.14.14 C-Meshes

A mesh is an open or closed surface that is subdivided to four or three-sided tiles. In addition to the types of meshes discussed in section 4.10 (plain and smooth meshes), form-Z also offers the ability to create controlled meshes (or c-meshes). These are generated from control lines that form a control net. The net is the three dimensional counterpart of the control lines used with the c-curves. Like the c-curves the c-meshes are objects that carry their controls within their internal representations, which makes them easily editable and changeable after their initial creation. The discussion in this section starts with an overview of the c-mesh generation process and the c-mesh features. A more detailed presentation of each part of the process follows.

The stages of the c-mesh generation process are illustrated in Figure 4.14.1.1. A c-mesh is created from control lines which are drawn first and are positioned in 3D space (Figure 4.14.1.1(a)). These lines may be open or closed and they are selected in the order they will be connected. The direction along a control line is called the length and the direction across the control lines is called the depth of the mesh (Figure 4.14.1.1(a)). The distinction between length and depth is significant since independent parameters are applied in each direction.

From the control lines, a control net is created, as shown in Figure 4.14.1.1(b). In its simplest form, the net is constructed by connecting the points of the control lines in the depth direction.

Next, each control line is curved (lengthwise), as shown in Figure 4.14.1.1(c). Any of the available curve types can be used.

Then, on each curve new points are generated. An equal number of points is generated on each curve and these points are evenly distributed along each curve (Figure 4.14.1.1(d)).

The new points are connected along the depth direction resulting in new control lines, as shown in Figure 4.14.1.1(e). These depth control lines are also curved (Figure 4.14.1.1(f)).

New points are generated on the depth curves. As for the length, an equal number of evenly distributed new points are generated on all the depth curves. That number may be different from the number of points generated in the length direction. After this stage is completed, all the points required for the creation of a c-mesh are available and the c-mesh is constructed, as shown in Figure 4.14.1.1(g). In Figure 4.14.1.1(h), the c-mesh is shown together with its control net as it appears in edit mode.

Note that the c-mesh has a “rectangular” structure (topology). However, its shape does not have to be rectangular. There are many variations that can be used when generating c-meshes. The most notable is the option to bypass some of the stages of the process and use the control lines directly. These variations are discussed later in this section.
Figure 4.14.1: The c-mesh generation process.
C-Mesh

This tool is used for generating and editing c-meshes. The C-Mesh tool is first used for the initial generation of a c-mesh. After it has been generated, the shape of a c-mesh may be edited and changed by using the C-Mesh tool again.

The **C-Mesh Options** dialog (Figure 4.14.14.2) can be invoked directly from the C-Mesh tool. It contains four tabs: **Options**, **Object Type** (Figure 4.14.14.3(a)), **Smoothing** (Figure 4.14.14.3(b)), and **Status Of Objects** (section 4.5).

Typically, the creation of a c-mesh requires the selection of a number of control lines and is best executed using the prepick method. The postpick method can be used when a c-mesh is generated between only two control lines. To generate a mesh using the prepick method, the control lines are first selected with the Pick tool, in the order in which they will be connected. Then the C-Mesh tool is activated and the mouse is clicked anywhere in the project window.
The distinction between the *length* and the *depth* of a mesh made earlier is highly significant since independently selected parameters are applied in each of these directions. The dialogs that contain the optional parameters of the c-meshes are structured in a manner which clearly reflects this distinction.

At its top, the **C-Mesh Options** dialog contains options similar to those used in the generation of the c-curves. When the **Construct Plain Mesh** option is selected, the system will generate a plain mesh object rather than a c-mesh. A plain mesh can only be edited when it is first generated and the C-Mesh tool cannot be applied to it. By default this option is off.

When the **Adjust To New Parameters** option is selected and the C-Mesh operator is used to edit a previously generated mesh, the mesh will be regenerated using the parameters currently selected. If the **Adjust To New Parameters** option is off, the mesh will retain the parameters with which it was last generated.

A mesh can be constructed directly, by selecting the **Construct Directly** option, or it can be edited, by selecting the **Edit** option. This is true for both the initial creation and subsequent editing of c-meshes. When the C-Mesh tool is use to edit an existing c-mesh, the **Construct Directly** option is on, and the **Adjust To New Parameters** option off, it has no effect.

As with the c-curves, a c-mesh can be edited without showing its controls (*Hide Controls* selected) and the **Snap** option may be selected to facilitate the placement of points on other points.
Creating the control net

By definition, the control net is a structure of a rectangular topology, even though its shape by no means has to be rectangular. The net consists of a constant number of tiles across its length and a constant number of tiles across its depth. The numbers of tiles across the length and across the depth may differ. These numbers are determined by the currently selected options in the **Length Of Net** and **Depth Of Net** sections of the **C-Mesh Options** dialog (Figure 4.14.14.4).

**Length Of Net**: Closed or open control lines may be selected for the generation of a c-mesh, but the two types cannot be mixed (all should be closed or all should be open). The control lines may or may not have the same number of points, even though the use of lines with the same number of points is recommended (see below). When control lines with variable numbers of points are selected, the system will either refuse to execute or will make adjustments to the number of points, depending on which option is selected.

When the **Same Points** option is selected, the program expects all the control lines picked for the creation of a c-mesh to have the same number of points. These points are used for the generation of the control net, as shown in Figure 4.14.14.5(a). If the control lines are found to have variable numbers of points, the program beeps and issues an error message.

Using control lines with the same number of points is possibly the most appropriate method for generating c-meshes, since the user has complete control of how the net and subsequently the c-mesh will be structured. Colinear segments on control lines are acceptable, allowing you to insert points on segments so that the number of points of one control line may match the number of points on another. Thus, a control line may appear to consist of a single segment when in fact it has a sufficient number of “hidden” points to match the points of another line which appears to have more points. An example is shown in Figure 4.14.14.5(b). In general, inserting, deleting, and moving points on control lines, before these lines are selected for the generation of a c-mesh, are excellent tools for manipulating the structure and the shape of a c-mesh.

All the other options available for the net length do not require that the control lines have the same number of points. They will make adjustments to produce control lines with the same number of points before these lines can be used for the creation of a control net. Each represents a different method for making these adjustments. With all three methods, if the program encounters control lines with the same number of points, it behaves as if the **Same Points** option were selected. That is, it keeps the points as they are and makes no adjustments.
When the **Use Existing** points option is selected and the program encounters control lines with variable numbers of points, it uses the largest number of points encountered as a base number. All the control lines that have the same number of points with the base number remain as they are. New points are inserted in all the lines which have fewer points. These new points are inserted on top of existing points, which preserves the original shape of the control line. The additional points are evenly distributed among the existing points, by starting at the two ends of the line and moving towards its middle (see Figure 4.14.14.6).

When the **New Points** option is selected and the program encounters control lines with variable numbers of points, it again uses the largest number of points encountered as a base number. However, new points will be generated for all the control lines and the original points will be discarded (with the exception of the end points of open lines). The new points lie on the original segments always and are evenly distributed along the length of the control line. In general, the shapes of the control lines change when this method is used (see Figure 4.14.14.7).

When the **Partially New** option is selected and the program encounters control lines with variable numbers of points, it again uses the largest number of points encountered as a base number. For lines with fewer than the base number of points, it preserves the current points and inserts new points between them. It uses a matching process to find the correspondence of the current points to points of the base line and it then proportionally distributes the additional points between pairs of matched points. An example is shown in Figure 4.14.14.8.

**Depth Of Net:** The depth of net options control how many control lines will be used for the generation of the depth dimension of the net and how they will be positioned. The normal method is to use the selected control lines only in the position they are. This method is applied when the **Use Control Line Intervals** option is selected (default). However, you may also instruct the system to generate additional control lines from the control lines selected for the operation. This is done when the **Use Mesh Intervals** option is selected. With the latter option, the program will use the parameters currently selected for the generation of the mesh and also apply them for the generation of the net. These parameters are discussed in the “Creating the c-mesh” section. The effect of this option is to generate a net and a c-mesh which coincide along their depth.
Generating the control curves

After the construction of the control net, the original control lines selected for the generation of a c-mesh become part of the net, in their original or adjusted version, and constitute the control lines across the length of the net. New control lines are also generated for the depth of the mesh. These control lines are curved according to the parameters selected from the Options tab in the C-Mesh Options dialog. These parameters are selected independently for the length and the depth of the c-mesh. The control lines picked for a c-mesh may or may not have been curved with the C-Curve tool, prior to their selection. form•Z allows you to mix the two cases and to pick some control lines that are already curved and some that are not.

The Options tab also contains Adjust To New parameters options for both the length and the depth of the c-mesh. Note that these options are independent of the Adjust To New Parameters option found in the C-Mesh Options dialog. The latter affects all the other parameters except the curve smoothing (see next section). The Adjust To New options affect the generation of the curves only.

Length curves: When a c-mesh is generated for the first time and the Adjust To New option is selected for its length, all the control curves picked will be regenerated. In contrast, if the Adjust To New option is not selected, then all the control lines curved prior to selection will remain as they are and only the control lines that have not been curved will be curved at this time. This allows different types of curves to be used for the generation of a c-mesh. However, this is only true for control lines which did not require adjustments when the net was constructed. Recall that these adjustments insert new points to the control lines which are now different from what they were at the time they were curved. Because of this, the previously generated curves cannot be used and new curves are generated. To make certain that previously created curves are preserved and used, control lines with equal numbers of points should be used and the Same Points option (under Length Of Net) should be selected.

After the control lines have been curved, new evenly distributed points are generated on them, according to the selected parameters (see next section). These points are then used to create control lines in the depth direction.

Depth curves: When generating a new c-mesh, whether or not the Adjust To New option is selected for its depth makes no difference since all the depth curves will be generated for the first time. The depth control lines only exist after the length curves have been created and new points have been generated on them. These points are connected in the depth direction to produce the depth control lines. These are next curved according to the depth parameters selected in the C-Mesh Smoothing Options tab.
Creating the c-mesh

After the control lines of the net have been curved, they are used for the generation of the c-mesh. The critical aspect of this stage of the c-mesh generation process is the determination of how many points and patches (in the length and depth directions) will be created for the mesh and where they will be placed. These numbers depend on the options selected from the Mesh Length and Mesh Depth sections of the C-Mesh Options dialog, shown in Figure 4.14.14.9.

Mesh Length: Three methods are available for determining the number of points to be generated in the length direction of a c-mesh. The At Intervals option, which is complemented by an interval value typed in the field next to it, uses the lengths of the control lines to determine how many points will be created. Since the lengths of the control lines may vary widely, the average length is used and divided by the interval value to determine how many points will be created. This method works best where the lengths of the control lines are roughly the same.

The # Of Segments option, complemented by the numeric value typed in the field next to it, is the most direct and simply creates a number of points equal to the number of segments indicated plus one.

The At Control Points option causes the program to generate as many points as in the control lines. Whenever different curve parameters than those previously used are desired, the Smoothing tab has to be invoked to select them. When At Control Points and None is selected from the Smoothing tab, no new points are generated but the points of the net are used instead. When some curve (other than None) is selected, new points are generated and are placed as close to the original points of the control net as possible.

The At Control Points option with None has special significance. It does not apply any smoothing in the length direction but uses the control lines in their original shape instead. This allows more control over the form of a mesh. The points of the control lines can be placed exactly where we want the segments of the mesh to be generated and we can expect them to stay exactly where they are when the mesh is generated. This may include corner points and variable densities for the patches. Note that such control is not available when smoothing is applied since the system generates new points as it distributes the curvatures evenly. The distinction is illustrated in Figure 4.14.14.10. Note that in 4.14.14.10(b) we inserted “hidden” points where we wanted the segments of our mesh to be placed. Thus the control lines we used have the same number of points. In 4.14.14.10(c) we used the control lines as they were originally drawn and we let the system determine where to position the points.
**Mesh Depth:** The methods available for determining the number of points that will be created for the depth of a c-mesh are largely the same as those used for the mesh length, except for an additional suboption which applies only to the depth. When using the **# Of Segments** method you can tell the system to place the number of points shown in the segments field between pairs of control lines rather than between the first and the last control lines. This is done by selecting the **Per Segment** switch, whose effect is illustrated in Figure 4.14.11. Placing the same number of points between pairs of control lines produces unevenly distributed mesh points, which is sometimes desirable.

![Figure 4.14.10](image1)

**Figure 4.14.10:** (a) The control lines used to generate a mesh with (b) **At Control Points** and no **Smooth** selected, and (c) **# Of Segments** selected and set to 15, and **Smooth** selected for the length direction. In both cases, **At Control Lines** and no **Smooth** was selected for the depth.

![Figure 4.14.11](image2)

**Figure 4.14.11:** (a) The control lines used to generate a mesh with depth **# Of Segments** 8 and **Per Segment** (b) **off** and (c) **on**.
Types of c-mesh objects

Different types of c-meshes can be generated as determined by the options selected from the **Object Type** tab (Figure 4.14.14(a)).

When the **Open Ends** option is selected, which is the default, a simple surface mesh is created (Figure 4.14.14.12(a)). It starts with the control line selected first (front) and ends with the control line selected last (back). When the **Closed Ends** option is selected, the program will also add faces at the two ends of the mesh. If open control lines are used, the closed shapes required for these faces are created by connecting the endpoints of the first and the last control lines. For open control lines, this option produces a surface object. For closed control lines, this option produces a solid (Figure 4.14.14.12(b)). When **All Closed** is selected and closed control lines are used, this option is equivalent to the previous. When open control lines are used, then one more face is added to close the mesh object completely. This option always results to a solid object.

The **Ring** option produces meshes which are closed along their depth. That is, the last control line is also connected to the first. When open control lines are used, the resulting mesh is a surface object. When closed control lines are used, this option produces solid objects (Figure 4.14.14.13). Note that this option works well only when the control lines are arranged in a circular manner, or otherwise objects that cross themselves may be produced. **form-Z** will not restrict you from doing this, but objects with crossing faces are rarely desirable.

![Figure 4.14.14.12](image)
Figure 4.14.14.12: (a) One-sided surface mesh generated from open control lines.
(b) Meshed solid generated from closed control lines and with the **Closed Ends** option selected.
The **Round Front** and/or **Round Back** options can be selected to produce round ends for all types of mesh objects, except the rings (which have no ends). These options are complemented by the **Rounding Ratio** field. The ratio value is a percentage that is used as a scaling factor. This factor is applied to the end control line and determines the extent of the rounding. The rounding options work best when used in conjunction with the **Per Segment** option for the mesh depth (Figure 4.14.14.14).

By default, all the mesh objects that are not solids are generated as one-sided surface objects. Two-sided surface meshes (surface solids) may be produced instead by selecting the **Two Sided** option. By default, the c-mesh objects consist of four sided faces. The **Triangulate** option can be selected to produce triangular faces. While the latter guarantees planarity, it should be used cautiously since it can double the number of faces in a c-mesh. When simultaneously selected, the **Two Sided** and **Triangulate** options can quadruple the number of faces in a c-mesh object.

The directions of the faces of a mesh depend on the direction in which the control lines were drawn as well as the order in which they were selected. It is quite possible that the final directions are not what the user expected. When this occurs, a tool is available that allows you to reverse the directions, after the mesh has been generated (see subsection 4.21.8). When the mesh is a solid object, selection of the control lines in a reverse order will produce an inside-out object. You can instruct the program to check for this condition and to adjust the directions of the faces automatically. This is done by selecting the **Adjust Direction** option, which is selected by default. While there are rare cases where inside-out objects are desirable, typically they are not.
The c-mesh edit mode

The system enters the c-mesh edit mode when the Edit option is selected in the C-Mesh Options dialog and the C-Mesh operator is executed to generate a c-mesh for the first time or to reshape a c-mesh. You can tell that you have entered the c-mesh edit mode by the fact that the tool palette is grayed out, indicating that it is inactive. The window tools remain active and you can use them as you edit and manipulate the controls of the c-mesh. You can exit the c-mesh edit mode by double clicking anywhere in the window.

When in mesh edit mode and assuming that the Hide Controls option is not selected, the control net appears rubber banded and is shown in the pick color (default red). The c-mesh is also shown, as illustrated in Figure 4.14.15. When applying an editing operation, such as moving a point, the affected parts of the net follow the motion of the mouse. The c-mesh is regenerated as soon as the operation is completed (Figure 4.14.16).

As with the c-curves, c-mesh editing operations are applied by moving the mouse cursor onto a control, clicking, repositioning the mouse, and clicking again. As the mouse is moved from one entity to another, its cursor changes to indicate what entity is currently selected. What operation is applied is determined by the cursor at the time the mouse is clicked. Key commands may be used to switch to another cursor and another operation. The available operations and their cursors are summarized in the table of Figure 4.14.17. The cursors and the corresponding controls are also shown in Figure 4.14.15.

Figure 4.14.15: A c-mesh and its controls in edit mode.

Figure 4.14.16: The effects of manipulating the geometry of the net controls (a) to the mesh and (b) while in edit mode.
To **move a point** the mouse is clicked when the cursor is close to a control point and the cursor shows . After the first click the net segments around the point are rubber banded and follow the motion of the mouse. When the point is at the desired position, the mouse is clicked again to complete the operation.

To **move a segment of a length** or **a depth control line** the mouse is clicked when the cursor is close to a control segment and the cursor shows or , respectively. Everything else is as for the move point operation.

A complete **length** or **depth control line** can be moved by pressing control (Macintosh) or ctrl+alt (Windows) while the mouse is placed on a segment and the cursor icon changes to or , respectively.

The **stretch net** operation can be applied by placing the mouse anywhere on the control net while the option key (Macintosh) or ctrl+shift (Windows) is pressed, and the mouse icon changes to . The stretch operation moves all the points of the net, while the endpoints of the control lines remain locked in their positions. It can be applied to c-meshes generated from both open and closed control lines, however, the operation is more meaningful when applied to meshes created from open lines.

Any of the length **control lines** can also be **edited**, while in c-mesh edit mode. This is done by placing the mouse on a length line while pressing the shift key and the cursor icon shows . The program enters the curve edit mode as soon as the mouse is clicked. When in curve edit mode all the operations of that environment become available, and the shape of the control curve can be manipulated as described in section 4.13.9. You exit the curve edit mode by double clicking the mouse, which returns you to the c-mesh edit mode. The control net is again shown rubber banded and the c-mesh is regenerated reflecting the change made while under curve edit mode. Another double click of the mouse will exit you from the c-mesh edit mode.
Examples

The direct generation of c-meshes and the manipulation of c-meshes in edit mode essentially represent two different methods that can be used for the creation of meshed objects. In the first case, control lines are created which approximate the desired 3D form. The meshed object is then generated. The chances are that it will not be exactly right the first time. Inspection of the produced form should offer suggestions for adjusting the control lines and/or the parameters used. The meshed object may be undone, the control lines may be corrected, and the object may be generated again. This process may be repeated until a satisfactory form is derived. It is appropriate to work with lower density meshes at the beginning and then make them denser as your form evolves into what you have in mind. Some of the examples in this section were created using this method.

The other method is to generate a meshed object in edit mode and to manipulate its geometry dynamically, until a satisfactory form is obtained. Admittedly, for very complex meshes, this method may not be practical. Quite frequently a combination of the two methods will be most appropriate.

In general, it should be noted that form•Z offers the tools. How they are used to create imaginative forms is “art” for which no manual can be written. We can only hope that the few examples we offer in this section will inspire you to create better forms.

Figure 4.14.18: Freely drawn open control lines were used for the generation of this (abstract) form. The Two Sided option was selected (C-Mesh Options dialog) resulting in a double faced surface solid.

Figure 4.14.19: Closed control lines were used to create a Ring type of a meshed object, which is a true solid. Note how the control lines are “paired” to derive the inside and outside portions of the mesh.
**Figure 4.14.14.20:** The body of the dolphin has been derived as a single solid from the control lines shown. The fins were derived independently, and were then stitched to the body (see Trim & Stitch tool in section 4.17.2).

**Figure 4.14.14.21:** The shoe has been generated in a fashion similar to that illustrated in Figure 4.14.14.19. However, the control lines are not flat but have been manipulated in 3D space before using them for the generation of the mesh.
4.14.15 Formula surfaces

The Formula Surface tool constructs a new controlled smooth object consisting of a surface defined by a mathematical formula.

Formula Surface

Formula surfaces are generated based on mathematical formulas that are defined by the user or chosen from the Pre Defined menu. This tool is useful if a surface needs to follow specific parameters.

The formulas are specified by three parametric functions: \( X(u, v) \), \( Y(u, v) \), and \( Z(u, v) \) where parameters \( u \) and \( v \) are defined in the domains: \( [U_{\text{min}} < u < U_{\text{max}}] \) and \( [V_{\text{min}} < v < V_{\text{max}}] \). These functions must map two dimensional parameter space \((u, v)\) onto three dimensional object space \((x, y, z)\). The minimum and maximum values specify face boundaries in parameter space. These and other options are selected from the Formula Surface Options dialog (Figure 4.14.15.1) that is invoked directly from the tool.

The Formula Surface Options dialog consists of two tabs, Options and Display Resolution. The first tab contains the options that affect the generation of the formula surfaces. The second tab contains the common parameters that dictate how the facets and wires are displayed on the screen.

Figure 4.14.15.1: The Formula Surface Options dialog.
**Parametric Functions:** This group of options specifies the three parametric functions $X(u,v)$, $Y(u,v)$, and $Z(u,v)$. The user may either choose a pre-defined function or type in a new function.

**Domain:** These options determine the domain for the $u$ and $v$ parameters.

- $U_{\text{min}}, U_{\text{max}}$: These values specify the lower and upper limits for parameters $u$ and $v$, respectively.

**Unit Size (Scale):** This value determines the size of the formula surface that will be created. The surface is originally generated in normalized unit space and is then scaled by this value.

**Pre Defined:** This pop up menu contains the names of a number of mathematical functions, as shown in Figure 4.14.15.2. Selecting one of these items enters the respective formula in the $X(u,v)$, $Y(u,v)$, and $Z(u,v)$ fields. This formula will be used to generate a surface next time the Formula Surface tool is applied. Examples of surfaces generated from predefined formulas are shown in Figures 4.14.15.3 and 4.14.15.4.

**Recent:** This pop up menu stores up to ten most recently executed formulas. These can be selected again from this menu to re-executed.

![Figure 4.14.15.2: The Pre Defined pop up menu.](image)

![Figure 4.14.15.3: Examples of predefined formula surfaces: (a) Paraboloid, (b) Single Hyperboloid, (c) Double Hyperboloid, (d) Hyperbolic Paraboloid, (e) Hyperbolic Paraboloid 2, and (f) Waves.](image)
Figure 4.14.15.4: Examples of predefined formula surfaces, shown in wire frame and shaded renderings: (a) Mobius Strip, (b) Catalan, (c) Catenoid Helicoid, (d) Enneper, (e) Henneberg, (f) Monkey Saddle, (g) Whitney Umbrella, and (h) Steinback Screw.
Previewing and editing formula surface objects

When the Formula Surface tool is executed with the Preview option selected in the Formula Surface Options dialog, the Formula Surface Edit dialog, shown in Figure 4.14.15.5, is invoked. This dialog offers the opportunity to preview a formula surface and to experiment with different variations. It contains a standard preview window on the left, the options found in the Formula Surface Options dialog, and a few additional options.

**Update, Automatic:** These button and option are as in the Formula Curve Edit dialog.

Figure 4.14.15.6 shows a formula surface derived with a custom function.

**Figure 4.14.15.5:** The Formula Surface Edit dialog.

**Figure 4.14.15.6:**
Example of formula surface using custom functions
\[ X(u,v)=u\cos(v), \ Y(u,v)=u\sin(v), \ Z(u,v)=2\cos(5u)2^{-(u)} \]
in the domains \( U_{\text{min}},U_{\text{max}} = [0,3] \) and \( V_{\text{min}},V_{\text{max}} = [0,3.14] \).
4.15 Patches

The following four tools that generate and manipulate patches are discussed in this section:

- Patch Derive
- Patch Grow
- Patch Divide
- Patch Attach

A patch may be thought of as a 4 or 3-sided parametric face or as a parametric curve-bounded surface. A number of such patches can be stitched together to produce freely curved surfaces or solids. Such configurations of patches are referred to as patch meshes or patch objects.

The two main advantages of patch modeling over other parametric entities, including NURBS, is that they are by definition bi-directional, which makes them more efficient for surface modeling. They also support variable resolutions in areas where more detail is required. Possibly their only disadvantage is that they are less standard as parametric surfaces and they are less widely supported than NURBS, especially by CAM (computer aided manufacturing) processes.

As implemented in form•Z there are two types of patches: bicubic Bezier and Coons patches, named after the scientists that developed them. Each type has a different control structure. While they both appear the same when initially generated, they behave quite differently when they are edited. Their controls and their names are shown in Figure 4.15.0.1.

![Figure 4.15.0.1: The controls of (a) bicubic Bezier and (b) Coons patches.](image)
When a patch is initially constructed, the outer perimeter of the object retains the color selected for it, while the faces are subdivided and the lines of subdivision are a dimmed version of the object’s color. When the Edit Controls tool is used to click on a patch object, the inherent differences between the a Bezier and Coons patch become visible, as shown in Figure 4.15.0.1.

Bezier patches are defined by a net of sixteen points on each face, which describe two sets of Bezier curves on the surface. Each curve is defined by four control points. The surface generally follows the shape of the defining polygon net, and in general only the four corner points of the net actually lie on the patch. The tangent vectors at the patch corners are controlled both in direction and magnitude by the position of the adjacent points along the edges of the control net. The four interior points (called twists) influence the direction and magnitude of the twist vectors at the corners of the patch.

The Coons patch is a bilinear surface patch that is defined by sixteen vectors: four position vectors at the corners of the patch, the tangent vectors in both directions, and the twist vectors.

Patch objects are derivative structures constructed from other objects, which may be closed surfaces or solid objects. A patch can not be constructed from an open vector line. Typically one patch is constructed from each face of an object. The option to subdivide a patch at generation time or after its initial generation is available. After a patch object has been generated, it can be edited and its geometry can be transformed, using the Edit Controls and Edit Surface tool. These editing operations are discussed at the end of this section. They can be combined with topology manipulating operations that are provided by additional tools, each of which represents a distinct method for modeling with patches, as follows:

• Patch Derive: Sketch a 3D form as a low resolution “cage”, then generate a patch object from it and refine it by applying additional geometric transformations.

• Patch Grow: Starting with an initial patch with open edges, patches are grown from and are attached to its open edges, then to the open edges of the new patches, and so on. This is a method that allows you to build a complete form gradually, a piece at a time.

• Patch Divide: After starting with an initial surface or solid patch object, parts of it are selectively subdivided, as necessary to introduce more detail. Combined with geometric transformations, subdivision of patches can easily turn a cube into an animal head or other organic form.

• Patch Attach: A 3D form is roughly sketched in space with surface patches whose edges are then attached and stitched to each other to produce a potentially complex form.

Needless to say that most frequently these tools are combined and each is used for the partial task it can do best. They are discussed next in this section.
4.15.1 Deriving patches

Patch Derive

This tool is used to generate a patch from a plain facetted object, which can be a closed surface or solid object. It can also be used to derive a patch from an analytic primitive, when it always produces a Bezier patch.

When deriving a patch from a facetted object, this tool can be applied at either the Object or Face topological levels. Selection can be done using either the prepick or postpick method. To generate a patch using the prepick method, you first select the objects or faces with the Pick tool and then, with the Patch Derive tool active, you click the mouse anywhere in the graphic window. To use the postpick method, simply select an object or face with the Patch Derive tool.

The type of patch object that will be created is determined by options selected in the Patch Derive Options dialog (Figure 4.15.1.1), which is invoked directly from the tool. Two types of patches can be derived, as follows:

- **Bezier, Coons**: Depending on which of these two options is selected, a bicubic Bezier or a Coons patch is constructed for each face of the object, respectively. **Bezier** is on by default.

![Image](image_url)

**Figure 4.15.1.2**: Patches generated from a cube with **Subdivide** (1) off and (2) on. (a) The original cubes; (b) **Smoothing** off; **Smoothing Out** on at (c) 0%; and (d) 100%.
**Maintain 4-Sided Patches**: When this option is selected, the object or faces used to construct the patches are decomposed using only 4-sided shapes. If this option is off (default), the faces are decomposed to 4-sided and/or 3-sided shapes (triangulated).

**Subdivide**: When this option is selected, the patches are subdivided by splitting each one with lines that start at its center and end at the midpoints of its boundary edges. Figure 4.15.1.2 illustrates the effect of this option.

**Smoothing**: When this option is off (default), patches are constructed flat on the surfaces of objects. When this option is on, flat patches are first constructed on an object and then the geometric positions of their points are adjusted in a manner that smooths their edges. There are two variations of smoothing, as shown in Figure 4.15.1.3:

- **In**: When this option is on, smoothing occurs toward the inside of the patch object.

- **Equal Portions**: When this option is on, which is only available when **In** is on, the control lines at the tangent points in both the U and V directions are constrained to being equal. This produces a different smooth continuity effect than when these lines have different lengths.

- **Out**: When this option is on, smoothing occurs outside the patch object. Characteristic of this smoothing is that the patch surface passes through the control points, which is not true with **In**.

**Smoothness**: This slider bar controls the degree of smoothness. Examples for values of 0% and 100% are shown in Figure 4.15.1.4.

This **Smooth** operation is similar to that applied by the Q-Subz tool. The slider bar controls the amount of curvature applied to the patch. At 0% there will be less curvature around the centers and edges of the original faces. Higher values will result in more curvature. The difference between patch smoothing and faceted smoothing is that the former offers the ability to move individual points without disrupting the smoothness (see editing operations at the end of this section).
4.15.2 Growing patches

**Patch Grow**

This tool is used to generate patches from open edges (segments) of previously created patches. An edge is open when it is not connected to another patch, such as the edges of the patch derived from a surface object. The dimensions of the new patch are based on the length of the edge from which it is derived or “grown”. The parameters for the newly grown patch are selected in the **Patch Grow Options** dialog (Figure 4.15.2.1), which is invoked directly from the tool. Two types of Patch Grow operations can be selected from this dialog:

**From Edges**: This method grows patches from single edges. It can be meaningfully applied at the Object, Face, Outline, and Segment topological levels, as shown in Figure 4.15.2.2.

**Between Edges**: This method derives patches from two, three, or four edges. Can only be applied at the Segment level (Figure 4.15.2.3).

Both Patch Grow types can generally be applied using either the prepick or the postpick method.

To use the postpick method, with the Patch Grow tool active and **From Edges** on, select an object, face, outline, or segment. A new patch will be grown at each open edge of the entity you select. If **Between Edges** is the active operation, click on two segments. A new patch will be grown starting at one edge and ending at the other.

When using the prepick method with **From Edges** on in the **Patch Grow Options** dialog, with the Pick tool preselect all the entities from whose segments you wish to derive a patch. Then, with the Patch Grow tool active, click anywhere in the graphics window. Patches are generated at all the open edges of the preselected entities.
When using the prepick method with **Between Edges** on, use the Pick tool to preselect two, three, or four segments. Then, with the Patch Grow tool active, click anywhere in the graphics window. Patches are generated between the selected edges. However, to generate a patch between three or four segments, their end points need to coincide. If this condition is not satisfied, the operation is refused. This condition is not required when generating a patch between two segments. Actually, such patches are more meaningful when generated between segments that are apart. The patches are 4-sided, except they are 3-sided when the **Triangular Patches** option is on. When more than four segment are selected, those beyond the four are ignored. The order in which the edges are picked for this operation is not significant. Examples of patches grown between edges are shown in Figure 4.15.2.3.

When growing patches from single segments, they are always equilateral (have equal sides) and the lengths of their sides are equal to the length of the segment they were grown from. Their orientation may be relative to the patch they are grown from or relative to the reference plane. This is determined by selecting one of four options in the **From Edges** group.

**Align Patch Normal**: When this option is selected, the normal of the newly grown patch is aligned with the normal of the patch from which it is grown (Figure 4.15.2.4).

**Perpendicular To Patch**: With this option, the new patch is perpendicular to the patch from which it is grown (Figure 4.15.2.5). It is normally generated on the positive side of the patch from whose edge it is grown. When **option** (Macintosh) or **ctrl+shift** (Windows) is also pressed at the time the edge is selected, the new patch is created on the negative side of the patch it is grown from.

**Parallel To Reference Plane**: When this option is selected, the normal of the new patch is parallel to the normal of the reference plane, which can be any of the Cartesian planes or an arbitrary reference plane. Examples are shown in Figure 4.15.2.6. Note that, if the edge from which the patch is grown has such an orientation that derivation of a patch parallel to the reference plane is impossible, the patch is generated parallel to the patch it is grown from.

**Perpendicular To Reference Plane**: When this option is selected, the normal of the new patch is perpendicular to the normal of the reference plane (Figure 4.15.2.7). The patch is generated towards the positive side of the plane, unless **option** or **ctrl+shift** is pressed, in which case it is generated towards the negative side of the plane.
**Triangular Patches**: When this option is selected, the constructed patches will be 3-sided. By default patches are 4-sided.

**Stitch**: When this option is selected, the new patch is automatically stitched to the edge it is grown from. If this option is off, which is the default, one of the edges of the new patch becomes coincident with the edge it is grown from, but the two are not stitched. There is generally no visible distinction between stitched and non-stitched edges, except when the Show Directions option is on in the Wire Frame Options dialog.

**Smooth**: This option is only available when Stitch is on. When selected, the edges where patches are stitched are also smoothly curved.

**Equal Portions**: When this option is on (which is only available when Smooth is on) the line segments on each side of the tangent control points are equal, which affects the shape of the smoothing.
4.15.3 Dividing patches

While a patch can be subdivided at the time it is derived (by turning on Subdivide in the Patch Derive Options dialog), it can also be further subdivided after its initial generation, using this tool, which also offers additional flexibility. With this tool, any number of subdivisions can be applied and its application may be repeated. However, be aware that the number of patches and facets that subdivision produces multiply fast. If you overdo it, you may produce a patch object that does not fit into your machine’s memory.

The number of subdivisions in each direction is set in the Patch Divide Options dialog (Figure 4.15.3.1). From this dialog you can also select the reverse operation. That is, the Patch Divide tool can be used to both subdivide and undivide a patch.

The Patch Divide tool can be applied at Object, Face, or Segment topological level. Note that subdividing at the Face and Segment levels has some significant implications and should be done very carefully. Subdivision in general is intended to offer more resolution at areas where the model requires more detail. When this is done, portions inside the subdivided area can be moved conveniently to articulate the form. However, subdivision by necessity breaks the stitches at edges, one side of which has been subdivided while the other has not. For example, when a face is subdivided and all its neighboring faces are not, the boundary edges of the face have more control points than its neighbors and the points do not match to be able to remain stitched. Because they are not stitched anymore, moving a point at what appears to be a common and stitched edge will only move it for one of the faces, producing a gap between the faces. This suggests that, when selective subdivision is applied, it is intended to offer more resolution inside the subdivided area and not at its edges. form\textsuperscript{Z} actually offers an option to lock the edge patches when subdividing. When the subdivision is applied at the Object level, the division maintains a uniformity and continuity along all the faces of the object. When applied at the Face level, the division only occurs on the face selected. When subdividing at the Segment level, the division occurs on the two faces that are adjacent to that segment only and only in a direction perpendicular to the segment.

Selection can be done using either the prepick or postpick method. To subdivide a patch using the prepick method, the object, faces, or segments are first selected using the Pick tool, then the Patch Divide tool is activated and the mouse is clicked anywhere in the graphics window. To use the postpick method with the Patch Divide tool active, click on the entity you wish to subdivide.
The following options can be set in the **Patch Divide Options** dialog (Figure 4.15.3.1), which is invoked directly from the tool.

**Divide:** When this option is on (default) the selected entity or entities are subdivided as many times in each direction as the numbers in the following fields indicate.

**Divisions In 1st Direction, Divisions In 2nd Direction:** The numbers entered in these fields represent the number of subdivisions in the two directions of the patch. The defaults are 2.

**Lock Edge Points:** When this option is on, the control points on both sides of the divided edges are locked, and cannot be moved. Locking points prevents fractures along shared edges in a patch surface, when such edges have been unevenly subdivided.

**Undivide:** When this operator is on and the Patch Divide tool is applied to a previously subdivided entity (object, face, or segment), the subdivision is reversed. Note that the **Undo** operation is still available, but **Undivide** is more flexible than undoing. For example, after subdividing a complete object, the **Undo** operation will reverse all the subdivisions. If instead the **Undivide** operation is used, individual subdivisions can be reversed selectively, while other subdivisions remain intact.

Figure 4.15.3.2 illustrates subdivisions at different topological levels.
4.15.4 Attaching patches

While the Patch Grow tool generates patches between two, three, or four existing patches and optionally connects patches that are apart, this tool achieves the same result by moving parts of or even complete patches and attaching them to other patches. This tool resembles the general Attach tool available for faceted objects, but applies to patch objects.

The options for the Patch Attach tool are set in the Patch Attach Options dialog (Figure 4.15.4.1) invoked from the tool. This dialog lets you select one of four different operations that can be executed with the Patch Attach tool, namely, Attach, Detach, Stitch, and Unstitch. Each one is executed slightly differently and/or has its own options, as follows:

**Attach**: This operation attaches an open edge of a patch to an open edge of another patch. It is always executed at the Segment level (it ignores the current topological level) and requires the selection of two segments. Note that normally the two segments make sense if they are from two different patches. However, they are also allowed to be from the same patch, which frequently is not meaningful, but it is quite useful in certain cases. For example, when a four-sided patch takes a cylindrical shape and two opposite edges need to coincide, attaching them achieves the desired result.

For the edges of two patches that are apart to be made to coincide, something has to move. This motion is always applied to the patch whose segment is selected first and the following options are available. They are illustrated in Figure 4.15.4.2.

**Figure 4.15.4.1**: The Patch Attach Options dialog.

**Figure 4.15.4.2**: Patch Attach operations:
(a) The original patches; (b) Move Edge;
(c) Move Patch; and (d) Move Object.
With Scale (1) off and (2) on.
Move Edge: When this option is on, the edge selected first is moved and made coincident with the edge selected second (Figure 4.15.4.2(b)). The remaining part of the patch stays in its original position.

Move Patch: When this option is on, the whole patch whose edge was selected first is moved to make its edge coincident with the edge selected second. If the first object has more than one patch, those that do not share edges with the patch that moves remain in their original position (Figure 4.15.4.2(c)).

Move Object: When this option is on, the whole object whose edge was selected first is moved so that the selected edge becomes coincident with the edge selected second (Figure 4.15.4.2(d)).

Align Normals: This option affects only Move Patch and Move Object. It is dimmed and inactive when Move Edge is selected. When on, the normal of the patch selected first is aligned to the normal of the patch selected second. When off, the normal of the first patch preserves the direction of its normal. Examples are shown in Figure 4.15.4.3.

Scale: When this check box is selected, the edge picked first is scaled up or down to match the length of the edge picked second. When Move Patch or Move Object is the Patch Attach variation being executed, the whole patch or object is scaled by the same factor, respectively. This option is on by default. It is illustrated in Figure 4.15.4.2(2).

Stitch: When this check box is selected, the attached edges are also stitched. Whenever necessary to complete the stitching, the first edge geometry is adapted to the geometry of the second edge, to maintain continuity between the patches. Recall that, when two edges are stitched, they become a double edge that behaves like one in all operations.
**Detach**: This operation can be applied at the Object, Face, or Segment level and can be used in postpick or prepick mode. It produces a result that is the reverse of the **Attach-Stitch** operation. That is, at the level it is applied it completely breaks the stitches. Note, however, that deleting an interior edge of a patch configuration may not be meaningful, unless the segment is the only edge that joins two configurations of patches, or a continuous sequence of segments is selected, which divides the patch. Applying Detach at the Object level, completely explodes the object. That is, all the patches in the object become separate pieces and even independent objects if the respective option is on in the **Status Of Objects** tab. Applying Detach at the Face level only breaks the stitches of the face, which becomes separated. Examples of these variations are shown in Figure 4.15.4.4.

**Stitch**: This operation makes one edge from two edges that were previously unstitched. It can be applied at the Object, Face, or Segment level in both prepick and postpick mode. Note that, for this operation to take effect, the edge to be stitched must have previously been unstitched. Thus this operation can be thought of as reversing the act of unstitching. It can not be used to stitch two touching patches, which can be done with the **Attach-Stitch** operation.

**Unstitch**: This operation is a partial detachment and can be applied at the Object, Face, or Segment level. That is, it breaks the stitches of the edge, but preserves the coincidence of the control points at their ends. Note that this is quite different from Detach, which also breaks the coincidence of the points. Each offers distinct possibilities for editing operations that manipulate the shape of a patch object, especially the way other patches can be grown from these edges. Examples of the Detach and Unstitch operations are illustrated in Figure 4.15.4.5.
Editing patches

There are two tools with which patch objects can be edited: Edit Controls and Edit Surface. The former applies editing through the controls of the patch parameters, while the latter applies editing transformations directly to the surface of the patch object.

To edit through the controls, with the Edit Controls tool click on the patch object. This invokes the **Edit Controls Options** dialog (Figure 4.15.4.6) and displays the patch controls. Recall that these are different for a Bezier and a Coons patch, as shown in Figure 4.15.4.7.

Once in edit mode, the following operations are available:

- Clicking on a control point selects that point.
- Clicking on a control point while pressing control (Macintosh) or ctrl+alt (Windows) selects that point and all the adjacent points. Clicking likewise on a segment, selects the segment and all the sequence of segments across the patch.
- Clicking on a segment while pressing command (Macintosh) or ctrl (Windows) selects the whole patch.

The iconic buttons in the **Action** section represent operations (move, rotate, independent scale, and uniform scale) that determine how the graphic input will be interpreted. These are discussed in more detail below. Options Reference Plane, Local Normal, and Average Normal are as for the nurbz editing operations, discussed in the previous section.

The **Selection** section contains two pop up menus: Locked and Point Type. The former has two items, on and off, which lock or unlock a selected point, respectively. The latter contains six items, which are six types of points. Selecting one of these items works like a command and is executed once and immediately.

Recall that the edit points of a Bezier patch are of three types: corner, tangent, and twist (see the beginning of this section). The **Locked** operations can be applied to all types of points. The **Point Type** assignments can not be applied to the twist points and they are dimmed when such a point is selected. For the other types of points they work as follows:
**Corner**: When this item is selected, the picked controls lose any smoothing that may have been assigned to them.

**Smooth Corner**: When this item is selected the respective controls become of a mixed smooth and corner character.

**Smooth Out, Smooth Out Equal, Smooth In, Smooth In Equal**: When one of these items is selected, the respective type of smoothing is applied to the selected controls.

When one of the icons in the **Action** group of the **Edit Controls Options** dialog is selected, the respective geometric transformation is applied to the selected controls. Note however that, as with nurbs editing, the execution of these operations varies. Move can be executed using either the click-and-drag or the click-and-click method. The others always use click-and-click.

**Move**: With the click-and-drag method, click on a control and move the mouse while you keep its button pressed. With the click-and-click method, click on a selected control, release the mouse button, move it to the desired position, and click again.

**Rotate**: Using the click-and-click, which is the only method available, click on a selected control (1), click to establish the center of the rotation (2), click to start the rotation (3), and click to establish the angle of rotation (4). An example is shown in Figure 4.15.4.8.

**Independent Scale, Uniform Scale**: Click on a selected control (1), click to establish the basis of the scaling operation (2), click to start the scaling (3), and click to establish the scaling factor (4). An example is shown in Figure 4.15.4.9, where four preselected controls are scaled with a single Uniform Scale operation. While both examples illustrate transformations applied to Bezier patches, transformations on controls of Coons patches work the same way.

Direct surface editing with the Edit Surface tool is similar to editing the surface of nurbs objects (section 4.14.2). Clicking on a point on the surface of the patch with the Edit Surface tool, invokes surface editing and the **Edit Surface Options** dialog, which is a simpler version of the **Edit Controls Options** dialog. Moving the point is the only operation available. As the mouse moves, the surface follows the motion. As with nurbs, the point can be moved relative to the **Reference Plane** or along the **Surface Normal**.
4.16 Metaformz

As also discussed in subsection 4.1.12, the metaformz are parametric (or controlled) objects that have the ability to blend smoothly when they overlap. A metaform has a region of influence, determined by its radius, which is controlled by the user. This region of influence interacts with other metaformz to create the ultimate surface that blends a number of metaformz together. This surface is formally called the implicit surface of the metaformz.

The following two metaformz tools are discussed in this section:

- Metaformz Derive/Edit
- Metaformz Evaluate

As implemented in form•Z, metaformz include the more commonly known metaballs, which are generated as primitives by a tool discussed in subsection 4.1.12. The implicit surface algorithms have also been extended to 3D shapes other than spheres, which significantly enriches the practicality of the metaformz.

There are four types of metaballs that can be created both directly and as derivative objects. They are the ball, the stretched ball, the ellipsoid, and the stretched ellipsoid. When generated directly, they are created as individual objects (see Figure 4.16.0.1, which repeats 4.1.12.1 shown in section 4.1.12). When they are generated as derivative objects, they are created as chains of individual objects that are arranged on paths (Figure 4.16.0.2).

Figure 4.16.0.1: Directly generated metaballs: (a) unevaluated and (b) evaluated.

Figure 4.16.0.2: Chains of metaballs generated as derivative objects: (a) original objects, (b) unevaluated, and (c) evaluated.
All the other metaformz can only be generated as derivative objects, from previously created common formZ objects, which can be open or closed surface objects or solids. Five types of metaformz (in addition to the metaballs) can be derived: the tube, the conic tube, the torus, the sheet, and the polyhedron. Most of them are illustrated in Figure 4.16.0.3.

Individual metaformz may be unevaluated, which they always are when they are generated, or evaluated. The former are displayed as wire frames that represent their parameters and cannot be rendered. The latter are displayed with their implicit surfaces and can be rendered. Figures 4.16.0.1 through 4.16.0.3 illustrate both unevaluated and evaluated metaformz.

The unevaluated representations of metaformz consist of a pair of similar sets of wires, one of which is larger than the other. The smaller set is displayed in the color of the object. The larger is in the ghost color, which is gray by default. The larger is also affected by the Hide Ghosted item in the Wire Frame Options dialog. When it is selected, the larger (ghosted) set of wires of the metaformz representation is not displayed.

A major factor (parameter) when metaformz are evaluated is the Threshold (set in the Metaformz Evaluation Parameters dialog, Figure 4.16.2.2), which can take a value from 0.1 to 99% inclusive. The larger this value, the "fatter" a metaform becomes when it is evaluated.
Implied in this is that the metaformz do not have an exact size, but their size is a range, and the final size is determined only when the evaluation parameters, most notably the **Threshold**, are specified. Unevaluated metaformz reflect this range. The smaller set of wires represents the size of a metaform when evaluated with a threshold value of 50%. The larger set of wires represent the size when evaluated with a threshold value of 99%.

Another significant parameter that affects the evaluation of the metaformz is the **Weight**. This parameter is set when the metaformz are first generated (either directly or as derivative objects) and can be changed using the Metaformz Derive/Edit tool. The normal range of the weight value is from -5.0 to +5.0. However, the program will allow you to enter values beyond this range, if exaggerated effects are desirable. Of major significance is the sign of the weight value. A positive value has an additive effect while a negative value has a subtractive effect. Recall that this was illustrated in Figure 4.1.12.3, early in this chapter.

**How to derive and evaluate metaformz**

- You first create unevaluated metaformz, either using the Metaballs tool or the Metaformz Derive/Edit tool. The former is used to create spherical metaformz directly, as discussed in subsection 4.1.12. The latter is used to derive metaformz from other common objects. The initial weight value of the metaformz is set at the time they are created.

- As soon as the metaformz are generated they are displayed as wireframes, which is their unevaluated representation. They can be moved, rotated or resized.

- After they are created, the metaformz can be edited and their parameters can be changed. This is done with the Metaformz Derive/Edit tool, by clicking on the wire (unevaluated) representation of a metaform. The parameters of metaformz can also be changed using Query.

- Individual metaformz can be evaluated using the Metaformz Evaluate tool. When you use this tool and click on an unevaluated metaform, a dialog is invoked in which you set the threshold value and the mesh resolution.

- To evaluate groups of metaformz, **you first need to group them** using the Group tool. Groups can also be constructed from within the Objects palette. When you are grouping the unevaluated metaformz they become evaluated.

- A group of evaluated metaformz can be re-evaluated using the Metaformz Evaluate tool again. With this tool you click on the evaluated mesh of the metaformz.

- The parameters of individual metaformz can be edited even after they have been evaluated as members of a group. You do this with the Metaformz Derive/Edit tool. With this tool, you click on the wire representation of the unevaluated metaform.

These operations are discussed in detail in the following subsections.
4.16.1 Deriving metaformz from other objects

Metaformz Derive/Edit

This tool is used to derive metaformz from other, previously created form•Z objects. The type of metaform to be created and its parameters are selected from the Metaformz Parameters dialog that is invoked directly from this tool. It is shown in Figure 4.16.1.1.

This tool can also be used to edit and change the parameters of a previously created metaform. The editing operations are discussed in section 4.16.3.

Metaform Type: This pop up menu consists of three groups of options. The first contains the single item None, the second group the four metaball types, and the third the remaining metaformz types.

Figure 4.16.1.1: (a) The Metaformz Parameters dialog and (b) its Metaform Type pop up menu.

None: This item is used to restore the object to its original state before the metaformz parameters were assigned to it. This operation cannot be applied to metaballs created directly with the Metaballs tool.

The metaball types in the second group of the pop up menu are as discussed in the previous sub-section. However, when they are derived from other form•Z objects, they are generated as chains of metaballs. For example, generating balls from an open or closed vector line by the default option, one ball is created at each point of the vector line. These balls are linked (chained) as a single object and cannot be edited individually. When one of the metaballs in the second group of types is selected, the Chain options box is also activated. Otherwise it remains dimmed.

Above the Chain box is the Parameters box, which also contains some options. Some of these options may be dimmed and inactive. Which options are active depends on the metaform type currently selected in the pop up menu.
Any type of form-Z object can be used to derive metaformz. However, not all types of metaformz can be generated from all types of objects. The table in Figure 4.16.1.2 summarizes what types of metaformz can be derived from what types of objects. It also summarizes which of the Radius parameters each type of metaform requires.

<table>
<thead>
<tr>
<th></th>
<th>ball</th>
<th>stretched ball</th>
<th>ellipsoid</th>
<th>stretched ellipsoid</th>
<th>tube</th>
<th>conic tube</th>
<th>sheet</th>
<th>polyhedron</th>
</tr>
</thead>
<tbody>
<tr>
<td>open vector line:</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>closed shape:</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>-</td>
</tr>
<tr>
<td>meshed surface:</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>√</td>
<td>-</td>
</tr>
<tr>
<td>surface solid:</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>√</td>
<td>-</td>
</tr>
<tr>
<td>solid:</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>√</td>
</tr>
</tbody>
</table>

Figure 4.16.1.2: Types of metaformz, from which object types they can be derived, and which radius parameters they require.

**Ball, Stretched Ball:** These are chains of spherical shapes. The balls are generated from a single uniform radius; the stretched balls are generated from three distinct radii, corresponding to their X, Y, and Z dimensions. When one of these types is selected, the options in the Chain box become active (Figure 4.16.1.3) and are used to derive a number of variations.

The balls can be arranged on an open or closed vector line using one of three available methods, which are mutually exclusive:

**At Existing Points:** When this method, which is the default, is selected, the chain is generated by placing one ball at each point of the vector line.

**By # Of Primitives:** When this method is selected, \( n \) evenly distributed balls are placed on the line. The value \( n \) is an integer number entered in the numeric field of the option.
**By Distance**: When this method is selected, the balls are placed at a distance \( d \) from each other, where \( d \) is the value entered in the numeric field of the option. How many balls are arranged for the same \( d \) value depends on the length of the vector line.

**Scale Radius**: When this option is selected, the scaling factor in its numeric field is used to scale the balls as they are arranged from one end of the vector to the other. The scaling factor is applied to the last ball in the arrangement and the balls between the two ends are scaled proportionally. When the vector line is a closed shape, the scaling factor is applied to a ball half way in the arrangement and the intermediate balls are again scaled proportionally.

**Scale Weight**: When this option is selected, the scaling factor in its numeric field is applied to the weights of the arranged balls, as above.

**Random %**: When this option is selected, a random deviation is applied to the radius (or radii) of the metaballs in the chain, which makes them look different in size. The range of the deviation is controlled by the % value entered in its numeric field.

**Ellipsoid, Stretched Ellipsoid**: These are chains of elliptical spheres. The ellipsoids are generated from a single radius parameter; the stretched ellipsoids are generated from two different radii. When one of them is selected in the types menu, some of the options in the Chain box are activated, as shown in Figure 4.16.1.4.

Note that there is only one method for generating chains of ellipsoids: **At Existing Points**. They are generated between pairs of consecutive points. All the other options on the right of the box apply to the ellipsoids as for the balls.

The third group of items in the **Metaform Type** pop out menu contains five types of metaformz that can be derived from common form-Z objects. Even though there are only five types, a vast variety of metaform shapes can be generated, especially with the latter two types.

**Tube**: This type of metaform can be created from an open or closed vector line consisting of one segment or more. As the name implies, it is a tube-like shape of uniform width. It is generated from a single radius. Examples are shown in Figure 4.16.1.5(a).

---

**Figure 4.16.1.4**: The Chain parameters for the ellipsoid and stretched ellipsoid types.

**Figure 4.16.1.5**: (a) Tubes and (b) conic tubes generated from a single segment, an open, and a closed vector line.
**Conic Tube:** This type is similar to the tube, except that it is generated from two radii and its width is not uniform. Examples are shown in Figure 4.16.1.5(b).

**Sheet:** This type of metaform is generated from a closed shape, a two sided surface (surface solid), or a meshed surface, which can be one or two sided. Closed shapes may convex, concave, and may or may not have holes. It is generated from one radius parameter. Examples are shown in Figure 4.16.1.6.

**Polyhedron:** This type of metaform is generated from a solid object, which may be convex, concave, and may or may not have holes. It is generated from one radius parameter. Examples are shown in Figure 4.16.1.7.

**Weight:** This slider bar controls the intensity with which a metaform is blended and whether it will have an additive or subtractive affect. This parameter works as for the metaballs (see previous section).

Note that, on its right, the Parameters box contains three parameters labeled Center X, Y, and Z. These are only used when editing a meta-ball which was previously generated directly and are discussed in section 4.16.3.

---

*Figure 4.16.1.6:* Sheets generated from closed shapes without and with holes.

*Figure 4.16.1.7:* Polyhedra generated from solids without and with holes.
4.16.2 Evaluating metaformz

When metaformz are initially generated, either directly or from other objects, they are unevaluated. They can subsequently be evaluated individually or in groups. To evaluate them as groups, the individual metaformz first need to be grouped.

Metaformz Evaluate

This tool is used to evaluate either individual or groups of metaformz. It invokes the Metaformz Evaluate Options dialog (Figure 4.16.2.1) whose options determine when the evaluation of the metaformz occurs. The specific evaluation parameters are set in the Metaformz Evaluation Parameters dialog, invoked every time metaformz are evaluated.

**Automatic Evaluation:** When this option is selected, a metaform or group of metaformz is automatically evaluated or re-evaluated every time one of its parameters is changed, through editing or graphic input, such as when it is moved to a new location.

**Prompt For Evaluation:** When this option is selected, each time a parameter of a metaform is changed, a prompt is posted, asking the user whether re-evaluation is desired.

**Evaluate Only When Selected:** When this option is on, the metaform is evaluated only when it is selected with the Metaformz Evaluate tool.

To evaluate an unevaluated metaform for the first time, click on it with the Metaformz Evaluate tool. This invokes the Metaformz Evaluation Parameters dialog (Figure 4.16.2.2). When using the prepick method, you select any number of unevaluated metaformz with the Pick tool and then with the Evaluate tool you click anywhere in the window. The Metaformz Evaluation Parameters dialog is invoked for each preselected metaform. To re-evaluate previously evaluated metaformz, the operation is executed the same way, except that you now select evaluated metaformz. Thus this tool can be used both for the initial evaluation and also as an editing tool to re-evaluate a metaform, after changing some of the evaluation parameters.

The practical value of the metaformz is, of course, when a number of them are evaluated together to produce smoothly blended surfaces. Before this can be done, the metaformz that will be blended need to be grouped together. This is done in the usual manner, using the Group tool. Grouping operations can also be applied directly in the Objects palette, as follows:

- To create a group, click under the last item in the palette.
- To place a metaform or a group of metaformz into a group, drag it into the name of the group.
- To remove a metaform or a group of metaformz from a group, drag it outside that group.
Note that regular form-Z objects and metaformz cannot be grouped together. If such an attempt is made, an error message is posted. Also note that the metaformz groups are slightly different than the regular groups. After they are evaluated, they include the evaluated implicit surface as a special member of the group, together with the parametric (unevaluated) definitions of the individual metaformz. The timing of the evaluation depends on the option selected from the Metaformz Evaluate Options dialog. If Automatic Evaluation is selected, the group is evaluated as soon as the grouping occurs. If another option is selected, evaluation is postponed and occurs when it is requested.

The evaluation of the metaformz is affected by their individual parameters as well as by specific evaluation parameters, which are selected from the Metaformz Evaluation Parameters dialog, which is invoked every time an evaluation or re-evaluation is executed. This dialog is shown in Figure 4.16.2.2.

**Threshold**: This slider bar parameter controls the extent of a metaform object. Its range is between 0.1 and 99%.

**Metaform Operation**: This pair of options specifies how the metaformz will be joined. When Blend is selected, the metaformz blend smoothly into each other and may interact even when they do not intersect, provided they have sufficiently large radii. When Join is selected, the metaformz merge only when they intersect. The purpose of the Join option is to regulate the blending of hierarchical groupings, as illustrated in the following example.

In Figure 4.16.2.3, metaballs are arranged to construct a hand (a). They are grouped at two levels. At the lower level the balls that form the fingers are grouped. At the next level the finger groups are grouped. When evaluated with the Blend option, the result is as shown in (b); that is two of the fingers blend. If the same groups are evaluated with the Join option, the fingers remain separate as is appropriate for this model.

![Metaformz Evaluation Parameters](image)

**Figure 4.16.2.2**: The Metaformz Evaluation Parameters dialog.

![Figure 4.16.2.3](image)

**Figure 4.16.2.3**: (a) Metaballs evaluated with (b) the Blend and (c) the Join options.
**Generate Mesh:** When this option is selected, the metaformz are evaluated as facetted objects, according to the resolution set in the following numeric fields.

**Mesh Resolution:** The numeric parameters in this group specify the size of the mesh tiles generated for the implicit surface. They can take independent values or they can be locked to the same value, by turning the lock option on.

**Optimize Surface:** When this option is on, an additional pass is made on the metaformz mesh to reduce creases in the mesh, such that the object will render smoother.

**Generate Nurbz:** When this option is selected, the metaformz are evaluated as smooth objects. For each object, the evaluation process generates many nurbz surfaces, which may or may not be joined together to form a single smooth object, depending on the status of the **Stitch Surfaces** option.

**Accuracy:** This slider bar parameter controls the density of the points obtained when sampling metaformz. When this parameter increases, more sample points are derived, resulting in more nurbz surfaces and a better representation of the metaform. When the slider marker moves close to the left end of the slider, fewer sampling points are obtained, which may not be enough for the creation of nurbz surfaces. In this case, the tool automatically and incrementally adjusts the accuracy value and samples the points until it is able to create nurbz surfaces. As it does this, it actually moves the slider marker and displays it in the new position. Thus, the next time the dialog is invoked, the slider marker is positioned at the new position on the bar. The nurbz surfaces thus created may not accurately represent the metaform. For more accurate nurbz surfaces, the slider marker can be pushed closer to the right end of the bar. However, more accurate surfaces come at the cost of more processing time.

**Number Of Optimizations:** This option is used to specify the number of optimizations to be performed in each surface fitting iteration and it controls the quality of the surfaces generated. The higher the value, the better will be the fit and the slower the process. Recommended range for this value is between 1 and 4. The default value is 4.

**Max Number Of Iterations:** The surface fitting process performs a number of iterative steps in order to meet the specified **Error Tolerance** (see below). This option can be used to stop the process by specifying the maximum number of iterations. The surface fitting process will stop after performing this number of iterations even if the error tolerance is not met. Range for this value is 1 through 4. The default is 1.

**Error Tolerance:** This option can be used to specify the maximum error allowed while fitting the nurbz surfaces to the metaformz surface or to the form-Z mesh. The error is measured as the root mean square distance (average of all the distances between each point and the nurbz surfaces) between the points or the mesh vertices and the nurbz surfaces.
The surface fitting process stops when the error tolerance is met or it has performed the maximum number of iterations specified by the **Max Number Of Iterations** criterion. A lower error tolerance gives a better fit to the data by producing more nurbs surfaces at the cost of longer processing times. If the slider is placed at the rightmost end of the bar, it means that the maximum error allowed is 0.1 inches. Higher percentage values can be typed in the percentage box next to the slider bar to allow error tolerances higher than 0.1. The slider is placed at 50% by default, which means the maximum error allowed is 0.05 inches.

**Clean Mesh Tolerance**: Thin long triangular faces in the mesh can cause the nurbs surface fitting process to fail. This parameter can be used to delete such faces before the fitting procedure starts. The slider bar is mapped on to a float value between 0.0 and 0.03. That is, when the slider is at 100%, the value of this parameter is 0.03; when at 50%, the value is 0.015, and so on. This value represents the ratio of the shortest edge of the triangular face to its perimeter (sum of all three edges). Faces with a ratio smaller than this value are eliminated. Note that a 0.0 value corresponds to a triangular face one edge of which has 0 length or coincident end points. A 0.03 value corresponds to a triangle the shortest edge of which is 1/30 of its perimeter. The slider is placed at the leftmost end of the bar by default.

**Fairness Tolerance**: The final nurbs surfaces may have unwanted wiggles in order to represent the data as faithfully as possible. This parameter helps to control the amount of wiggles on the surfaces. Usually noisy data produces surfaces with more wiggles that are hard to eliminate. There is no linear relationship between this slider and the quality of surfaces; that is, higher values may not necessarily yield better results. This is a parameter that has to be set by experimentation. In some cases, the wiggles may persist for any value of this parameter because of the nature of the data and the other parameter settings. The slider is placed at the rightmost end of the bar by default.

**Preserve Boundary**: When this option is on, boundaries are recovered more accurately at the cost of more processing time and more nurbs surfaces.

**Split At Discontinuities**: When this option is on, the nurbs surfaces are split at G1 discontinuities. Splitting produces more nurbs surfaces. If not split, a lesser number of nurbs surfaces will be created, but subsequent operations like parallel, Booleans may fail.

**Stitch Surfaces**: When this option is on, the nurbs surfaces are stitched together.

An example of a meraform generated as a smooth nurbs object is shown in Figure 4.16.2.4.

---

**Figure 4.16.2.4**: A bone as (a) an unevaluated metaform, (b) evaluated as a facetted object, and (c) evaluated as a smooth nurbs object.
4.16.3 Editing metaformz

Editing can be applied to both the individual parameters of metaformz and to the evaluation parameters of blended groups of metaformz. The former are executed with the Metaformz Derive/Edit tool; the latter with the Metaformz Evaluate tool.

The individual parameters of both metaballs generated directly and metaformz generated as derivative objects, can be edited. To edit a metaform, click on its wire representation with the Metaformz Derive/Edit tool. This invokes the Metaformz Parameters dialog, which displays the parameters used for the generation of the metaform. Change the parameters you want and click on OK. If the Automatic Evaluation option is on (Metaformz Evaluate Options dialog), the metaform is regenerated using the new parameters. When using the prepick method, with the Pick tool select any number of metaformz. Then, with the Metaformz Derive/Edit tool click anywhere in the window. The program will parade through the metaformz you preselected and, for each, it will invoke the Metaformz Parameters dialog. After the last metaform is edited, all will be regenerated using the new parameters.

When editing with the Metaformz Derive/Edit tool, unevaluated metaformz will remain unevaluated but their graphic representation will be regenerated to reflect the new parameters. When editing an evaluated metaform and Automatic Evaluation is on, the metaform will be re-evaluated. When editing the individual parameters of a metaform that is already blended into an evaluated group of metaformz, the complete group will be re-evaluated.

When you edit metaballs that were generated directly, you can also change their position. The Metaformz Parameters dialog, when invoked for one of these metaballs, displays the position of their centroid in the Center X, Center Y, and Center Z fields. Changing one or more of these values moves the metaball to a new position.

Using the Metaformz Evaluate tool you can change the evaluation parameters of a group of evaluated metaformz. With the tool active click on the group, which invokes the Metaformz Evaluation Parameters dialog. You can change any of the parameters it contains. As soon as you click on OK to exit the dialog, if Automatic Evaluation is on, the complete metaformz group is re-evaluated.

Note that, when you use the Metaformz Derive/Edit tool to edit the parameters of individual metaformz, you click on the wire representation of the metaform. When you use the Metaformz Evaluate tool to edit the evaluation parameters of an evaluated metaform or group of metaformz, you can click on either the wire representation or on the meshed surface of the metaformz. If with the Metaformz Derive/Edit tool you click on the meshed surface, a warning message is posted, which also asks you whether you really want to edit the evaluation parameters.
4.16.4 Examples

The two examples shown here range from two simple bones constructed with five metaformz (Figure 4.16.4.1), to a significantly more complex dinosaur head (Figure 4.16.4.2). We make no attempt to show dimensions or to offer instructions about how these models are constructed. In general, when you use metaformz to derive a composite implicit surface, you usually work visually as an artist would.

You create your unevaluated primitives at some initial size and with some initial parameters. You may then move them, resize them, or edit them and change their parameters. At any time you can evaluate them, look at the form you generated and, if not satisfactory, make more adjustments. This cycle continues until you are happy with the form you have generated. This is exactly the process that was used for modeling the dinosaur head.

![Figure 4.16.4.1: Bones modeled with metaformz.](image)

![Figure 4.16.4.2: A dinosaur head constructed with metaballs.](image)
4.17 Booleans, trim, stitch and sections

The following tools are discussed in this section:

- Union
- Intersection
- Difference
- Boolean Split
- Trim/Split
- Line of Intersection
- Stitch
- Section
- Contours
- Cage

The first four tools above are the Boolean operations. The next is a tool that may be thought of as the Boolean for surfaces. However, it is significantly different in its ability to automatically calculate the pieces to be returned as a result of the operation.

A common characteristic of all except one of the tools in this section is that they start by deriving the line of intersection of two entities. The sixth tool above does only this. That is, it produces a line of intersection as its result.

The seventh tool above (Stitch) does not calculate an intersection but links reversely coincident edges of surfaces. While it is most commonly used after a Trim operation, it is also available as an independent operation.

The next tool derives 2D or 3D sections. The former resemble the lines of intersection, but they are always produced by flat surfaces. Similarly the latter resemble the Difference or Trim/Stitch operations, except that the second operand is always a flat surface. The next tool (Contours) is essentially a set of 2D sections that are generated at constant intervals in any direction, which is determined by the active reference plane.

The last tool derives a Boolean Intersection of two volumes generated by extruding the 2D projections of an object in two (of the three) Cartesian orientations. It is intended to be used for producing a low resolution object (called cage) that may take the place of a typically high resolution object during manipulations of a scene.

All these operation can be applied to either facetted or smooth solids, or even a combination of the two. If one of the objects involved in a Boolean operation is smooth, the result is also smooth.
4.17.1 Boolean operations

In form·Z, the Boolean operators may be applied to either 2D closed surface objects (2D shapes) or to well formed 3D solids. However, the two types may not be mixed. The system automatically recognizes the type of the operands and applies the appropriate 2D or 3D operation.

**Union**

For 3D solids, the Union tool creates an object that consists of the volumes contained in either one or both of the two objects to which it is applied. For 2D surface objects, replace the word “volume,” in the above definition, with the word “area.” The union always produces an object as a result. Whenever the two objects to which it is applied do not intersect, the result consists of the two original objects. The Union operator, as it applies to 2D surface objects and to 3D solids, is illustrated in Figures 4.17.1.1 and 4.17.1.2, respectively.

**Intersection**

The Intersection tool creates an object that consists of the volume/area which is common to both objects involved in the operation. The intersection returns a null result if the two objects do not intersect. The Intersection operator as it applies to 2D surface objects and to 3D solids is illustrated in Figures 4.17.1.3 and 4.17.1.4, respectively.
Difference

The difference of objects A minus B returns the volume/area that is in A but not in B. The difference of objects A minus B returns A, whenever A and B do not intersect. The difference of objects A minus B returns a null result, whenever A is completely contained in B. The Difference operator as it applies to both surface and solid objects, is illustrated in Figures 4.17.1.5 and 4.17.1.6.

Figure 4.17.1.5: The difference operation (a) A-B and (b) B-A applied to a pair of 2D surface objects.

Figure 4.17.1.6: The difference operation (a) A-B and (b) B-A applied to a pair of 3D solids.
Boolean Split

This tool is a composite Boolean operator that can be applied to both 2D surface and 3D solid objects, but all objects selected must be of the same type (2D surface or solid). It can be applied as a one way or a two way operation, which is selected from the Boolean Split Options dialog (Figure 4.17.1.7) that can be invoked directly from the Boolean Split tool.

When **One Way** is selected (default) the first object is split by the second. That is, the result is the difference of the first object minus the second, plus the intersection of the two objects.

When **Two Way** is selected the two objects split each other. The result is the difference of the first object from the second, the difference of the second from the first, and the intersection of the two objects. Examples are shown in Figure 4.17.1.8.

![Boolean Split Options dialog](image)

**Figure 4.17.1.7:** The Boolean Split Options dialog.

![Examples of Boolean Split](image)

**Figure 4.17.1.8:** Examples of the two way Boolean Split operator: (a) Splitting two 2D surface objects. (b) Splitting a cuboid and a faceted sphere. (c) Splitting a smooth sphere and a cuboid. The results have been moved and are shown in both wire frame and hidden line plots.
Executing Boolean operations

Either the prepick or postpick method can be used with the Boolean operators. With the postpick method, a Boolean tool is selected first and then two objects are picked. The execution of the operation proceeds immediately. The order in which the objects are picked is significant only for the Difference and the One Way Split operations, where the second object is subtracted from (or splits) the first. When picking the objects, the topological level is ignored. The system automatically applies a pick at the Object level.

With the prepick method, any number of objects may be picked first, followed by the selection of a Boolean icon, followed by a mouse click anywhere on the graphics window, which initiates the execution of the operation. The Boolean operations are by their nature binary. If the prepick method is used to select more than two operands, the order in which they are picked is quite significant. The system will apply the operation to the two objects that were picked first and then apply the operation to the result and the third object, and so on until all picked objects are exhausted. Should, at some point, a null result be returned, the execution of the operation is terminated, and the objects that remain picked are ignored. The behavior of prepicked objects for each of the three Boolean operators is illustrated in Figure 4.17.1.9. Any entities picked at a topological level other than the Object level are also ignored. When such cases are encountered, the system looks for the next object on the list and uses that as an operand in the next operation. When using the prepick method with the Split operator, it will be applied only to the first two objects in the pick list. Any additional objects picked will be ignored.

Figure 4.17.1.9: Executing Boolean operations by the prepick method.
(a) Eleven objects have been drawn and prepicked, in the order they overlap, starting with the circle.
(b) The union operation is applied, resulting to a single “chain” shape.
(c) When the intersection is applied, it produces a null result and stops after the second step.
(d) When the difference is applied, it is executed all the way, but only the first step affects the result.
The objects that are involved in a Boolean operation may be \textit{ghosted} (default), \textit{deleted}, or \textit{kept} in their original state, after the execution of the operation. If a Boolean operation produces more than one volume, those volumes may be stored as \textit{separate} objects (default) or all volumes may be returned as a \textit{single} object. Both options may be set in the \textit{Status Of Objects} dialog, that can be invoked from the \textit{Options} menu or from the Boolean tool dialogs.

The latter option should be given particular consideration whenever the prepick method is used to select more than two objects. If the \textit{Create One Object Per Volume} option is selected and a Boolean operation produces more than one volume at some point of its execution, only one of these volumes will be picked as an operand for the next application of the operation. To cause all volumes to be picked as operands, the \textit{Store All Volumes As One Object} option should be selected. Figure 4.17.1.10 illustrates the two options.

The Boolean operations require \textit{planarity} for both the surface and the faceted solid objects. If the operands are 3D solids, they are also required to be well-formed. If inappropriate conditions are encountered, the system beeps, issues an error message, and cancels the execution of the Boolean operation. Examples of mechanical parts constructed by the use of the Boolean operator, are shown in Figures 4.17.1.11 and 4.17.1.12.

\textbf{Figure 4.17.1.10:}

(a) Four linear shapes subtracted from a decagon.
(b) With \textit{Create One Object Per Volume} on, individual objects result, only one of which is picked as an operand by the next operation.
(c) With \textit{Store All Volumes As One Object} on, the intended result is achieved.

\textbf{Figure 4.17.1.11:}

Six cylinders are unioned and are then subtracted from a rectangular solid.
A corner of the object is also cut out.

\textbf{Figure 4.17.1.12:}

A variety of Boolean operations have been applied to primitive solids to construct the mechanical assembly shown.
The Boolean operations automatically assign colors to the faces of the object(s) that they create. When both operands have the same color, that color is assigned to the resulting object and all its faces. If the two objects have a different color, the faces of the new object retain the colors they had in the original object. At the Object level the resulting object receives its color from the first operand.

The Boolean operations will occasionally be unable to complete the requested task, even though their execution appears to be done properly. When such a case occurs, the system beeps and issues a message.

The Boolean operations are unable to complete their task when they encounter a condition where a point of one object is neither on the surface of the other object, nor sufficiently far from it. That is, when a point is around the boundaries of a tolerance zone that causes it to be detected as being within the tolerance, in one instance, and outside the tolerance, in another. A minor adjustment to the position of one of the operands will usually suffice to overcome that apparent contradiction.

![Figure 4.17.1.13](image.png)

*Figure 4.17.1.13:* (a) The two objects appear to have coincident points, when viewed at a normal scale. (b) However, zooming in reveals that this is not the case.

Such a case is illustrated in Figure 4.17.1.13. At a usual scale, the two operand objects appear to be properly positioned (Figure 4.17.1.13(a)). However, when one zooms in (Figure 4.17.1.13(b)), it becomes apparent that some of the points that appear coincident, are not. Obviously, the objects were generated without the use of snaps, which typically produces fractional coordinates. While such coordinates are appropriate in many cases, they fail to produce properly touching objects when this is a desirable condition.

In general, the Boolean operators work the best when the user is decisive about the relative positions of the operands involved in the operation. If the two objects are intended to have common points or to be touching, the user should make certain that they are positioned accordingly. The use of snapping, when generating and/or positioning objects, usually guarantees appropriate results.
4.17.2 Trimming, splitting, and stitching

The Trim, Split, and Stitch operations, when applied to surfaces, produce results similar to those produced by the Boolean operations when applied to solid objects. They are based on the derivation of the line(s) of intersection of two crossing surfaces and they produce objects that consist of one or more of the pieces into which a surface is subdivided by the line of intersection. While the Boolean operations automatically select the part(s) that comprise the resulting object by applying volume containment criteria, the Trim, Split, and Stitch operations need to be told by the user which parts of the operands to return. How this is done depends on the Construct Result By option selected.

Trim/Split

The Trim/Split tool is used to execute variations of two operations, which are selected from the Trim/Split Options dialog, shown in Figure 4.17.2.1. This dialog can be invoked from the Trim/Split tool.

While it is possible to trim/split multiple objects or to apply multiple cutters, the basic trim/split operation is by its nature binary and involves two objects. Thus the initial discussion assumes two objects only. Multiple objects and multiple cutters are covered later in this section.

All operations can be executed using either the prepick or the postpick method. When using the postpick method, the Trim/Split tool is selected first (and the desired operation is selected from the Trim/Split Options dialog), then the two objects involved in the operation are picked. The order in which they are picked may be significant, depending on the particular operation that will be executed. Also, the location on the objects where the mouse is clicked when picking them is significant when Construct Result By is set to Pick Point. In this case the pick points determine the portions of the subdivided surfaces that will be returned (see below).

The Trim/Split operations, with or without stitching, can be applied to the First Object only (one way execution) or they can be applied to Both Objects (two way execution). The latter is equivalent to executing the operation twice. These options appear as columns in the Trim/Split Options dialog. Selecting an operation in one or the other column determines whether it will be executed in a one or a two way fashion.
Trimming

The Trim operation is used to cut a surface by another surface and returns one part of the subdivided surface only. This is the part where the mouse is clicked when the **By Pick Point** option is used. Which surface trims the other surface is determined by the order in which the surfaces are selected. The surface picked first is trimmed by the surface picked second. The Trim operation is illustrated in Figure 4.17.2.2.

Splitting

The Split operation is similar to the Trim, but returns all the pieces of the intersected surface, regardless of where the mouse is clicked when the surface is picked (Figure 4.17.2.3).

Trimming or splitting, stitching, and rounding

The parts that result from the intersection of two surfaces can also be joined together, after the execution of a trim or split operation. This is done when the **Stitch** option is on in the **Trim/Split Options** dialog.

When trimming and stitching are executed, the part of the first object returned is stitched with the part returned from the second object. If necessary, the directions of the part of the second object are adjusted to match the directions of the first object. The example in Figure 4.17.2.4 was executed using the **Both Objects** option.

When trimming and stitching, if the **Round** option is also on, the stitch is rounded according to the settings in the **Round** group of options. These are discussed in section 4.17.4.

When splitting and stitching are executed, the parts into which an object is subdivided are stitched back together. The result is essentially the original object with its line of intersection with the other object inserted into it. The example shown in Figure 4.17.2.5 was executed with the **Both Objects** option on.
Trimming and splitting with lines

The Trim/Split operations can also be applied by picking an object first and then a line. This can be done by selecting an option under the column labeled With Line, in the Trim/Split Options dialog. This operation uses the projection of the line on the currently active reference plane to trim the surface picked first. The line is projected in a direction perpendicular to the reference plane, which may be any of the Cartesian planes or an arbitrarily positioned plane. Examples of Split by line operations are shown in Figure 4.17.2.6.

Both Trim and Split With Line can be executed with the Stitch option on. For Trim, a surface is generated from the trim line, which closes the trimmed end of the object. For Split, the parts into which the line subdivides the object are stitched back together, resulting in an object into which the lines of intersection have been engraved. An example is shown in Figure 4.17.2.7.
Some conditions

There is one condition that needs to be satisfied before a Trim or a Split operation can be executed. The intersection line should cut the surface from one end to the other or the intersection line should be closed. In other words, the intersection line should be dividing the surface into two (or more) pieces. If the intersection line starts or ends somewhere within the boundaries of the surface, the operation will not be executed. A beep and an error message is issued instead. Whenever the Trim or the Split operations are applied to Both Objects, then the above condition should be satisfied for both objects.

When the Trim and Split operations are applied to meshed surfaces, one needs to be aware of their planarity conditions. In general, meshed surfaces contain non planar faces, but their planarity disturbances are typically small. Consequently, the trimming procedures do not check for planarity, nor do they refuse to execute the trimming when non planar surfaces are encountered (as the Booleans do). However, there is always a possibility that the mathematical calculations may produce conflicting results, in which case the program will be unable to complete the operation. The cases most likely to return incorrect results are intersection points close to boundaries of faces. When this happens you may try again after you triangulate the surfaces you wish to trim. Since triangulation consumes extra memory you may wish to triangulate only the faces that are affected by the Trim operation (rather than the whole object), whenever these faces are easy to identify.

The Trim and Split operations can also be applied to solid objects, or solid and surface objects can be mixed. This is illustrated in Figures 4.17.2.8 and 4.17.2.9.

While the Trim/Split operations can be freely applied to solid and surface objects, they cannot be applied to surface solids (double sided surfaces that contain no volume). These objects need to be transformed to one sided surface objects first (see next section).

Figure 4.17.2.8: (a) Trim/Split of both solids with and without stitching: (b) Trim and no stitch, (c) Split and no stitch, (d) Trim and stitch, (e) Split and stitch.

Figure 4.17.2.9: (a) A surface and a solid (b) are trimmed and stitched.
**Trim and stitch examples**

Possibly the most frequent use of the Trim operation with stitching is illustrated by the examples on this page. In both cases parts of an object are independently created as surface objects and are then trimmed and stitched to produce the final product.

The container in Figure 4.17.2.10 was generated as three objects that were subsequently trimmed and stitched into a single piece. The tea pot in Figure 4.17.2.11 was created as four independent mesh objects, three of which were stitched together. The fourth is the lid which remains independent.

**Figure 4.17.2.10:** (a) A container and handles were created independently and (b) stitched together.

**Figure 4.17.2.11:** A tea pot stitched from three independently created meshes.
Constructing the results

Recall that the first step of the trim/split operations (and also of the Boolean operations) is to cut one or both objects at the line of intersection with another object. After this is done, the operation needs to determine which of the two (or more) pieces of an intersected object to use for constructing the resulting object. While the Booleans, which only apply to solid objects, can determine this by themselves, the trim/split operations, which can also be applied to surface objects, need to be told by the user which parts of an intersected object should be retained. There are two methods by which this can be done, as follows:

**Construct Result By:** This group of options determines which method will be used to construct the resulting object.

**Pick Point:** When this option is on, which is the default, the pieces of the intersected object to be retained will be selected on the basis of the pick points. That is, the portion of an object where the user clicked to select it will be included in the result.

**Side Of Object:** When this option is selected, the parts of the intersected object to be retained and used in the construction of the resulting object are determined from whether they lie on one or the other side of the intersecting object. Which ones depends on the selection of the following options.

**Positive, Negative:** There are two pairs of radio buttons, one for the first and one for the second object. The ones under **First Object** determine which parts of the first object will be returned; either those that are on the **Positive** side or on the **Negative** side of the other object. Similarly for the buttons under **Second Object**.

Recall that some of the trim/split operations return parts of only the first and some return parts of both objects. Consequently, the **Second Object** buttons only apply to some of the operations. They are dimmed when they do not. The table in Figure 4.17.2.12 summarizes when they do and do not apply. Examples of the **Trim Both Objects** operation with variable combinations of **Positive** / **Negative** are shown in Figure 4.17.2.13.
<table>
<thead>
<tr>
<th>Operation</th>
<th>Stitch</th>
<th>First Object</th>
<th>Second Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim First Object</td>
<td>off</td>
<td>√</td>
<td>-</td>
</tr>
<tr>
<td>Trim First Object</td>
<td>on</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Trim Both Objects</td>
<td>on/off</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Trim With Line</td>
<td>on/off</td>
<td>√</td>
<td>-</td>
</tr>
</tbody>
</table>

*Figure 4.17.2.12:* When do the second object Side Of Object buttons apply.

*Figure 4.17.2.13:* Example of the Side Of Object options for the Trim Both Objects operation. First and Second Object set to (a) ++, (b) --, (c) +-, and (d) -. Shown in (1) wire frame and (2) hidden line.
Trimming and splitting multiple objects

The discussion to this point involved two objects only. It is also possible to cut multiple objects, or to apply multiple cutting objects, or a combination of the two. Multiple objects can be cut when Construct Result By is set to either Pick Point or to Side Of Object and the operation is not applied to Both Objects. When applied to Both Objects, multiple objects can only be cut when the Side Of Object method is used. Similarly, for all operations, multiple cutters can only be used when the Side Of Object method is used. The multiple cutters are discussed in the next section.

When applying the Trim/Split operation to multiple objects, the prepick method is always necessary. With the Pick tool, click on any number of objects to pick them. These are the objects that will be cut. Next activate the Trim/Split tool and click on the cutting object. All the preselected objects will be trimmed or split by the cutting object. Examples are shown in Figures 4.17.2.14 and 4.17.2.15.

![Figure 4.17.2.14: Trim/Split operations applied to multiple objects, which are prepicked as shown: (a) With the Pick tool click on 1, 2, 3, and 4. Then with the Trim/Split tool click on 5. (b) Trim One Object by Click Point. (c) Trim One Object by Side Of Object set to Negative. (d) Split One Object with Stitch and Side Of Object on.](image-url)
Figure 4.17.2.15: Trim With Line applied to 3 objects, which are prepicked. Stitch is on
(a) Picking the objects: With the Pick tool click on 1, 2, and 3. Then with the Trim/Split tool click on 4.
(b) The original objects in top view. (c) and (d) Trimmed by Pick Point, shown in wire frame and hidden line.
(e) and (f) Trimmed by Side Of Object set to Positive, shown in wire frame and hidden line.
Using multiple cutters

To use multiple cutters, you need to set the # Of Cutters field in the Trim/Split Options dialog, partially shown in Figure 4.17.2.16, to the desired number. Its default value is 1. Note that this field is only active and available when Side Of Object is selected. It is dimmed and inactive when Pick Point is on. This means that multiple cutters can only be used with the Side Of Object construction method.

Multiple cutters can be applied to either single objects or multiple objects. In the former case, either the prepick or postpick method can be used. In the latter case, the prepick method is required.

When cutting a single object and using the postpick method with the Trim/Split tool active, click on the object and then on as many cutting objects as indicated in the # Of Cutters field. The operation is executed as soon as the last cutter is selected. This is illustrated in Figure 4.17.2.17.

Figure 4.17.2.16: The Trim/Split Options dialog.

Figure 4.17.2.17: Trimming With Line, using three cutting lines and the postpick method: Stitch is on, Side Of Object is on, First Object is Positive, and # Of Cutters is 3. (a) The original objects and how they are picked: With the Trim/Split tool active click on 1, then 2, 3, and 4. (b) The result.

Modeling • Booleans, trim, stitch, and sections
When using the prepick method, either one or more objects to be cut can be selected. With the Pick tool, select as many objects as you wish to cut. Then activate the Trim/Split tool and click on as many cutters as indicated in the # Of Cutters field. The operation is executed as soon as the last cutter is picked. This is illustrated in Figure 4.17.2.18.

**Figure 4.17.2.18:** Trimming Both Objects, by cutting 4 objects with 3 cutters, using the prepick method: Stitch off, Side Of Object on, both First and Second Object are Positive, and # Of Cutters is 3.
(a) The original objects and how they are picked: With the Pick tool click on 1, 2, 3, and 4. Next with the Trim/Split tool active click on 5, 6, and 7.
(b) The result. (c) The result after moving objects to improve visibility.
4.17.3 Lines of intersection

While the Boolean, trim, and split operations start with the calculation of the line of intersection of two overlapping objects, there is also a tool that returns the line of intersection of two objects as its result. The objects may be solids, simple or meshed surfaces, or a combination of these, as illustrated in Figure 4.17.3.1. This tool can also be used to derive lines of intersection from the intersecting extrusions of two flat shapes. form-Z will recognize what type of objects are picked and will act accordingly. If the types are mixed, the operation will be refused. The flat shapes picked for the operation can be either open or closed and are extruded in a direction perpendicular to their plane. The line of intersection is then derived from the extruded surfaces. Examples are shown in Figure 4.17.3.2.

Figure 4.17.3.1: Lines of intersection of (a) two 2D rectangles, (b) a solid and a rectangle, (c) two surface meshes, (d), (e), and (f) two solids.

Figure 4.17.3.2: Line of Intersection from planar shapes: (a) two open shapes, (b) two closed shapes, (c) one closed and one open shape.
When thinking of the line of intersection of two surfaces, it corresponds to the volume of intersection of two solids, which is the result of the Boolean Intersection. That is, the common portion of two intersecting surfaces is their line of intersection.

The result of the Line of Intersection operation is an open or closed vector line, which is a true form-Z object. The line of intersection of solids is always a closed line (Figure 4.17.3.1(d), (e), and (f)). That of surfaces may be closed (Figure 4.17.3.1(c)) or open (Figure 4.17.3.1(a)). It may consist of one or more vector lines. When more than one, they can be returned as independent objects, or all in one object, depending on the option selected in the Status Of Objects dialog.

**Line of Intersection**

This tool is used to derive the line of intersection of two overlapping objects, when such a line exists, or of the intersecting extrusions of two flat shapes. It can be applied using either the postpick or prepick selection method. It is executed at the Object topological level.

This tool invokes the **Line of Intersection Options** dialog, shown in Figure 4.17.3.3, which contains options affecting the operation when it is applied to nurbs shapes. Note that these options are similar with options available for the Parallel tool (see section 4.5.5).

**Nurbs Control Points:** These options affect the control points of nurbs curves created from the line of intersection of one or more smooth objects.

**Relative To Curvature:** When this option is selected, the intersection curve is built such that control points are distributed relative to the curvature of the curve. The intersection curve will have more control points in areas of high curvature and less in flatter regions.

**Max Normal Deviation:** The value entered in this field specifies in degrees the density of the control point distribution along the curve. Higher values produce fewer and lower values produce more control points.

**Accurate:** When this option is on, a very accurate intersection curve is derived. This requires a significant number of control points and may change the degree of the curve if necessary.
4.17.4 Stitching

In solid modeling, it is not enough for two bounded surfaces (such as faces) to have coincident edges, if they are to be considered continuous. These edges need to be specifically linked in a manner that results in a continuous surface. Sequences of linked edges are called stitches and an operation exists for stitching coincident edges together. The operation is available both as an option, when trimming or splitting is executed, and as an independent tool.

**Stitch**

This tool is used to stitch open ends of surface objects that coincide. Note that it is necessary that edges coincide in opposite directions to be stitched. However, under proper conditions form\-Z can adjust the directions of a surface, if necessary, to stitch it with another surface (see below). Stitches can also be optionally rounded, if the proper conditions are present. This option and the parameters of rounding are selected from the Stitch Options dialog (Figure 4.17.4.1) that can be invoked directly from the tool. One of two variations can also be selected from the dialog: stitching many objects and stitching a single object.

**Stitch Multiple Objects**: When this variation is on, the Stitch tool is used to stitch two or more surfaces. When using the postpick selection method, with the Stitch tool active click on two surfaces. The operation is executed immediately. With the postpick method only two surfaces can be stitched. If you need to stitch more, you use the Pick tool to prepick any number of surfaces. Then, with the Stitch tool active you click anywhere in the project window. All surfaces are stitched, two at a time, in the order they were selected.

**Stitch Object Parts**: When this variation is selected, the stitch operation is applied to open edges within a single object. These may have resulted from a variety of other operations, such as joining, or moving open edges of objects to become coincident. This variation can also be executed with both the postpick and prepick method, and both produce the same results.

The Stitch operation is useful in many ways and facilitates transforming surfaces into solid objects. These surfaces may be generated in a variety of different ways ranging from drawing them directly to creating them through a Trim operation. In the example of Figure 4.17.4.2, the control lines in (a) were used to generate the “square” mesh surface in (b). The rotation operation was used next in Copy mode to generate another five copies of the surface and to arrange them in a manner corresponding to the sides of a cube (c). Then these surfaces were prepicked and stitched, resulting in the solid shown in (d). To confirm that this is a true solid, the object was triangulated and a 3D Section was applied to cut it into the two pieces shown in (e).
The Stitch operation is applied one segment at a time and only *perimeter* segments (single segments that do not have a reversely coincident pair) can be stitched. For a segment to be “stitchable” its points should be on a perimeter point or segment of the other surface. When on a segment, the system inserts a point on that segment of the other surface. For two segments to be stitched, they should be in reverse directions. When the directions of the segments do not match, the directions of the second surface are adjusted.

When stitching more than two surfaces, they can be prepicked in any order and stitched properly, provided all their directions are consistent. However, if the directions of some also need to be adjusted, they need to be prepicked in the order in which they touch or otherwise *form-Z* may be unable to adjust the directions and stitch them properly. This is illustrated in Figure 4.17.4.3. The third shape from the left has an opposite direction. When the shapes are prepicked in order, left to right, the program is able to adjust the direction and stitch them all properly. When they are picked out of order, as in (b), stitching is incomplete.

**Figure 4.17.4.2:** (a) The control lines and the c-mesh surface they generate.
(b) Five copies are made and rotated to form a cuboid, shown here exploded.
(c) The pieces are stitched together, producing a true solid.
(d) The solid is triangulated and a 3D section is cut through it.

**Figure 4.17.4.3:** Stitching four rectangles, one of which has opposite direction.
(a) Prepicking them in the order they touch adjusts direction and stitches them all correctly.
(b) Prepicking them in a non-continuous manner skips one stitch due to that it can not adjust direction.
Rounding stitches

Both the **Trim/Split Options** and the **Stitch Options** dialogs contain options which allow you to also round a stitch after two surfaces have been stitched. The **Round** section of their dialogs includes two parts: one for **Facetted** and one for **Smooth**. In the **Trim/Split Options** dialog, the rounding options are only available when **Stitch** is on. They are dimmed and inactive otherwise.

**Facetted Options**: This group of options applies to stitches of facetted objects.

**Distance**: The value entered in this field determines the distance from the stitch line at which the end control lines of the rounding mesh will be placed.

**# Of Rounding Points**: This number, which can only be an even number, determines the rounding resolution, which is how many mesh tiles will be created between the two ends of the rounding mesh.

Examples of both plain and rounded stitches are shown in Figures 4.17.4.4, 4.17.4.5, and 4.17.4.6.

---

![Figure 4.17.4.4:](image)

*(a) Trim First Object, Stitch, and Round (b) off and (c) on.*
Figure 4.17.4.5: (a) Trim Both Objects, Stitch on, and Round (b) off and (c) on.

Figure 4.17.4.6: (a) Trim With Line, Stitch on, and Round (b) off and (c) on.
**Smooth Options:** This group of options applies when the stitch of smooth objects is rounded.

**Edge Type:** One of three options can be selected from this pop up menu: Circular, Elliptical, or Bevel. Each of these items has a different set of options which are displayed in the dialog when the respective item is selected. These options are discussed in more detail in section 4.12.1 and are shown in Figure 4.12.1.1. Examples of smooth rounding options are shown in Figures 4.17.4.7.

*Figure 4.17.4.7:* Examples of smooth rounding options for the Trim/Split tool.
4.17.5 Sections of 3D objects

Both 2D and 3D sections can be derived from 3D solid objects. The 2D sections are surface objects. The 3D section operation effectively splits a 3D object into two or more pieces, each of which is a well formed 3D solid.

Section

This tool is used to execute 2D and 3D sections. Each can be derived using either a cutting plane or a cutting line.

The type of section to be derived and the type of cutting entity are selected from the Section Options dialog, shown in Figure 4.17.5.1, which is invoked directly from the Section tool. The only other options the 2D and 3D sections require are those selected from the Status Of Objects dialog that can be invoked from the Section Options dialog.

2D section

When 2D is selected in the Section Options dialog, the Section tool is used to generate a 2D section from one or more 3D solids, using either the postpick or prepick selection method. With the postpick method, the Section tool is used to first click on a solid object and then on a cutting plane or a cutting line. Which one is used depends on the Cut Object With option selected in the Section Options dialog. With the prepick method, any number of solid objects can be prepicked with the Pick tool and then, with the Section tool active, you click on a cutting plane or line. All the preselected solids are sectioned with the cutting entity.

A cutting plane is a plane defined by a 2D shape, or a face or outline of a 3D object. Any one of these entities can be selected when the Plane option is on. The 2D section is generated at the intersection of the solid with the plane, provided such an intersection exists. Note that the cutting plane extends to infinity and is not limited by the borders of the 2D shape, which is used to define it. Sections are commonly derived at positions parallel to the Cartesian planes (Figure 4.17.5.2). However, any arbitrarily positioned plane may be used to derive a section.
A cutting line is any single segment line or the segment of a 3D object, from which a plane perpendicular to the active reference plane is derived. This plane is then used to intersect a solid, as with the cutting plane. A cutting line can be selected when the **Line** option is on in the **Section Options** dialog. Examples are shown in Figure 4.17.5.3.

Note that, when a vector line that consists of more than one segment is selected as a cutting line (and is selected at the Object topological level), its first segment is used only and always. This is illustrated in Figure 4.17.5.4(a) and (b). In (a), even though the vector line is selected by clicking on its second segment, its first segment is used for the cutting plane. In (b), the direction of the line has been reversed and the new first segment is used for the cutting plane.

If another segment of the vector line needs to be used for the section, then it should be selected at the Segment topological level. Note that this is only possible when the prepick method is used. That is, after selecting one or more solids with the Pick tool at the Object level, the Section tool is activated, the topological level is set to Segment, and you click on the segment that needs to be used to define the cutting plane. This is illustrated in Figure 4.17.5.4(c).

Sections are typically used for construction drawings, in architectural design or other fields. Examples of horizontal and vertical sections of a building are shown in Figure 4.17.5.5. 2D sections generated in modeling can easily be transferred to drafting for further detailing and annotation.
Figure 4.17.5.5: Deriving sections of a building model. (a) The building in hidden line. (b) Positioning the planes of the sections. (c) The position of the vertical sections, in XY projection. (d) The position of the horizontal section, in ZX projection. (e) Horizontal and (f) vertical sections.
3D section

The 3D section operation is executed exactly as the 2D section operation is. A 3D section of one or more 3D solids may be derived by designating the position of a cutting plane or a cutting line, after the solids have been picked. The result is one or more solid objects. Whether a Plane or a Line is used is determined by the selection in the Section Options dialog, as for the 2D section.

Any plane parallel to one of the Cartesian planes or any arbitrarily positioned plane may be used to derive a section. The position of the cutting plane is determined by selecting a 2D shape, a face, or an outline of a solid object. The proper topological level should be used to tell the system what entity will be used for the definition of the plane. Selection of an entity at a level other than object can only be done using the prepick selection method. When a 3D solid object is selected as a cutting shape (with topological level at Object), the system will use its first face to determine the cutting plane. Examples of 3D sections with Plane are shown in Figure 4.17.5.6.

As for 2D sections, any line can be used for generating a 3D section with Line. Examples are shown in Figure 4.17.5.7. Normally a single segment line is used, but a vector line, or a segment of a 3D object can also be used. The 3D section is always cut with a single flat plane, even when a vector line is used. If a section in the shape of a vector line is desired, then a trim with line and stitch operation should be used.

Figure 4.17.5.6: 3D sections of solid with Plane:
(a) The model and three cutting shapes.
(b) Model split by planes 1 and 2, and moved.
(c) Model split by cutting plane 3 and rotated.

Figure 4.17.5.7: Sections of solids with Line.
(a) Three objects and a line with XY plane active.
(b) 3D sections, moved, and shown in hidden line.
4.17.6 Contours

The contours are sets of 2D sections which can be derived from one or more objects. They require only the selection of the object(s) from which the set of sections will be derived. They do not require you to select a cutting shape, but they are derived in orientations parallel to the active reference plane. For solid objects, the contour lines are closed surface objects. For one sided surface objects, the contour lines are open lines (open surface objects). The distances between the contours and other parametric options are selected from the Contours Options dialog, and are discussed below.

Contours

The Contours Options dialog (Figure 4.17.6.1) invoked from this tool contains the options that determine the positions and density of the contour lines to be derived from an object.

With the postpick method, activate the desired reference plane, select the Contours tool, and click on the object whose contours you wish to derive. With the prepick method, use the Pick tool to preselect any number of objects, select the Contours tool, and click anywhere in the graphics window.

![Contours Options dialog](image)

Figure 4.17.6.1: The Contours Options dialog.

Use Contour Increment: When this option is selected, the contour intervals (distance between consecutive contours) will be determined by the value entered in the Increment numeric field. Three sub options determine where the contours will begin.

Start At Increment Multiple: When this option is selected, the first contour will be at the lowest position that intersects the object and is a multiple of the contour increment. The last contour will be at the highest position that intersects the object and is a multiple of the contour increment.

Start At Min Plus: When this option is selected, the first contour will be at the lowest point of the object plus an increment given by the value entered in the field next to it. The default for this value is 0 (zero).

Start At: With this option selected, the first contour will be placed at the position indicated by the value in the field next to it. If this position is beyond the ends of the object, then the contours will start at the first increment that intersects the object, if such position can be found.
Use # Of Increments: When this option is selected, the number entered in the # Of Incre-ments field will determine how many contour lines will be generated. The contour interval will be determined by dividing the distance of the start and end positions of the contour lines by the number of increments. The start and end are determined by the following two parameters:

Start At Min Plus: The value entered in this field is added to the lowest point of the object to determine the position of the first contour. Default is 0.

End At Max Minus: This value is subtracted from the highest point of the object to determine the position of the last contour. Default is 0.

Note that the terms lowest and highest as used above are relative to the position of the reference plane, which determines the orientation of the contour lines. For example if the YZ reference plane is active, then lowest means the position with the lowest X value, and highest means the position with the highest X value.

One Object Per Contour: This option applies only when a section consists of more than one outline. When deselected (default), all of the section’s outlines will be returned as a single object. When selected, one object will be produced for each outline.

This option also relates to the Status Of New Objects options set in the Status Of Objects tab. If New Object Status: is set to Single Object, then the complete set of contours derived from one or more objects will be returned as a single object, regardless of whether One Object Per Contour is off or on.

Zero Heights (On Reference Plane): Normally, all contour lines will preserve the positions where the respective sections were derived. If this option is selected, then all contours will be assigned a 0 elevation, where the meaning of “elevation” is relative to the position of the reference plane. Selection of this option places all the contour lines on the reference plane. This is illustrated in Figure 4.17.6.2.

Contours are typically used to represent landforms, however, the contours of any object can be derived and can also be used in a variety of other ways, such as to represent abstract sculptures. Examples are shown in Figures 4.17.6.3 through 4.17.6.5.
Figure 4.17.6.2: Contours at 0 elevation derived from a pyramid: (a) XY, (b) YZ, and (c) ZX orientation.

Figure 4.17.6.3: Contours parallel to an arbitrary plane are derived from a building model.

Figure 4.17.6.4: (a) Given a set of contours and a site, (b) a terrain model is derived, from which (c) horizontal contours at 1' increment and 0 elevations and (d) vertical contours at 4' increments are derived.

Figure 4.17.6.5: (a) From the model of a glass (b) horizontal (XY) and (c) vertical (YZ) contours are derived.
4.17.7 Generating cages

When modeling large scenes, especially when these scenes are manipulated for animation, it is useful to temporarily use low resolution objects in the place of the real object, which may be high resolution. While there are many methods in form-Z that can be employed to produce a low resolution object, which is volumetrically equivalent to the real object, the following tool is a direct method for producing such a result.

A **cage** is a low resolution object that is derived from another object. It is derived by first projecting the perimeter of an object in two coordinate planes (which are always perpendicular to each other), extruding them at low and high heights such that they cover the other projection (from min to max) and finally deriving the Boolean Intersection of the two extrusions. Because of the latter operation the Cage tool may be thought of as a special Intersection, and its default place on the tool palette is together with the Booleans.

![Cage](image)

This tool can be used to generate low resolution objects from other objects by deriving the intersection of two objects extruded from two projections of their perimeter. The projection can be to any combination of two Cartesian coordinate planes (world space) or relative to the reference planes. These options are selected from the **Cage Options** dialog (Figure 4.17.7.1), which is invoked directly from the tool.

![Cage Options Dialog](image)

*Figure 4.17.7.1: The Cage Options dialog.*
**Extrusion Directions:** The three radio buttons in this group of options determine the two directions in which the extrusions will be derived. All three possible combinations are available. When, for example, the default $Y, Z$ is selected, a projection is derived on the XZ plane (front view) and extruded in the Y direction and another projection is derived on the XY plane (top view) and extruded in the Z direction. Two typical cages derived in $Y, Z$ and $X, Y$ are shown in Figure 4.17.7.2.

**Relative To Reference Plane:** When this option is on the directions in which the object’s perimeter is projected and extruded are relative to the reference plane, as opposed to relative to the world Cartesian planes when this option is off, which is the default. This option is illustrated in Figure 4.17.7.3.

**Filter Segments Shorter Than $n$:** This option offers control of the resolution of the cage. When on, the projected two shapes are filtered according to the value entered in its numeric field. That is, segments longer than $n$ are deleted. The extrusions are then derived from the filtered projections. An example is shown in Figure 4.17.7.4.

---

**Figure 4.17.7.2:**
(1) A high resolution object and cages derived in 
(2) $Y, Z$ and (3) $X, Y$.

**Figure 4.17.7.3:**
Cages of an ellipsoid in $X, Y$ and relative to 
(1) the XY Cartesian plane and 
(2) an arbitrary reference plane.

**Figure 4.17.7.4:**
Cages at different resolutions. 
(a) Object is about 50’ long. 
Cage with (b) Filter option off, 
(c) Filter on and set to 4’, and 
(d) Filter on and set to 8’.
Note that cages consist of a single volume only. Consequently, if the object from which the cage is derived contains a hole, it will not be included in the cage. This is illustrated in Figure 4.17.7.5 and 4.17.7.6. It is also possible that an object with cavities may produce more than one outline when projected in some direction. When this condition occurs the outline with the largest area will be used, which may result in an inaccurate volumetric equivalency of the original object. When this happens, or whenever cages of very complex objects are needed, it is advised that the user manually outlines the proper projections, extrudes them and intersects them to achieve the desired result.

*Figure 4.17.7.5:* A cage derived from a torus does not include the hole.

While most frequently cages are generated from solid objects, they can also be generated from surface objects (closed shapes and meshed surfaces, but not from open lines), provided they project to shapes that contain some area rather than being single lines. An example is shown in Figure 4.17.7.7.

The restriction that objects should project to more than a line to be able to derive a cage from them is illustrated in Figure 4.17.7.8. The meshed surface in (a) is flat and parallel the XY plane. Thus its YZ and ZX projections are single lines and objects extruded from them contain no volume and no Boolean Intersection can be derived from them. After the meshed surface has been rotated twice, as in (b), a cage can be derived.

*Figure 4.17.7.6:* Cages don’t include the holes of the objects they are derived from and may have significantly larger volumes.

*Figure 4.17.7.7:* (a) A meshed surface and cages derived in (b) X, Y, (c) Y, Z, and (d) Z, X.

*Figure 4.17.7.8:* (a) Meshed rectangle is parallel to XY and no cage can be derived from it. (b) After it is rotated, (c) a cage can be derived.
4.18 Joining, separating, grouping, and ungrouping

The following tools are discussed in this section:

- Join
- Separate
- Group
- Ungroup
- Unclone

The first tool above makes a single object out of two or more objects, volumes, or simply pieces. The second tool does the opposite and can also be used to separate parts of an object.

A solid volume is a continuous and closed collection of faces. In form•Z the volume is an implicit topological level. Most frequently a solid object, i.e. a cube, consists of a single volume. All the object generation tools of the program produce one-volume objects. However, other operations, most notably the Boolean Difference, may split an object into more than one volume. Recall that, in such cases, the option is available to instruct the program to return these volumes as a single object or to make one object for each volume (which is the default). There are also the functions that can be performed by the first two tools above.

The third tool above places objects into groups. Grouping is a linkage that can be superimposed to objects to organize them in plain or hierarchical structures. While grouping may be thought of as a linkage weaker than that resulting from the Join and Union operations, in actuality it is a different linkage. It is such that objects preserve their individuality while they are also linked. Grouping relates to layers, which represent yet another method of grouping and, in form•Z, the two coexist and can overlap. The fourth tool above does the reverse of grouping. It can be used to remove an object from a group or to break a group completely.

The last group is used to remove an object from a clone family or to completely dismantle a clone family. Cloning is a special type of grouping that is discussed in section 4.23.5.
4.18.1 Joining volumes

The Join tool can be used to make a single object by combining the volumes of any number of objects selected. Generally, this operator is quite permissive, and does not check whether or not the volumes of the joined objects intersect. Normally, they should not overlap. If they do and an attempt is made to use the respective object for a Boolean operation, the program will be unable to complete it. The program does not check whether the objects being joined are all solids or all surface objects, and allows you to mix the two. While such mixed objects may sometimes produce interesting renderings, in general they are not acceptable by the modeling operations, and we recommend that you use the Join tool only to join uniform types of non intersecting objects.

The Join operation can be executed using the postpick or the prepick method. However, the prepick method will be most frequently used, because more than two objects will be joined in most cases, and the postpick method will allow you to select only two objects.

The differences and similarities between the Join operation, and the Boolean Union and the grouping operations should be noted. In a way, the Join operation is weaker than the Union and stronger than the Group operation. The Union operation actually calculates the intersections of objects and resolves them, possibly returning a number of volumes structured as a single object. The latter is the only similarity with the Join Volumes operation. Only when non overlapping objects are selected for the Union is the result identical to that of the Join Volumes. The Group operation, on the other hand, does not affect the structure of the objects, which remain in their original form. It simply superimposes additional links that designate objects as belonging to the same group.
4.18.2 Separating volumes and parts of objects

Separate

This tool can be used to execute four types of separations: it separates the volumes of objects, separates along a stitch line, along a sequence of segments, or along the boundaries of faces. The type of separation to be executed is selected from the Separate Options dialog (Figure 4.18.2.1), invoked directly from the Separate tool.

Separating volumes

When an object consists of more than one volume, this operation can be used to create an independent object for each volume. In essence, the result of this operation is the reverse of that produced by the Join tool. This operation can be executed using either the prepick or the postpick method.

While volume is a term initially meaningful only for solids, it has been extended to include any continuity of faces, not necessarily closed. Continuity is established by any pair of reversely coincident segments that allows two faces to be identified as neighbors. Since joining non solid volumes has been permitted, the separation of such volumes is also possible. Separate works at two different modes and automatically switches to one or the other. If the volumes it is given are recognized to be true solids, it applies a complete hierarchical analysis of the volumes and structures them properly before returning the resulting objects.

Proper and hierarchically structured solids means that “void” volumes that may be contained within other volumes are returned as parts of the same object and are actually spaces contained within a solid. A typical example is the 3D Enclosure that is created with closed top and bottom. If the program detects entities that do not qualify as solids, it simply makes one object for each continuous group of faces. For example, both 2D and 3D holes will be returned as independent objects. These distinctions are illustrated in Figure 4.18.2.2.
Separating along a stitch

The definition of a stitch is as for the stitch rounding and the 2D Surface Object From Stitch operations. The Separate Along Stitch operation is executed by clicking on a single segment with the Separate tool active. The system automatically detects a closed sequence of segments that constitute a stitch (if one exists), breaks (splits) the object along the stitch line, and creates two objects if the stitch line breaks the object into two volumes.

This operation can also be executed using the prepick method. Preselect any number of segments with the Pick tool, then activate the Separate tool and click anywhere in the graphics window. The object(s) to which the stitch lines belong are broken into two or more pieces, and a separate object is constructed from each piece. Examples of the Separate Along Stitch operation are shown in Figure 4.18.2.3.

Separating along selected segments

This operation breaks an object along a preselected sequence of segments that is either closed, or starts and ends at the boundary of a surface object. The selected sequence of segments must split the object or an error message will be posted.

While this operation can be executed using the postpick method, it can postselect only a single segment, and will work only when such a single segment actually splits the object into two pieces. Consequently, the prepick method is generally more appropriate for this operation.

With the prepick method, preselect any number of segments using the Pick tool, then activate the Separate tool and click anywhere in the graphics window. The system will use the preselected segments to construct all the possible closed sequences, or open sequences that start and end at the boundaries of surface objects. It will then split the object along these lines into two or more pieces, and construct an object from each piece. Examples are shown in Figure 4.18.2.4.
Separating at boundary of selected faces

This operation breaks an object along the boundary of selected groups of possibly adjacent faces. Both the postpick and the prepick methods can be used to execute this operation. However, the postpick method allows the selection of only a single face, and therefore can separate an object only along the boundary of this single face. Consequently, the prepick method is generally more appropriate.

When using the prepick method, any number of faces are preselected with the Pick tool. You then activate the Separate tool, and click anywhere in the graphics window. The system will detect all the groups of adjacent faces, and break the object along their boundaries. The operation may result in more than two pieces, each of which becomes an independent object. Examples of this operation are shown in Figure 4.18.2.5.
4.18.3 Grouping and ungrouping

Both surface and solid objects can be placed into groups. Grouping allows objects to be “linked” in a hierarchical structure. This provides the ability to manipulate the grouped objects simultaneously through operators such as the geometric transformations.

When an object is generated, the system automatically creates and places it in its own “private” group, to be referred to as the individual group. Individual groups are set at the highest level of grouping and represent the least complex state of grouping. Each individual group contains a single object. Any number of groups, including individual groups (objects), can be linked to form a new group. The new group is always created at the highest level, and reduces the grouping level of the objects involved by one level. Picking with the topological level set to Group, picks groups at the highest level only. Groups at lower grouping levels are of significance to the Ungroup operator. The grouping structure is of a hierarchical nature. Therefore, there is no limitation to the number of grouping levels that can be applied to the objects and groups of a project.

The grouping operators may be applied using either the prepick or the postpick method. When the prepick method is used, any number of operands are first picked at the Object or Group topological level. Then the Group or Ungroup tool is selected, followed by a click of the mouse anywhere in the project window. This initiates the execution of the grouping/ungrouping operation. With the postpick method, with the Group tool active you click on two objects or groups. Or with the Ungroup tool active you click on a single group. Note that postpicking always picks at the Group level.

Because with the postpick method the Group tool can be applied to two entities at a time only, and grouping typically needs to be applied to more than two entities, the postpick method is often restrictive. In these cases the prepick method is more appropriate.

Group

This tool creates a new group and all the picked groups (or objects) become its members. The new group is placed at the highest level of the grouping hierarchy.

Ungroup

This tool can be used to execute three distinct ungrouping operations, which are selected from the Ungroup Operations dialog shown in Figure 4.18.3.1.
**Dismantle One Level:** Recall that grouping may occur at any number of levels. When this operation is selected and you use the Ungroup tool to click on a group, the highest level of that group is broken. When using the prepick method, entities should be picked at the Group topological level. Postpicking ignores the current topological level setting and always picks at the Group level. This is the default setting in the **Ungroup Operations** dialog.

**Dismantle All Levels:** When this operation is selected and you use the Ungroup tool to click on a group, all grouping levels of that group are broken and all the objects become independent.

**Remove Object From Group:** When this operation is selected and you use the Unrgoup tool to click on an object, its grouping link is broken and the object is extracted from the group. All the other grouping relationships remain intact.

Due to their nature, the effects of the grouping operators are not readily visible on the screen. The only visible effect is that the operands that were prepicked and highlighted, are unpicked and unhighlighted, as soon as the grouping operation is completed. This is an exception to the general rule that prepicked entities remain picked after the completion of an operation. The current grouping status of objects can be verified by picking and highlighting groups. It can also be verified in the Objects palette, where the names of objects and the names of groups are displayed together with their grouping structures. Whenever an object is placed within a group, its individuality remains intact. Grouped objects, and their parts, can still be accessed individually by setting the topological level to levels other than the Group level.

Figure 4.18.3.2 shows a sample grouping structure and how it is affected by the grouping/ungrouping operators. Objects are labeled with arabic and groups with roman numerals. Groups are shown as nodes, which are graphically represented as dots (small circles). Three levels of nodes are used: There is a single node at the **project level**, called the root, shown as a large black dot (●). The groups at the highest level, right after the root, are shown as smaller black dots (●). The groups at all other levels are shown as white dots (○). The significance of these distinctions is that only the groups marked with the small black dots can be picked as groups. To be able to pick the lower level groups, they first need to be raised to the highest grouping level by applying the appropriate operation.

The example in Figure 4.18.3.2 is structured in ten steps, labeled (a) through (j). Each step illustrates the result produced by a single grouping/ungrouping operation, both graphically and by showing the content of the Objects palette.
(a) Objects 1 through 7 are generated sequentially, which places them in individual groups at the highest grouping level.

(b) Objects 6 and 7 are grouped: the Group tool is selected, the topological level is set to Group, and the mouse is clicked on objects 6 and 7. This pushes them one level down in the grouping structure and the new group, labeled i, is placed at the highest grouping level, together with objects 1 through 5.

(c) Objects 1, 2, and 3 are grouped into a new group labeled ii: they are prepicked, using the Pick tool, with the topological level set to Group, the Group tool is selected, and the mouse is clicked anywhere in the graphics window. The top level of the grouping structure now consists of two groups and two individual objects.

(d) Object 5 and group i are grouped: the Group tool is selected, the topological level is set to Group, and the mouse is clicked on object 5 and on one of the objects of group i. The new group is labeled iii. The grouping structure now consists of three levels.

(e) Groups ii and iii are grouped, creating a new group labeled iv: the Group tool is selected, the topological level is set to Group, and groups ii and iii are picked. This adds one more level to the grouping structure, which now has four levels.

(f) New objects 8 and 9 are created, which places them in individual groups, at the highest level of grouping.

(g) Group iv is ungrouped: while Dismantle One Level is on in the Ungroup Operations dialog, the Ungroup tool is selected, the topological level is set to Group and the mouse is clicked on any of the objects contained in group iv (object 1, 2, 3, 5, 6, or 7). This eliminates group iv and moves groups ii and iii up one level in the grouping structure. The grouping structure now has one level less.

(h) Object 2 is taken out of group ii: while Remove Object From Group is on in the Ungroup Operations dialog, the Ungroup tool is selected, the topological level is set to Object, and the mouse is clicked on object 2. This takes object 2 out of its group and places it in an individual group at the highest level of the grouping structure.

(i) Group iii is dismantled: the Ungroup tool is selected, Dismantle All Levels is selected from the Ungroup Operations dialog, the topological level is set to Group, and the mouse is clicked on one of the objects in group iii (5, 6, or 7).

(j) Group ii is dismantled, the same way group iii was dismantled in step (i). This results in a set of nine objects that are completely free from any grouping links. They are all placed in individual groups at the highest (and single) level of the grouping structure.
Figure 4.18.3.2: Hierarchical structure of the grouping of objects and changes in the structure created by the grouping tools.
4.18.4 Unclone

The Unclone tool offers the ability to cancel cloning, which is a kind of grouping and is discussed in section 4.23.5.

Unclone

This tool can be executed in either postpick or prepick mode. In postpick mode, with the Unclone tool active, you click on a member of a clone family. In prepick mode, you use the Pick tool to preselect a number of clones, from the same or different families, and then, with the Unclone tool active, you click anywhere in the project window. In both modes, either the picked object or objects are removed from the clone family or the complete clone family is dismantled.

The effect of the unclone operation is determined by the setting selected in the Unclone Options dialog, shown in Figure 4.18.4.1, which is invoked directly from the tool.

Individually: When this option is on, only the picked clone is affected by the operation. It is removed from the clone family and it becomes an independent object.

Entire Family: When this option is on, the complete family of the picked clone is dismantled and all its members become independent objects.

Figure 4.18.4.1:
The Unclone Options dialog.
4.19 3D text

The following three tools that relate to placing and manipulating text are discussed in this section:

- Text Place
- Text Edit
- Text Search and Replace

Text can be created using the first tool above and can be edited using the second tool. The extent of available editing depends on the type of text that is created.

When text is generated, the type of text to be created is selected from the **Text Place Options** dialog. To create and place new text, with the Text Place tool active, use the mouse to enter one or two points, or to select one or two control lines. As soon as this is done, the **3D Text Edit** dialog is invoked, in which you type the text you wish to generate and select the text parameters you desire. When you close the dialog (click on **OK**), the text is generated.

**form•Z** supports both True Type and Post Script fonts, which are all selected from the same pop up menu. The font type currently selected is listed next to the menu. **form•Z** also supports Open Type fonts, which are displayed as True Type fonts.

Letters and other characters are stored by assigning (encoding) a number for each character. Traditionally each language or language variant has its own encoding system on each platform (Macintosh and Windows). Even for a single language like English, no single encoding is adequate for all the letters, punctuation, and technical symbols in common use. Unicode solves this problem by providing a unique number for every character, for all platforms and languages. Unicode fonts and characters are now supported in **form•Z**.

Internally **form•Z** stores all text with the Unicode encoding. This improves the transfer of text between platforms and languages. It also improves the ability to change fonts, use graphical characters, use right to left reading languages and use characters of different languages without switching fonts. There are two ways to create unicode characters. The keyboard is used to enter the traditional common characters for the current language. Both OS X and Windows provide an additional external method for entering characters in addition to the keyboard. On OS X, the Character palette provides a listing of the characters which can be entered in the Text Editor dialog by clicking on the character in the palette. The Character palette is enabled in the Input Menu section of the International Preferences. Once enabled, the palette can be invoked by selecting it from the Input menu at the upper right of the menu bar. See the OS X documentation for more details on the Character palette. On Windows, the Character Map accessory provides a copy and paste mechanism to get the characters into the Text Editor. The Character Map accessory is accessed by selecting it from the System Tools sub-menu of the Accessories menu from the Start menu. See the Windows documentation for more details on the Character Map.
Note that not all fonts support all of the unicode characters. The font must create a glyph (graphical representation of a character) for each supported unicode character. If the glyph for a character is not present in the selected font, then a default rectangle is shown in place of the character.

The majority of languages have text that is typed left-to-right. However, there are also languages, such as Arabic and Hebrew, whose text is written right-to-left. These two types of text behave differently when they are entered. Left-to-right text moves the cursor to the right as characters are typed and are appended to the right end of the character string. In contrast, right-to-left pushes the character string to the right as new characters are typed, but the new characters are placed at the left end of the string.

Note that, to generate right-to-left scripts, you need to be running form•Z under an operating system that supports such writing. Also note that right-to-left writing systems are in essence bi-directional, because they typically mix numbers, which are always written left-to-right, and foreign words. Consequently, in form•Z, right-to-left writing has been implemented as bi-directional.

After text has been generated, there is no visible difference between left-to-right and right-to-left strings, as shown in Figure 4.19.0.1. Only those that can read the respective language can recognize the direction in which the text should be read.

You can create plain text or objects that have the shape of text. These can be plain or controlled objects. Controlled text objects can be edited and changed. This is done using the Text Edit tool. With the types of text that use control lines, you can edit and change the shape of the control lines using the Edit Controls tool. Examples of 3D text are shown in Figure 4.19.0.2.
**Text as Object**

Text as Object can be generated in a standing position and can be a surface or a **solid** object.

Note that this text is center justified.

**Plain Text** can be generated on the active reference plane.
Note that this text is justified and how texts and slides can be mixed in the same block of text.

**Text Between Points**

**Text on curved line.** Long close

**Text between parallel lines.**

**Figure 4.19.0.2:** Examples of 3D text available in the modeling environment.
4.19.1 Placing text

The desired type of 3D text is selected from the Text Place Options dialog (Figure 4.19.1.1), which is invoked directly from the Place Text tool.

There are three types of text, each of which has additional variations and options. Two of these types are objects, which can be facetted or smooth.

Plain Text: This is text which is not an object. Such text is redrawn on the screen every time the modeling image is regenerated. Plain text can be moved, rotated, and scaled using the geometric transformation tools.

Plain text is placed on the active reference plane, which can be one of the orthogonal planes or any arbitrarily positioned plane. The plain text may lie flat on the reference plane or may be perpendicular to it (standing up). These as well as some additional options are selected from the 3D Text Editor dialog which is invoked as soon as the 3D text placement points (mouse clicks) are completed. Two methods are available for placing plain text:

At Point: When this placement type is selected, after the selection of the Place Text tool, the mouse is clicked at some position on the reference plane. The point where the mouse is clicked becomes the basis for the text placement. The exact placement of the text depends on the Justification and other parameters selected from the 3D Text Editor dialog.

Between Points: This type places text between two points and requires two mouse clicks. After selection of the Text Place tool, the mouse is clicked on the reference plane, which causes a line to be rubber banded until the mouse is clicked again. The line determines the direction of the text while its end points determine the justification points of the text. The text may be left, center, right, or force justified between the two end points. The desired options are selected from the 3D Text Editor dialog.

Model Type: These options allow you to generate either Facetted or Smooth types of text, when generating text objects. They have no effect on plain text.

Facetted: When this option is on, polygonal text objects are generated.

Smooth: This type of text generates smooth objects which behave as the other smooth objects in form•Z. They are resolution independent and they are editable as smooth objects.
**Text As Object**: This is text which is generated in the same two ways plain text is. However, internally it is represented as an object which, once it has been generated, can be further manipulated as any object. A single object can be created for a string of text, or each individual letter may be an independent object. This and other options are selected from the 3D Text Editor dialog when the text is first created or edited. **Text As Object** is placed as the Plain Text, but some additional options, selected from the 3D Text Editor dialog, are available to it.

**Text As Object Along Path**: This is text which is placed along splines that function as control lines. These lines are first drawn and are selected at the time the Text Place tool is executed. A single or a pair of lines may be used for placing this type of text. **Text As Object Along Path** may be executed using either the prepick or the postpick method. With the postpick method, the Text Place tool is selected first, then one or two control lines are selected. With the prepick method, one or two control lines are selected using the Pick tool, the Text Place tool is selected and the mouse is clicked anywhere in the graphics window. There are three variations of **Text As Object Along Path**. The first two require a single control line. The third requires two lines.

**On Line**: When this type of text is selected, the text is placed along a single spline. The text may be left, center, right, or force justified from one end of the line to the other. How the justification is applied depends on the selection of one of three available options.

**Preserve Height & Width**: When this option is selected, the dimensions of each text character are preserved and retain the dimensions entered in the 3D Text Editor dialog. With this option the Justification options found on the top of the 3D Text Editor dialog are also available. Selecting Left, Center, or Right will justify the text accordingly, using only part of the control line, provided the length of the text string is less than the length of the line. If it is too long to fit on the line, an error message will be issued. When Fit is selected, the control line is subdivided proportionally to the characters in the given text string and then each character is centered in the respective section of the line (Figure 4.19.1.2(a)). This option essentially “fills” the control line by enlarging or decreasing the spaces between the letters. Note that, if the selected line is shorter than the total length of the string, then the individual characters will overlap.

**Scale Width Only**: When this option is selected, the width of the characters and the spaces between them are scaled proportionally to fill the length of the control line. The heights of the characters retain their original sizes (Figure 4.19.1.2(b)).

![Figure 4.19.1.2: Text On Line using the: (a) Preserve Height & Width with Fit Justification, (b) Scale Width Only, and (c) Scale Height & Width options.](image-url)
**Scale Height & Width:** This option applies the same scaling factor used for the widths of the characters to their heights. Thus the resulting individual characters retain a width/height ratio analogous to their original sizes (Figure 4.19.1.2(c)). However, they are typically larger or smaller, unless the total width of the complete string (including spaces) is equal to the length of the control line.

**Between Parallel Lines:** This type of text requires the selection of a single control line, however, the text is generated between two lines. The second line is derived from the first and is parallel to it at a distance equal to the height of the text. To generate and place a given string of text, each line is divided proportionally to the widths of the characters in the text. Pairs of opposite sections of the subdivided control lines (one from each of the lines) form a bounding box, within which the corresponding character is generated. This type of text is illustrated in Figure 4.19.1.3.

![Figure 4.19.1.3: 3D text Between Parallel Lines.](image)

**Between Lines:** This type of text requires the selection of two control lines. The order in which the control lines are selected is significant and determines the direction of the text. The base of the characters is placed on the first line selected. The text string is generated between the two control lines, as for the **Between Parallel Lines** text. This type of text is illustrated in Figure 4.19.1.4.

![Figure 4.19.1.4: 3D text Between Lines.](image)

The control lines selected for all the **Text As Object Along Path** types are not required to be planar. While planar control lines will frequently give cleaner results, imaginative shapes of text can also be created using non planar lines. The control lines may consist of one or more segments, they may be open or closed, and they may be used as they are, or they may be curved by selecting an option available in the **Text Place Options** dialog.
The direction in which the text control lines are drawn is significant, since it determines the direction of the text. For example, drawing a line right to left on the XY plane will produce reversed text (see Figure 4.19.1.5(a)). For the **Between Lines** option, both control lines should be drawn in the same direction, or the text string will be twisted (Figure 4.19.1.5(b)).

**Figure 4.19.1.5:** (a) Text is reversed when the control line is drawn right to left. (b) Text is twisted when control lines drawn in opposite directions are used for the **Between Lines** type of text.

**Adjusted To Line/Curve:** When this option is not selected, each character of the text retains its overall rectangular shape, as illustrated in Figure 4.19.1.6(a). That is, while the corner points of the bounding boxes of the characters lie on the control lines, the bottom and top edges remain straight lines and are typically positioned away from the control lines, if the corresponding portions of the control lines are curved. Selection of this option forces the overall shape of a character to be completely adjusted to the shape of the control line. With faceted text, this is achieved by inserting new points on the outline shape of the character, which correspond to the points of the control line. These points are then proportionally adjusted to the shape of the control line. With smooth text, a ruled surface is created from the control lines and the characters are mapped with its parameter space. Thus the horizontal segments of a character are curved and follow the curves of the control lines. The horizontal top and bottom segments of characters, if they exist (as in letters T, E, and F) are positioned exactly on the control line. Note that, while this option can produce some nice effects for logos, it may be memory intensive, when faceted text objects are generated. Figure 4.19.1.6(b) shows how the control line is curved at the area of the character and (c) shows the resulting character shape.

**Figure 4.19.1.6:** 3D text **Adjusted To Line/Curve**.
Note that the Adjusted To Line/Curve option is more restrictive for the smooth text than it is for the facetted.

When the Adjusted To Line/Curve and Standing (Perp To Plane) options are both on with Smooth text, the Preserve Height And Width, Scale Height And Width, and Scale Width Only options cannot be applied. Also, when Between Parallel Lines type of Smooth text is being generated and Adjusted To Line/Curve is on, the control line must be planar and must consist of only one segment. Examples of Smooth text with Adjusted To Line/Curve on and off are shown in Figures 4.19.1.7, 4.19.1.8, and 4.19.1.9.

Figure 4.19.1.7: Text On Line Standing (Perp To Plane) with Adjusted To Line/Curve (a) off and (b) on.

Figure 4.19.1.8: Text Between Parallel Lines with Adjusted To Line/Curve (a) off and (b) on.

Figure 4.19.1.9: Text Between Lines with Adjusted To Line/Curve (a) off and (b) on.
4.19.2 Generation of text

After the desired text type has been selected from the **Text Place Options** dialog, the Text Place tool (usahaan) is selected, and the required graphic input is entered through mouse clicks. The required input per type of text is as follows:

**Plain Text:**
- **Text At Point:** One mouse click for the base point of the placement.
- **Text Between Points:** Two mouse clicks for the base line of the placement.

**Text As Object:**
- **Text At Point:** One mouse click for the base point of the placement.
- **Text Between Points:** Two mouse clicks for the base line of the placement.

**Text As Object Along Path:**
- **On Line:** One mouse click to pick a control line or, if line is preselected, one click to execute operation.
- **Between Parallel Lines:** One mouse click to pick a control line or, if line is preselected, one click to execute operation.
- **Between Lines:** Two mouse clicks to pick two control lines or, if lines are preselected, one click to execute operation.

As soon as the graphic input is completed, the **3D Text Editor** dialog, shown in Figure 4.19.2.1, appears on the screen. In it, type the text and select the options you desire. As soon as you click on **OK**, the text is generated.

![3D Text Editor dialog](image)

**Figure 4.19.2.1:** The **3D Text Editor** dialog.
At the top of the dialog are four pull down menus containing the usual text parameters: **Font**, **Size**, **Style**, and **Justification**. Examples of pull down menus are shown in Figure 4.19.2.2.

![Figure 4.19.2.2](image)

*Figure 4.19.2.2: (a) The Font menu. (b) The Size menu. (c) The Style menu. (d) The Justification menu.*

The content of the **Font** menu depends on the fonts you have loaded on your machine. A directory of the fonts available is built by the text placement procedures of form•Z at the time the first string of text is placed. Consequently, especially if you have a lot of fonts installed on your system, you will notice some time delay the first time you place text. Since the fonts directory is built only once per session, there will be no delay when you place more text strings.

The size of the 3D text is controlled by the **Height** parameter, found lower in the **3D Text Editor** dialog (discussed below). The sizes selected from the **Size** menu are only useful for defining relative sizes of text, when different sizes are mixed in the same string. For example, a string may consist of certain words of size 12 points and certain words of size 8. When 3D text is generated, the size of the 12 point characters will be determined by the value in the **Height** field and the 8 point characters will be sized proportionally.

The styles in the **Style** menu depend on the currently selected font. For example, those shown in Figure 4.19.2.2(c) are for TrueType Times.

The four options in the **Justification** menu apply to all fonts, and determine how the text will be lined up relative to the base point or line. They only affect the **Plain Text** and **Text As Object** types. The **Text At Point** placement can be **Left**, **Center**, or **Right** justified. The **Fit** (forced justification) option does not affect it. All four options affect the **Text Between Points** method of placement. The **Justification** options are ignored by most of the **Text As Object Along Path** types, which typically force justify the text between the two end points of the control lines. The only exception is placing **On Line** with the **Preserve Height & Width** option.
Next in the 3D Text Editor dialog is a text window in which the text to be created is typed. This text can be scrolled up and down and can be edited using the standard text edit operations. The currently selected font, size, style, and justification is reflected in the text typed in the text window, except for the Fit justification option which shows the text centered.

For Plain Text and Text As Object, one or more lines of text can be typed and generated, including blank lines. For Text As Object Along Path, only one line at a time can be generated. If more lines are typed for this type of text, only the first line will be used for the generation of the text string and the remaining will be ignored. When multiple lines are entered, the height and depth justification options (see below) apply to the complete block of text.

Under the text window there are three tabs containing a variety of options which determine the exact shape, size, and type of text, as follows:

The Options tab

Model Type: As in the Text Place Options dialog, either Facetted or Smooth can be selected and the respective type of text is generated.

Topology Type: This is a pull down menu, shown in Figure 4.19.2.3, and the object types it contains apply to the Text As Object and Text As Object Along Path types of text only. This menu appears dimmed when Plain Text is created. The object types available correspond to the general object types of form•Z. A text object can be a Surface object (one side visible only) or a Solid (extruded object). All text objects can be controlled objects or plain objects, which is determined by an option discussed below.

Height Base: This is a pull down menu, shown in Figure 4.19.2.4, and determines which part of a text string or text entity will be used as a basis for calculating the heights of the characters. The text entity selected is scaled to the applicable height. For all types of text, except for the Between Lines text, the height is given by the value entered in the Height field (see next paragraph). For the Between Lines text, the height depends on the distances of the respective portions of the control lines. The scaling factor used for this adjustment is then applied to all the other characters in the string. This preserves the original proportions of the individual characters in the string. Any of the following can be selected as a Height Base:

Font Height: When this option is selected, the height of the selected font, contained in the internal representation of the font, is used as a basis.
Uppercase, Lowercase: When one of these options is selected, the actual height of either an uppercase or a lowercase character of the selected font is used as a basis.

Character...: Selection of this option invokes the Custom Text Height dialog shown in Figure 4.19.2.5, where a character is typed. The actual height of this character determines the height base.

String Height: When this option is selected the maximum height in the given string is used as a height base.

Height: The value entered in this field determines the height that will be applied to the text entity selected as a Height Base. The other characters are scaled proportionally. This value also determines the distance at which the parallel line is generated for the Between Parallel Lines type of text. This value has no effect on the Between Lines type of text since the heights of its characters are determined by the relative positions of the two control lines.

Next to the Height field is the height justification pull down menu, shown in Figure 4.19.2.6. It determines whether the Top, the Bottom, or the Center of the text bounding box will be placed on the placement point or line. For the Between Parallel Lines type of text, this option determines on which side of the selected control line (one or two) parallel lines will be generated. This pull down menu is dimmed when text Between Lines is generated, since it has no effect on it.

Depth: The value in this field applies to Solid text objects only and corresponds to the “height” value used for the extrusions of the text outline. It is dimmed and unavailable when Surface or Surface Solid is selected.

Next to the Depth field is the depth justification pull down menu, shown in Figure 4.19.2.7. It determines whether the Front, Middle, or Back of the extruded solid text will be placed on the placement point or line. This pull down menu is dimmed and inactive when Surface or Surface Solid text is generated.

Width: This field contains a percentage used for the determination of the widths of characters. The default value is 100% which produces characters that retain the height/width proportions contained in the internal representation of the particular font. Changing this value to a different percentage allows you to manipulate the widths of the characters.
**Leading:** This field is a percentage which controls the space between lines, when multiple lines of text are generated. Its value represents a percentage of the height entered in the **Height** field.

**Outline Smoothness:** This bar controls the resolution of the curves in the text. This is determined by positioning the movable marker between the two ends of the bar, labeled **Low** and **High**. At the lowest end two segments will be generated for each quarter-circle arc segment, while at the highest end 48 segments will be generated. While significantly meaningful for Facetted text, it does not apply to smooth text since the latter is resolution independent and is inherently smooth to a maximum. Consequently, this option is dimmed for Smooth text.

**Plain Object:** By default, text objects are controlled objects, which allows them to be edited and changed after their initial creation. Which parameters of the text can be changed depends on the type of text. Selection of this option produces plain objects rather than controlled objects. The resulting plain objects can be manipulated as any other object (for example, moved, scaled, Booleaned, etc.) but cannot be regenerated using their control parameters (i.e. cannot be edited as controlled text objects).

**Per Character:** When **Plain Object** is selected, you can also instruct the program to create one object per character. Depending on what your needs are, this may offer you more flexibility with respect to further manipulating individual characters. This option is not available when controlled text objects are created.

**Standing (Perp To Plane):** When this option is selected, text perpendicular to the reference plane is generated, as opposed to text parallel to the reference plane, which is generated otherwise. Examples of both cases are shown in Figure 4.19.2.8. This option only affects the **Text As Object** type of text. It is dimmed when a plain type of text is selected.
Figure 4.19.2.8: (a) Flat text and (b) standing text.
(1) Text placed with all the angle values set to 0. (2) Text placed with String Angle = 15°.
(3) Text placed with Character Angle = 30°. (4) Text placed with Incline Angle = 30°.

For some types of text the text string can be generated at a slope other than the horizontal, individual characters may be rotated about their center lines, or the characters may be inclined. These optional transformations are applied when a value (other than the default 0) is entered in the following angle fields. These values represent angles in degrees.

**String Angle:** The value entered in this field represents the angle by which the whole string will be rotated. The center of rotation is at the placement point. Thus left justified strings are rotated about their beginning, centered strings about their midpoint, and right justified strings about their end. The angle is measured from the horizontal direction. This option only affects the Text At Point type for both Plain Text and Text As Object. It is dimmed and unavailable for all other types of text. Examples of rotated text strings are shown in Figure 4.19.2.8(2a) and (2b) for flat and standing text, respectively.

**Character Angle:** When this value is not zero, each character is rotated about an axis passing through its middle. The direction of the axis of rotation follows the position of the text and is different for flat and standing characters. Examples are shown in Figure 4.19.2.8(3a) and (3b). This option is only available for Text As Object placed At Point.

**Incline Angle:** When this value is not zero, each character is rotated about its base line, which produces inclined characters. Examples are shown in Figure 4.19.2.8(4a) and (4b). This option is only available for Text As Object placed At Point.

Note that the angle parameters are not mutually exclusive and may be combined on the same string of text.
The Rounding Options tab

When the type of text object selected for the Text As Object and Along Path types of text is Solid, then the text can also be rendered using settings from this tab, which displays different options depending on whether the text object is faceted or smooth. The Rounding options for faceted objects shown in Figure 4.19.2.9. You can round or bevel the boundaries of the Front, Back, or both faces of each text character by selecting the proper option in the respective column. Each option is complemented by numeric parameters which determine the extent of the rounding. Note that many text characters of most fonts contain sharp corners which do not allow large rounding radii. Consequently, relatively small values are recommended for the rounding radii and bevel offsets. An Adjust Radius/Offset option is also available. When selected, the system will adjust the specified rounding parameters up to a maximum 25% of the original values. Examples of normal, rounded, and beveled characters are shown in Figure 4.19.2.10.

Figure 4.19.2.9: The Rounding Options tab for faceted objects.

Figure 4.19.2.10: (a) Normal, (b) Beveled, and (c) Rounded text.

Figure 4.19.2.11: Rounding smooth solid text: (a) None, (b) Circular, (c) Elliptical, and (d) Bevel rounding.
Note that, depending on the item selected from the **Edge Type** pop up menu, different sets of options are displayed. When **Smooth** is on, the smooth rounding options are displayed, as shown in Figure 4.19.2.12. These are the same rounding options as for the general smooth plain rounding and are discussed in section 4.12.1. They are illustrated in Figure 4.19.2.11.

![Figure 4.19.2.12: The Rounding Options tab of the 3D Text Editor dialog displaying the options for (a) None, (b) Bevel, (c) Circular, and (d) Elliptical rounding.](image)

The **Display Resolution** tab

This tab contains the standard facetting options for smooth objects that control the number of facets and iso lines created.
4.19.3 Editing text and its control lines

Text Edit

After a text string has been generated as an object, it can be edited and changed, provided it is a controlled object. Editing is executed using the Text Edit tool, found on the 9th row of the modeling tool palette.

While the editing of Facetted text is generally interactive, it is not when Smooth text is edited. This is because the Smooth text generating procedures are computationally intensive and require some noticeable time to complete. This is especially true when rounding is also applied. The Smooth text object is regenerated as soon as the editing operation is completed.

To edit previously created text objects, select the Text Edit tool and click on any of the characters of the text string. The 3D Text Editor dialog is presented. This is the same dialog you use when you first create the text. The parameters used for the creation of the text appear in the dialog’s fields. You can change any of them and then click on the OK button. Your text will be regenerated to reflect the parametric changes you just made.

When placing text using the On Line, Between Parallel Lines, or Between Lines option, any control curve object that is selected is maintained with the text object. To edit the curve object, the Edit Controls tool is used. For text controls, the Edit Controls tool works in the same manner as editing the source object of other derivative objects.

For example, it is possible to select an arc as the line object for Text On Line and edit the radius parameter of the arc with the Edit Controls tool. The text will automatically follow the adjusted arc size. The line used for the placement of text may be a vector line or one of the smooth curve types, such as a nurbz, tangent, spline, or arc curve. Simply select the line you desire and its type will be maintained when the text is generated. After the generation the line can be edited according to its type.
4.19.4 Font files and utilities

The fonts listed in the Font menu of the 3D Text Editor dialog are the fonts that have been installed on the computer. In addition to the installed fonts, PostScript fonts that are in directories specified in the PostScript Font Folders section of the Font Options (Preference) dialog (see section 3.2.7) are available when the PostScript font type is selected in the 3D Text Editor dialog. Additional fonts can be added using font utilities, as discussed below.

There are four types of font files that are used by form•Z: bit mapped, TrueType, PostScript and stick fonts. Each type of font is stored in a separate type of font file. Stick fonts are used exclusively by the drafting environment and are discussed in subsection 5.7.2. The stick font files are a custom format used by form•Z. These files are not affected by font utilities and must be stored in the form•Z application’s directory.

Font files on the Macintosh

On the Macintosh, the fonts are installed in the Fonts folder inside the system folder. Font files are recognized by the file’s type and icon. Typical icons used for the types of fonts are shown in Figure 4.19.4.1. Bit mapped font files contain single size and style representations of a font. The names of these files often contain a reference to the size and style of the bit map font that they represent (see Figure 4.19.4.1(a)). A TrueType font file contains an outline or scalable font of a single style. A PostScript font file also contains an outline or scalable font of a single style, however it uses the PostScript language instead of TrueType. When available, form•Z uses a bit mapped or TrueType version of a PostScript font for displaying text in the 3D Text Editor dialog.

A suitcase (not to be confused with the font utility “Suitcase”) is a special type of Macintosh file that functions a lot like a folder. It is used to store system resources such as fonts, sounds, and desk accessories. Bit mapped and TrueType font files can be stored in a suitcase, but PostScript files can not. PostScript fonts will also be recognized when they are placed in a folder containing an open font file or suitcase. This follows the same rule that Macintosh PostScript printer drivers use when looking for a PostScript font file to download text.

Figure 4.19.4.1: Icons of font files: (a) bitmapped fonts, (b) TrueType fonts, (c) PostScript fonts, and (d) suitcases.
Font files on Windows

On Windows, the fonts are installed in the fonts directory inside the windows directory. Font files are recognized by the file’s 3 character extension. The name of the file that proceeds the extension, is usually the name of the font. Bit mapped font files have the extension “.fon” and contain various size and style bit map representations of a font. TrueType font files have the extension “.ttf” and contain an outline or scalable font of a single style. PostScript font files have the extension “.PFM” or “.PFB” and an outline or scalable font of a single style, however it uses the PostScript language instead of TrueType. When available, form•Z uses a bit mapped or TrueType version of a PostScript font for displaying text in the 3D Text Editor dialog.

Font utilities

Font utility software, such as Adobe Type Manager (ATM), Suitcase and Now Utilities, allows the user to customize the font environment by selecting which fonts are available to be used. This includes the installed fonts as well as additional fonts stored in other directories. Consult the documentation for the font utility software that you are using for details on their use.

When you use a font utility and change the font environment while form•Z is running, the changes will be recognized and the font menus will be updated. We recommend against disabling or turning off fonts that you have used in a project. When you do, and plain text is encountered, an error message appears immediately and warns you that the font is no longer available and that the text will be redrawn with a default font. When the disabled font was used for the generation of text objects, the warning does not appear immediately. It appears when you use the Text Edit or Text Line Edit tool to edit the text object.

Since some font utilities have no way of communicating to form•Z that the font environment has changed, form•Z checks the font environment every time you switch back to form•Z from the Finder, a desk accessory, or any other application. If you have an excessive number of fonts, you may notice a delay when switching back to form•Z. To avoid this delay, keep available only the fonts that you are actually using. If for some reason you notice that the font menu in the Text Editor dialog does not match the fonts you have made available, you can force form•Z to re-initialize the font menus. To do so, with the Text Place or Text Edit tool active and while pressing the option key (MacOS) or ctrl+shift (Windows), click anywhere in the graphics window.
4.19.5 Searching and replacing text

Text Search and Replace

After a text string has been generated as an object, it can be edited and changed, provided it is a controlled object. One of the ways text strings can be changed is by searching the string and replacing matching text with new text.

The Text Search and Replace tool searches the text strings of all selected objects (or all objects) for text matching the search string. All instances of this text can then be replaced with the replacement string. This tool can also be used to search and change formatting of text objects, such as font, size, and style.

This tool can either search one or more selected objects or all the text objects, which is determined by an option in the Text Search And Replace Options dialog, shown in Figure 4.19.5.1

To search and replace text strings in previously created text objects, select the Text Search and Replace tool and click on any of the characters of the text object. Or, prepick any number of text objects, select the Text Search and Replace tool, then click anywhere in the modeling window.

4.19.5.1: The Text Search And Replace Options dialog.
If Search All Objects is selected, then just click in the model window and all text objects will be searched. No objects need to be picked in this case. The Text Search And Replace dialog is presented (Figure 4.19.5.2). You can select which of the text string or formatting parameters you wish to replace and the string and formatting you wish to replace it with.

**Search:** This box of options contains the specifications of the string to be searched.

**Font:** The font to search for. The content of the Font menu depends on the fonts you have loaded on your machine.

**Size:** The font size to search for.

**4.19.5.2:** The Text Search And Replace dialog.
Style: The font style to search for. The content of this menu depends on the selected font. This option is only available if a font is selected.

Search String: The string to search for.

Ignore Case: When this option is selected, the case of the strings does not need match the case of the search string in order to be replaced. If this option is not selected, the case must match.

Match Whole Word Only: When this option is selected, the search string must not be part of a larger word in the element’s text to be considered a match.

Search All Elements: When this option is selected, all elements are searched. If it is not selected, only picked elements are searched. This item is only available if objects are picked. Otherwise, it is disabled and selected.

Replace: This box of options contains the specifications of the replacement string.

Font: The font to assign to the matching text.

Size: The font size to assign to the matching text.

Style: The font style to assign to the matching text. The content of this menu depends on the selected font. This option is only available if a font is selected.

Replacement String: The text to replace the matching text with.

Found Text For: Displays the element’s text with the current selection highlighted. Pressing the replace button will replace the highlighted text with the replacement string.

After the proper selections have been made in the Search and Replace boxes, the following buttons execute the desired operation.

Find: Find the next occurrence of text that matches the selected search parameters.

Replace: This item replaces the currently found occurrence of text that matches the selected search parameters with the selected replace parameters.

Restart: This item restarts the search from the beginning.

Replace All: This item silently replace all occurrences of text that matches the selected search parameters with the selected replace parameters.

Following are the operations that can be executed with the Text Search And Replace tool:

Replacing the entire text of all selected elements with a new string:

• In the Replace box, select Replacement String and enter the replacement string.
• In the Search box, make sure that Font, Style, Size, and Search String are off.
• Click the Replace All button.

The entire text of the selected elements will be replaced with the new string.
Replacing all instances of a search string in the text of all selected elements with a new string:

- In the **Search** box, select **Search String** and enter the search string.
- In the **Replace** box, select **Replacement String** and enter the new string.
- Click on the **Find** button.

Each instance of the **Search String** will be displayed in the **Found Text** text area.
- Click on the **Replace** button to perform the replacement.
- Click on the **Find** button to find the next instance of the **Search String**.

This method provides a review and selective replacement of the text. If desired, all instances of the **Search String** can be replaced without review, as follows:

- Select the **Replace All** button.

Instances of the search string will be replaced with the new string.

Changing the font, size, and/or style within the text of all selected elements:

- In the **Replace** box, select **Font, Size**, and/or **Style** and select the new font, size or style.
- In the **Search** box, make sure that **Font, Style, Size**, and **Search String** are off.
- Click the **Replace All** button.

The entire text of the selected elements will be set to the new font, size, and/or style.

Replacing all instances of a font, size, and/or style within the formatting of the text of all selected elements:

- In the **Search** box, select **Font, Size**, and/or **Style** and set the font, size, and/or style to search for.
- In the **Replace** box, select **Font, Size**, and/or **Style** and set the new font, size, style.
- Click the **Find and Replace** Buttons or the **Replace All** button.

Instances of the selected combination of the search font, size and/or style will be replaced with the new font, size, and/or style.

Changing the formatting of all instances of a search string within the text of all selected elements:

- In the **Search** box, select **Search String** and enter the search string.
- In the **Replace** box, select **Font, Size**, and/or **Style** and set the new font, size, and/or style.
- Click the **Find and Replace** Buttons or the **Replace All** button.

Instances of the search string, will have the new font, size, and/or style.

If more than one search parameter is selected, all parameters in the object must match in order for the replacement or selection to occur.

If more than one replace parameter is selected, the text matching the search parameters will be replaced by all selected replace parameters.

Note that if only checkboxes in the **Search** section of the **Text Search And Replace** dialog are selected and **Replace** is selected, no changes will be made to the picked text objects.
4.20 3D modeling symbols

Symbols are a method by which groups of entities that need to be repeated in a project or are frequently used in a variety of projects can be stored in libraries and then placed as many times as desirable by referring to a single copy of their representation. In form-Z, both 3D modeling symbols and 2D drafting symbols are available. This section discusses the modeling symbols. The drafting symbols work the same way, except that they have one less dimension (no Z). The drafting symbols are discussed in the respective chapter, which refers to this section for the details.

A symbol needs to be first defined and stored in a symbols library before it can be placed. A symbol definition (or simply definition) is initially constructed using the Symbol Create tool and by selecting the definition’s component objects. New symbol definitions are automatically stored into the active symbol library, the content of which is displayed in the symbols palette.

One of the symbol definitions displayed in the symbols palette is highlighted and is the active definition. Using the Symbol Place tool and by clicking the mouse on the desired location, the active definition can be placed in a project as a symbol instance (or simply instance). Parametric options are available that allow you to scale, rotate, and in general transform the instance before it is placed, using either graphic or numeric methods.

Symbol instances can be edited and their parameters can be changed using the Symbol Edit tool. Using the Symbol Explode tool, symbol instances can be replaced by the component objects that were used to create the symbol definition that it references. This allows you to redesign some or all of the component parts and then use them to create a new symbol definition that may replace the previous.

The form-Z symbol libraries consist of a modeling and a drafting part. Which is currently displayed depends on the type of window that is active. The symbols environment also provides methods for managing symbol libraries and for editing the parameters of the definitions in a library.

Symbols offer several advantages over the use of regular modeling objects. The first is one of convenience, as frequently used configurations of objects can be saved in a library to be easily retrieved and placed in other projects. The second is an efficiency advantage that economizes memory and file sizes, since repetitive instances placed in the same project only need to reference a single representation of the symbol definition. A third advantage is that symbols offer the ability to apply global replacement operations in one step. A fourth advantage is that symbols facilitate the organization of projects, including the ability to nest symbols in a hierarchical manner. It is possible (but not always advisable) to define a whole project as a symbol containing other nested symbols, which would allow alternative designs to be easily paraded by applying instance replacement operations.
4.20.1 Symbol libraries and the Symbols palette

Symbol definitions are stored in symbol libraries, any number of which can be loaded in a **form-Z** project. The Symbols palette is the compact interface of the symbols environment. The names of the loaded libraries appear in the pop up menu located at the top of the Symbols palette (next to the word “Library”). One of the libraries is selected from the menu as the active library, and its name appears at the top of the palette. Examples are shown in Figure 4.20.1.1, where the name “furniture” is displayed at the top of the Symbols palette. This relatively small library will be used throughout the examples of this chapter.

When new symbol definitions are created using the Symbol Create tool they are placed into the active library. The definitions in the active library are listed or shown in the body of the Symbols palette in one of three available formats. You can switch from one format to another by selecting one of the three icons in the lower left of the palette.

- Selecting this icon produces a name list. With this format, the definitions that are contained in the active library are listed by name in the palette (Figure 4.20.1.1(a)).

- Selecting this icon produces a table that contains small graphic representations of the definitions in the active library (Figure 4.20.1.1(b)).

- Selecting this icon produces a table containing larger graphic representations of the definitions and their names (Figure 4.20.1.1(c)).

In all three formats, one of the definitions in the palette can be highlighted and is the **active** definition. You can select another definition as active by clicking on its icon or name. The active definition is referenced by the Symbol Place tool.

The symbol definitions displayed in the palette may be sorted in either one of two ways: **alphabetically** (by name) or by **chronological order** (new symbols are inserted at the end of the list). The type of sorting depends on whether the **Alphabetical Sorting** option is selected in the **Symbol Libraries** dialog, discussed in more detail in the next section.

*Figure 4.20.1.1:* The Symbols palette: definitions displayed (a) by name, (b) by small icon, and (c) by large icon.
As have all the other palettes in form-Z, the Symbols palette has the standard window features. It can be closed, resized, and scrolled vertically. When closed it can be opened by selecting Symbols from the Palettes submenu in the Windows menu. When resized, the palette increases/decreases at increments depending on whether the definitions are viewed by name, by small graphic, or by large graphic.

Each symbol definition can consist of up to three essentially independent definitions, called levels. The levels are intended to be used for storing symbol definitions at different levels of detail. When placing a symbol instance, definitions with less detail may be used in small scale drawings, while definitions with more detail may be used at large scales. Which level is currently active is determined by selecting one of the level icons found next to the format display icons at the bottom of the Symbols palette.

Selection of one of these icons makes the respective level the active level, and the symbol definitions of that level are shown or listed in the Symbols palette. When a definition is not defined for the active level, it can not be shown graphically in the palette. In these cases, a gray square replaces the graphic representation, and the names of these definitions appear in italics (see Figures 4.20.1.2(b) and (c)). When the name list format is used, the names of definitions not defined for that level are shown in italics and cannot be selected (see Figure 4.20.1.2(a)).

When using the Symbol Create tool to generate a symbol definition, the new definition is stored in the level that is currently selected as the active level in the Symbols palette. Similarly, when placing a symbol instance, the definition at the level that is active in the Symbols palette will be referenced.

At the bottom of the Symbols palette, next to the format display icons, the name of the active definition is displayed. This is useful when using the small icons format that contains no names, and when, due to scrolling, the active definition is not shown in the palette body.

Figure 4.20.1.2: Missing symbol definition levels displayed (a) by name, (b) by small icon, and (c) by large icon.
4.20.2 Managing symbol libraries

Symbol libraries and symbol definitions are manipulated in the **Symbol Libraries** dialog, shown in Figure 4.20.2.1, that can be invoked by clicking in the top row of the Symbols palette (where the word “Library” is, but not in the pop up menu), or by selecting the **Symbol Libraries...** item from the **Options** menu.

The **Symbol Libraries** dialog consists of two main parts: the symbol libraries and the symbol definitions area. The upper left part contains a scrollable area that lists the names of the currently loaded libraries, and button commands that allow you to manipulate the libraries. The active library is highlighted. The right part contains an area that corresponds to the Symbols palette, and displays the symbol definitions in the active library with the active symbol highlighted. The dialog also contains buttons and options that can be used to manipulate the symbol definitions.

If a library is locked, a $\Box$ marker is displayed to the right of the library name. New definitions cannot be added to a locked library, which is when the library file cannot be changed. For example, a library loaded from a read only CD will be labeled as locked.

Symbol definitions can be copied from one library into another, or their relative position in the display window can be changed, using graphic operations. Clicking and dragging the mouse will rubber band the outline of the selected definition icon. If the cursor is located in the library window when the mouse button is released, the dragged definition is copied into the highlighted library.

If the cursor stays within the definition area, the dragged definition icon is repositioned in the definition list. The new position of the dragged definition icon is the location of the last highlighted definition icon when the mouse button is released. The definition icon at this location will be moved one position to the front if the dragged definition icon is moved backwards, or one position to the back if the dragged definition icon is moved forward. If the cursor is in neither the library nor the definition view area when the mouse button is released, no operation is executed. If a library is locked, the position of definition icons in the library cannot be changed, and definitions cannot be copied into the library.

![Figure 4.20.2.1: The Symbol Libraries dialog.](image)
Manipulating symbol libraries

**New...**: When you click on this button, a new, initially empty library is created. The standard File Save dialog appears which allows you to name the new library and to select the destination folder. Selecting **Save** adds the new library to the active project’s list of libraries, as the active library.

**Save As...**: Clicking on this button invokes the standard File Save dialog which allows you to save a library under a different name.

**Save A Copy As...**: Clicking on this button invokes the **Symbol Library Save Options** dialog shown in Figure 4.20.2.2, which contains options for saving in different versions, and all or only used symbol definitions.

**Save As Version**: This pop up menu allows you to select a form·Z file version other than the current version, which is the default. The oldest version to which it is possible to backsave is version 2.8.0. If there are incompatibilities of data between versions, alerts are posted to inform you that you may lose some information. The pop up menu lists only versions in which the file format actually changed. form·Z file formats between the listed versions are always compatible with the immediately preceding file version.

**Save All Symbol Definitions**: When this option is selected, the complete symbol library, with all its symbol definitions regardless of whether they are used or not, is copied. This is the default option.

**Save Used Symbol Definitions Only**: When this option is selected, only symbol definitions that are used in the active project are copied into the saved symbol library.

When you click on the **OK** button of the **Symbol Library Save Options** dialog, the standard File Save dialog is invoked, where you can select the location where the library will be saved and type the name under which it will be saved.

**Load...**: Clicking on this button invokes the standard File Open dialog which allows you to select a library stored on disk. When you click on **Open**, the selected library is loaded and made the active library. All symbol definitions in that library are displayed in the definition area on the right. If a library with the selected name is already loaded, an alert dialog appears which asks you whether you wish to replace it. Clicking **Cancel** ends the load operation. Clicking **OK** replaces the already loaded library and all of its definitions with those in the selected library.

**Unload...**: Clicking on this button unloads the active library and all its definitions. Since this cannot be undone, a warning dialog appears to confirm the operation. All symbol instances which reference definitions in the unloaded library can no longer be properly displayed. A bounding box is drawn instead.
**Info**: When you click on this button, the **Symbol Library Info** dialog, shown in Figure 4.20.2.3, appears. It displays general information about the active library.

**Overview**: Clicking on this button invokes the **Overview Symbol Library** dialog (Figure 4.20.2.5) which displays the symbols in the active library in a uniform scale (see next subsection).

**Compact Library**: Clicking on this button invokes the **Symbol Library Compact Options** dialog (Figure 4.20.2.4).

**Free Memory Of Unused Definitions**: When a symbol is placed for the first time, its definition is brought into memory where it is kept even after a placed instance is deleted. Selection of this option erases from memory all the symbol definitions that are no longer referenced by a symbol instance. The memory recovered is displayed in a message.

**Clean Deleted Definitions From File**: When symbol definitions are deleted (see below), they are not erased from the disk file but are simply marked for deletion. Selecting this option clears the symbol definitions in the active library that have been marked for deletion. The file space saved by this option is displayed through a message. When the active library is locked or there are no deleted definitions in the active library, this option is not available.
Viewing symbol libraries

As mentioned in the previous subsection, clicking on the **Overview**... button of the **Symbol Libraries** dialog invokes the **Overview Symbol Library** dialog (Figure 4.20.2.5), which is used to overview the symbols in a library using a uniform scale. That is, rather than fitting the symbol images to the available spaces, as done when displaying symbols in their palette and dialog, the symbols are displayed in their relative sizes using a uniform scale for all of them. The **Overview Symbol Library** dialog consists of a display window (left), and a number of buttons and fields (right).

![Figure 4.20.2.5: The Overview Symbol Library dialog.](image)

**Color Preview**: These options determine what color will be used to display the symbol images.

- **Active**: When this option is selected, all symbol definitions are displayed in the active color.

- **Best Matching**: When this option is selected, all definitions are drawn in their own colors. If a particular color is not present in the current palette, the color that best matches it is used.

**Display Scale**: The value entered in this text field determines the size at which symbols will be displayed in the **Overview Symbol Library** dialog.

**Layout**: This pop up menu contains five options that determine the shape of the table within which the symbols will be displayed. For example, the default 2 x 2 is as shown in Figure 4.20.2.5.

**Detail**: This pop up menu contains three options that set the level of display detail. Initially, the level that is active at the time the **Overview Symbol Library** dialog is invoked is displayed.

**Show Origin**: When this option is selected the position of the origin of each symbol is marked with a cross using the project’s highlight color.

**Show Handle Point**: When this option is selected the position of the handle point of each symbol is marked with a cross using the project’s ghost color.
Manipulating symbol definitions

The buttons on the lower right portion of the Symbol Libraries dialog (Figure 4.20.2.1) allow you to operate on the symbols displayed in the Symbol Definitions window.

**Display:** Selecting Modeling or Drafting from this pair of options determines whether the modeling or drafting definitions of the active library are displayed. When the Symbol Libraries dialog is invoked, this option is set to the environment of the active window.

**Edit...:** When you click on this button the Symbol Definition Edit dialog, shown in Figure 4.20.4.1, is invoked and the active symbol definition can be edited and changed, as discussed in section 4.20.4. The Symbol Definition Edit dialog can also be invoked by double clicking on a symbol icon or name displayed in the Symbol Libraries dialog or in the Symbols palette. When a library is locked, this button changes to View..., which also invokes the Symbol Definition Edit dialog, but only for viewing purposes, and no changes can be made to the symbol definition.

**Alphabetical Sorting:** When this option is selected, the definitions in the active library are sorted alphabetically. When a new definition is created, it is inserted in the symbols list according to the alphabetical order of its name. When this option is not selected, all new definitions are appended to the end of the list. In the latter case, the symbols are sorted according to the order by which they were created, provided the user does not change that order by moving symbols to other positions. When the position of a symbol definition is changed by dragging it to a new position, this option is automatically deselected.

**Delete...:** Clicking on this button deletes the active symbol definition. Since this operation cannot be undone, an alert dialog appears to confirm the action. If the active definition has more than one defined level, the warning dialog allows you to choose between deleting the current level of the active definition or the entire definition (see Figure 4.20.2.6(a)). If only one level exists in the definition, the warning dialog shown in Figure 4.20.2.6(b) is displayed. Once deleted, all symbol instances that reference the deleted definition are displayed as bounding boxes.

Deleted definitions are not immediately erased from the library file, but only marked as deleted. The Compact Library... button can be used to eliminate the deleted symbols from the library file, as discussed earlier in this section.

**Duplicate...:** Clicking on this button produces a duplicate copy of the active symbol definition. The Name New Definition dialog (Figure 4.20.2.7) appears in which the name of the new symbol definition is typed.

*Figure 4.20.2.6:* Alert dialogs for when deleting a definition.

*Figure 4.20.2.7:* The Name New Definition dialog.
4.20.3 Creating symbol definitions

This tool creates a symbol definition from one or more modeling objects and/or other symbol instances. Light sources can also be included in symbol definitions together with objects or by themselves. The inclusion of light sources allows symbols to be used very effectively for representing light fixtures, by describing not only the shape of a light but also its emission parameters.

Both the prepick and postpick methods can be used for the creation of symbol definitions. However, the postpick method allows you to pick only one entity and since most symbols consist of more than one, the prepick method is expected to be used most frequently. In order to create a new symbol definition, there must be an active library. If no active library exists, an existing library must be loaded or a new library must be created first, as discussed in the previous section.

With the prepick method, with topological level at Object or Group, you use the Pick tool to pre-select any number of objects or other symbol instances. You then activate the Symbol Create tool and click the mouse in the graphics window. Where you click the mouse may be significant, depending on the option preselected for the creation of the symbol, as discussed below. As soon as you click the mouse the Name New Definition dialog appears. Enter a name of at most 31 characters in its text field and click OK. Your symbol is constructed, is stored in the active library, and its image, or name, or both appear in the Symbols palette as the active symbol.

When using the postpick method, you activate the Symbol Create tool and then click on the object. With this method you can only define symbols that consist of a single object. You are prompted to enter a name, and the symbol is constructed and stored as with the prepick method.

The options that apply to the creation of symbols are selected from the Symbol Create Options dialog (Figure 4.20.3.1) that can be invoked directly from the Symbol Create tool.

Associated with each symbol definition is a symbol origin or reference point and a handle point. These points are used when placing a symbol instance, as discussed in section 4.20.5. Their positions are defined when creating a symbol definition. How they are defined depends on the Origin/Handle Definition option selected.
**Reference Plane Origin:** When this option is selected, the origin of the current reference plane is used as the origin of the symbol definition. The handle point is placed at the corner of the box that bounds the objects selected for the symbol, which is furthest away from the origin.

**Centroid Of All Picked Objects:** When this option is selected, the origin of the symbol definition is determined by computing the average of all points of the selected modeling objects. The handle is defined as in the previous option.

**Pick Origin:** This option is the default. When selected, the origin of the symbol definition is determined by the position where the mouse is clicked when creating the symbol definition. The handle is defined as in the previous option.

**Pick Origin and Handle:** When this option is selected, the origin is defined as in the previous option. The handle point is also defined graphically by one additional click of the mouse. Note that this option requires one more mouse click than the previous options.

When you create a symbol definition it will be stored in the active level selected in the Symbols palette. By repeating the symbol creation process three times, each time selecting objects with different details, you can store symbol definitions in each of the three levels.

**Status Of Picked Objects:** These two options determine what will be done with the picked objects.

**Replace With Instance:** When this option is on, the picked objects are replaced with a symbol instance, which references the new definition.

**Use Status Of Objects:** When this option is on, the setting in the **Status Of Objects** tab is used.

When creating a symbol definition, if you enter a name and the active level is set to a level which already contains a symbol definition, the alert in Figure 4.20.3.2 will appear. If you accidentally entered the same name or selected the wrong level, clicking on **Cancel** will void the symbol definition creation operation. If you truly wish to replace the previous definition, click on **OK**.

When you replace the previous definition, you have the option to preserve its handle point. You do so by selecting the **Keep Handle From Existing Definition** option. The previous handle is preserved in a relative fashion. That is, the position of the previous handle relative to the previous origin is used to calculate a new handle position relative to the new origin.

When a previous symbol definition is overwritten by a new definition, all its instances that have been placed will display the image of the new definition.
Figure 4.20.3.3 illustrates the process of creating a new symbol definition. Recall that symbol library “furniture,” used in the examples of this chapter, has contained five symbol definitions until now. With the following example we shall add a sixth symbol definition.

- Generate four long cuboids (legs) and a circular extrusion (table top), roughly as shown. Place the table top so that it touches the tops of the legs.

- Select the desired option for determining the position of the origin and handle points in the Symbol Create Options dialog. In this example, assume that the default Pick Origin is selected.

- Use the Pick tool to preselect the five objects created above by clicking on points 1, 2, 3, 4, and 5, as shown in Figure 4.20.3.3(a).

- With the Symbol Create tool active, click on point 6.

This defines point 6 as the origin of your symbol definition. As soon as the mouse button is released, the Name New Definition dialog appears in which the name for the new definition is typed (Figure 4.20.3.3(b)). After clicking on OK, the definition is added to the active library and its icon appears in the Symbols palette. This is now the active symbol.

If the Alphabetical Sorting option is selected in the Symbol Libraries dialog, the symbol will be displayed between existing symbols (if any) in a position determined by its name. If Alphabetical Sorting is not selected, then the new symbol will be placed and displayed at the end of the existing symbols. This is the case illustrated in Figure 4.20.3.3(c).

Figure 4.20.3.3:
Creating a new symbol definition:
(a) Preselecting its components and determining the position of the origin.
(b) Entering its name.
(c) The new symbol appended at the end of the palette.
Attributes used by a symbol definition

In a form•Z project, attributes, such as surface styles or layers, that are used by symbols contained in the project are constrained from being changed. That is, no changes can be made to these attributes. This is indicated by a read only mode in the respective palettes. For example, the surface styles that have been used for components of symbols are shown in the Surface Styles palette dimmed and with a read only marker, as shown in Figure 4.20.3.4.

When double clicking on a read only surface style, the Surface Style Parameters dialog is opened as usual, but all fields and menus are disabled, as shown in Figure 4.20.3.5. No changes can be made to the parameters in this dialog, but are simply displayed for information purposes. However, they can be assigned to objects as any other surface style and they can be copied so that variations can be derived from them.
4.20.4 Editing symbol definitions

There are two different types of editing that can be applied to symbol definitions after they have been created and saved. One edits and adjusts the different parameters of a symbol definition, including its name and its placement parameters. The other allows you to edit and revise the actual shape of the components of a symbol. They are both discussed in this subsection.

Editing the parameters of a symbol definition

The placement parameters (origin and handle) of a symbol definition can be edited in the Symbol Definition Edit dialog (Figure 4.20.4.1), invoked by (i) double clicking on a symbol display in the Symbols palette, (ii) double clicking on the symbol display in the Symbol Libraries dialog, and (iii) clicking on Edit... in the Symbol Libraries dialog. If a library is locked, Edit... in the Symbol Libraries dialog is renamed to View.... In this case the parameters of the symbol definition can only be viewed and not changed.

The Symbol Definition Edit dialog consists of a viewing window, a number of button commands, and a few editable text fields. A 3D view of the symbol is displayed in the window, which can be changed using operations that are activated by selecting one of the icons found under the view window. These are the standard operations available in preview dialogs (see subsection 2.1.6).

The view is displayed with the coordinate axes that are local to the symbol definition. Their origin is at the origin of the symbol, and they are displayed in the project’s highlight color. The position of the handle point is marked with a 3D cross, and is shown in the project’s ghost color. Both the position of the origin and the position of the handle can be changed.

The position of the origin and/or handle point can be changed graphically or by typing their coordinates using the Move Origin... button and the Handle text fields (see below).
To move the origin or handle graphically, click on them, then move them, then click again to place them in their new position. That is, the click-click method is used.

While moving the point, the cursor automatically snaps to points, segments, midpoints of segments, and to the corners of the symbol’s bounding box. If the current view is an axonometric view, these snap points are all that can be used to position the origin or handle. If the view is an orthographic projection (top, bottom, left, right, front or back view), the origin or handle point can be positioned anywhere. When moving in the top or bottom view, only the x and y coordinates are changed, while the z coordinate stays constant. When moving in the front or back view, the z and x coordinates change, while y stays constant. When moving in the right or left view, only the y and z coordinates change, while x stays constant. When moving in an axonometric view, the coordinates of the moving point are set to those of the snapped point.

**Name:** The name of the definition can be changed by typing a new name in this text field. If the new name is already in use, a warning message will be posted.

**Color Preview:** These two options set the color for displaying the images of the symbols.

- **Active:** When this option is selected, the image of the symbol definition is drawn in the active color of the active project.

- **Best Matching:** When this option is selected, the symbol definition is drawn in its original colors. If a particular color that is used in the definition is not present in the current palette, a color which best matches it is used instead.

**Move Origin...:** When you click on this button the Move Definition Origin dialog (Figure 4.20.4.2) appears. Enter the distance by which the origin is to be moved in its X, Y, and Z text fields and click on OK. The move is executed immediately. The origin of a symbol definition is always the 0,0,0 position. When the origin is moved, all the other coordinates of the symbol definition are changed, including the position of the handle point, whose coordinates are displayed in the Symbol Definition dialog and change each time the origin is moved.

**Handle:** The values in the X, Y, and Z fields for the position of the handle point can be edited and changed to move the handle point. The move is executed as soon as you click on another button of the dialog or you exit the dialog by clicking on OK.

**Notes:** When a definition is first created, the notes contain the generation date and time. The notes can be changed by typing new information in this text edit field.

**Capture Preview:** When you click on this button, the current view angle and color of the definition objects in the preview area are used to create a new image for displaying the symbol definition (when one of the graphic formats is used). The new image is zoomed to fit the size of the display box.
Editing the geometry of a symbol definition

To edit the geometry and attributes attached to a member of a symbol definition, follow these steps:

• Open the symbol library file using the **Open...** command in the **File** menu.

Once the library file is selected in the File Open dialog and **OK** is pressed, a new **form-Z** project is created, whose name is the same as the library file. This special project serves as the environment for editing the definition in the library file. Once the library is opened, the definitions are displayed in the symbols palette as shown in Figure 4.20.4.3.

Note that the pop up menu on the title bar of the Symbols palette, which in a regular project shows the name of the library file, now contains two items: **Place Instance** and **Edit Definition**. They are discussed in further detail later in this section.

![Figure 4.20.4.3: The symbol editing environment.](image)

The objects or elements of the active definition are placed at the world origin and are shown in the project windows. It is now possible to use any **form-Z** tool to make changes to these objects, or elements, add new ones or delete the old ones.

• The changes to the active definition can be made permanent by clicking on another definition in the Symbols palette.
This will invoke a dialog to confirm, that the changes will be saved (Figure 4.20.4.4) when you press OK, the objects or elements currently shown in the project window are used to redefine the active definition using the world origin as the definition origin. If Cancel is selected, the changes are ignored. The old objects or elements are deleted and the objects or elements of the new active definition are shown in the project window.

If a surface style or layer in a modeling definition was edited, and the same surface style or layer also exists in other definitions in the library, it is possible to apply the same change to all these surface styles or layers. If this condition exists a dialog is invoked after the active definition is saved, which inquires whether to apply the changes universally (Figure 4.20.4.5). If you press OK, the changes will also be applied to all the other surface styles or layers. If you click on Cancel, the changes are made only to the active definition.

Symbol definitions may contain other symbols, called nested symbols. When editing a definition, it may be necessary to place such a symbol as part of the changes made to the definition. Selecting the symbol to be placed requires a click on the definition in the Symbols palette. If the menu in the palette is set to Place Instance, this click will select a new symbol for placement. If the menu is set to Edit Definition, clicking on another definition will apply the changes to the active definition as described above.

While editing definitions in a library file, the changes to the file are not permanent, until the Save command in the File menu is executed. The following items in the File menu also operate and have an effect on the symbol library file:

**Save As**, **Save A Copy As**: Save the library file to a file with a new name. The latter item, in addition offers the option to save the library to an older file version.

**Revert To Saved**: Deletes all the changes since the last Save and restores the library file.

**Close**: Closes the library project.

Note that certain items in the menu bar are disabled while editing a library file. These include all radiosity and animation items in the Display menu, the Animation From Keyframes... item in the Views menu, Save QuickTime VR..., Export Animation..., and Import... in the File menu.
4.20.5 Placing symbol instances

To place a symbol, a symbol library must already be loaded. When attempting to place a symbol with no library open, a warning dialog is posted that also makes it possible to load a library and to continue the operation. This is done by clicking on the *Load Library...* button found in the dialog.

**Symbol Place**

This tool can be used to place symbol instances in five different ways. All methods place the origin of the symbol at the click point. They differ in the way scaling factors and rotations are applied to the placed instance, as follows:

1. Numerically defined uniform or independent X, Y, Z scale and numeric rotation.
2. Graphic uniform scale and graphic rotation.
4. Individually controlled graphic scale and rotation.
5. Individually controlled numeric X, Y, Z scale and graphic rotation.

The method used is determined by the option selected in the **Symbol Place Options** dialog (Figure 4.20.5.2), invoked directly from the Symbol Place tool.

**Scale Factor**: Scale factors can be entered in the X, Y, and Z fields to be used when placing a symbol instance using three of the five available placement methods. The other two methods apply a scaling factor graphically. The default scaling factor is 1.000 for all the directions, which results in no scaling. When **Scale Lock** is on, all the directions are locked to the same scaling factor, and a value entered in one of the scale fields is automatically entered in the other fields.

**Pick Origin**: When this option is selected, the symbol instance is placed with one mouse click, which determines the position of the origin of the symbol instance. The scale factors entered in the **Scale Factor** fields and the rotation angles entered in the **Rotation** fields are applied to the symbol instance. Rotation angles are applied relative to the current reference plane. The default rotation angles are 0°, which applies no rotation.
Note that the **Symbol Place Options** dialog does not contain any text fields for entering mirroring (reflection) parameters. However, symbol instances can be mirrored when placed by entering a negative number in the respective **X**, **Y**, or **Z Scale Factor** field. For example, entering -1.00 in the **X** field will reflect the instance along the X direction, and entering -0.50 in the **Y** field will both scale and mirror the instance along the Y direction.

**Pick Origin And Uniform Scale:** When this option is selected, the symbol instance is placed by clicking the mouse twice to input two points. The first click determines the position of the origin. After the first click the instance is rubber banded and follows the motion of the mouse. The complete symbol image or only the bounding box may be rubber banded, depending on how quickly the mouse is moved after the first click. The rubber banded image rotates about the origin and is scaled according to the motion of the mouse. If **option** (Macintosh) or **ctrl+shift** (Windows) is pressed after the first click, the rubber banded image rotates, but its size remains constant. In the latter case, the **Scale Factors** entered in the **X**, **Y**, and **Z** fields are applied. When these keys are not pressed, a uniform scaling factor that is defined graphically by the motion of the mouse is applied in all directions.

The placing of a symbol using the **Pick Origin And Uniform Scale** method is illustrated in Figure 4.20.5.3. A symbol definition is shown in (a). The placement process starts by first clicking the mouse at point 1, as shown in (b). After the first click, the symbol is rubber banded and may be scaled and rotated. After the second click (3), the placement of the symbol is completed (c).

**Pick Origin And X, Y and Z Scale:** When this option is selected, the symbol instance is placed with four input points. The first point determines the location of the instance origin. The second point determines the rotation about the Z axis and, when the option/ctrl+shift keys are not pressed, the X scale factor. If the option/ctrl+shift keys are pressed, the value entered in the **X Scale Factor** field is applied. After the first click, a line that corresponds to the symbol definition’s extent in the X direction is rubber banded and follows the motion of the mouse until the second click.

After the second click, a rectangle is rubber banded and, if the option key is not pressed, it follows the motion of the mouse, applying a scaling factor in the Y direction. If the option key is pressed, then the value entered in the **Y Scale Factor** field is applied and no rubber banding occurs. After the third click, the complete symbol or a bounding box is rubber banded and, when the option key is not pressed, it follows the motion of the mouse, applying a scaling factor in the Z direction. If the option key is pressed, the value entered in the **Z Scale Factor** is applied and no rubber banding occurs. The placement is completed as soon as the mouse is clicked for the fourth time.
Figure 4.20.5.4: Using the **Pick Origin And X, Y And Z Scale** method:
(a) After the first click a line is rubber banded. (b) After the 2nd click a rectangle is rubber banded. (c) After the 3rd click a bounding box or the entire symbol is rubber banded. (d) The placed symbol.

**Color Display:** One of three options for handling the color of an instance may be selected from this pop up menu.

**Active Color:** When this item is selected, the color active in the Color palette is assigned to the placed instance and is used to display it.

**Best Matching Colors:** When this item is selected, the colors of the original objects contained in the symbol definition are retained if the identical colors are available in the Surface Styles palette of the project where the symbol is placed. If the identical colors are not available, colors that are closest to the original colors are used.

**Definition Colors:** Selection of this item preserves the colors in the symbol definition. If the necessary colors are not available in the Surface Styles palette of the project where the symbol is placed, new colors are added to the palette.

**Layer Display:** One of two options that determine which layer the symbol instance is placed on is selected from this pop up menu.

**Active Layer:** When this option is selected, the instance is treated as if all of the objects/elements in its definition were on the active layer, regardless of what layer the objects/elements that are included in the definition were on when the definition was created.

**Definition Layers:** When this option is selected, each object/element in the definition referenced by the instance is treated as if it were on the layer it was on when the definition was created. If the layer of a definition object or element is not in the layer palette, a new layer will automatically be added.

**Use Lights If Present:** When placing a symbol instance, which refers to a definition that contains lights, the instance can use the lights or not. When this option is on and a symbol definition contains at least one light source, a new light is added in the Lights palette. In other words, the symbol instance is both, an object and a light source and can be accessed through the Objects and Lights palette. In the Lights palette, the symbol light is marked with an “S” in the light type column and the name of the light is the same as the name of the object surrounded by square brackets. Double clicking on the symbol light’s name in the Lights palette invokes the **Symbol Instance Edit** dialog, where the instance parameters and light overwrite attributes can be modified.
If the **Use Lights If Present** option is off and a symbol instance is placed, no lights are added, even if there are lights in the symbol definition.

After a symbol with lights has been placed, the lights in the symbol will contribute to the illumination of the scene during a subsequent rendering, just as if the lights would have been added separately.

A symbol light listed in the Lights palette is shown in Figure 4.20.5.5.

An example of placing symbol instances is illustrated in Figure 4.20.5.6. The symbol definitions are selected from the “**furniture**” library we have been using throughout this chapter.

- After invoking the **Symbol Place Options** dialog, make certain that **Pick Origin** and **Active Color** (Color Display pull out menu) are selected. All the **Scale Factor** fields should display 1 and all the **Rotation** fields should show 0.
- Given that the shown grid is the default 2' x 2', set snapping to 6" (in the **Snap Options** dialog) and turn Grid Snap on.
- In the Symbols palette, click on the round table icon (the symbol created in the example of section 4.20.3) to make it the active symbol.
- With the Symbol Place tool active, click on point 1.
- In the palette, click on the chair symbol to make it active.
- Invoke the **Symbol Place Options** dialog and enter 45° in the **Z Rotation** field. Then click on 2.
- Repeat the placement operation three more times. With **Z Rotation** set to 135° click on 3, with 225° click on 4, and with 315° click on 5. The result should be as shown.

---

**Figure 4.20.5.5**: The Lights palette.

**Figure 4.20.5.6**: Placing five symbol instances.
4.20.6 Editing symbol instances

Symbol Edit

The parameters of symbol instances can be edited using the Symbol Edit tool. Both the prepick and postpick methods can be used.

When using the prepick method, with the Pick tool active, preselect any number of symbol instances, then activate the Symbol Edit tool, and click the mouse anywhere in the graphics window. When using the postpick method, activate the Symbol Edit tool and click the mouse on the instance to be edited. The postpick method allows you to edit only one symbol instance at a time.

When the prepick method is used to preselect several symbol instances, these instances can be edited individually or simultaneously. The method to be used is selected in the Symbol Edit Options dialog (Figure 4.20.6.1), which can be invoked directly from the Symbol Edit icon. Both methods invoke a symbol instance parameters dialog where the parametric values of symbol instances can be changed. The dialog for the individual editing also contains a viewing window while the dialog for the simultaneous editing does not.

Edit Individually: When this option is selected and a number of instances are preselected (prepick method), the Symbol Instance Edit dialog shown in Figure 4.20.6.2 will be displayed once for each of the symbol instances to be edited. That is, the editing process parades through all the preselected instances, one at a time, and each is edited individually.
For each instance, the dialog displays the parametric values of the instance being edited, and includes a view of the symbol. Each time the **OK** button is clicked, the parameters of the next instance are displayed, until all the preselected instances have been edited. Selecting **Cancel** stops the parading.

**Edit Simultaneously:** When this option is selected and several instances have been preselected, the **Symbol Instance Edit** dialog shown in Figure 4.20.6.3 is presented only once for all the instances and contains no viewing window. If a particular parameter has the same value in all the preselected instances, the actual value of the parameter is shown in the dialog. If the parameter has a different value in at least one instance, the “***” place holder is shown in the respective field of the dialog.

If the place holder is left in the field, the individual parameters for each of the picked instances will be unchanged. If the place holder is changed to a new value, then that parameter will be changed to the new value in each selected instance.

The two dialogs used for editing instances individually or simultaneously contain the same parametric options, but the individual editing dialog also contains the **Use Lights** group of options and a viewing window and icons that can be used to manipulate the views. These icons are the same and work as in all the preview dialogs (see subsection 2.1.6).

**Origin:** The X, Y, and Z values of the symbol instance origin are displayed and can be changed in these text edit fields.

**Rotation:** The X, Y, and Z rotation factors of the symbol instance are displayed and can be changed in these text edit fields.

**Scale:** The X, Y, and Z scale factors of the symbol instance are displayed and can be changed in these text edit fields.

**Relative To Reference Plane:** This option is only available when editing symbol instances individually. When selected, the **Origin** and **Rotation** values are displayed relative to the current reference plane. Otherwise, these values are relative to the world coordinate system. When editing symbol instances simultaneously, the origin and rotation parameters are always displayed relative to the world coordinate system.

**Scale Lock:** When this option is selected, all the **Scale** factors are locked to the same value. That is, the value entered in one of the fields is also entered in the others automatically.
**Color Display:** One of three options for defining the color of a symbol may be selected from this pop up menu. These are the same options available in the **Symbol Place Options** dialog, except for the first.

**Instance Color:** When this option is selected, the symbol instance is drawn in the color which is currently assigned to it.

**Best Matching Colors** and **Definition Colors:** As in the **Symbol Place Options** dialog.

**Layer Display:** This pop up menu offers two options for defining the layer display attributes of placed symbols.

**Instance Layer:** When this option is selected, the instance is treated as if all of the objects/elements in its definition were on the active layer, regardless of what layer the objects/elements that are included in the definition were on when the definition was created.

**Definition Layers:** When this option is selected, each object/element in the definition referenced by the instance is treated as if it were on the layer it was on when the definition was created. If the layer of a definition object or element is not in the layer palette, a new layer will automatically be added.

**Detail:** The option selected from this pop up menu determines which detail level of the active definition will be displayed by the symbol instance. When a "" is displayed next to the level in the pop up menu, it indicates that no symbol definition exists for this level. If such a level is selected, a bounding box is drawn to represent the symbol instance.

**Library:** This pop up menu contains a list of all currently loaded libraries. When the dialog first appears, the library of the symbol definition referenced by the symbol instance is shown. If a new library is selected from the menu and that library contains a definition with the same name as the definition name shown in the **Definition** pop up menu, this definition will become the new definition referenced by the symbol instance. If no such definition exists, the first definition in the library is used. If a "◊" is shown next to the library name, the library is currently not loaded.

**Definition:** This pop up menu contains a list of all the symbol definitions in the library currently shown in the **Library** pop up menu. When the dialog first appears, the name of the definition referenced by the symbol instance is shown. A new definition can be selected from the menu. A "◊" displayed next to the definition name indicates that the definition does not exist in the library. If such a symbol definition is selected, a bounding box is drawn to represent the symbol instance.
The following options are only available in the **Symbol Instance Edit** dialog invoked when **Edit Individually** is on. They apply when the symbol being edited includes lights.

**Use Lights**: If this option is on, the lights in a symbol are used. This option corresponds to the **Use Lights If Present** option in the **Symbol Placement** dialog. If a symbol was placed with the **Use Lights If Present** option off and the **Use Lights** option in the **Symbol Instance Edit** dialog is later turned on, a new light is added to the Lights palette. Likewise, if a symbol was placed with the **Use Lights If Present** option on and the **Use Lights** option in the **Symbol Instance Edit** dialog is later turned off, the symbol light is automatically removed from the Lights palette. Note that this option is intended to designate a symbol as a light source or not. In order to temporarily turn off the lights of a symbol, the **Shining** attribute of the symbol light should be used. This is described in the next paragraph.

**Visible, Shining, Shadows**: These three options determine the visibility, shining, and shadow attributes of the lights in the symbol. They work in the same fashion as the equivalent attributes in a light group. For example, if a symbol contains three light sources, two of which cast shadows and one does not, the **Shadows** menu will, by default, show “***”. This means, that the individual shadow casting parameter of each light is maintained. If **Yes** is selected from the menu, all three lights will cast shadows. If **No** is selected, none of the lights cast shadows. Likewise, the **Shining** menu can be used to turn all light sources in a symbol on, off, or leave their shining parameters unchanged.

**Override Attributes...**: When this button is pressed, the **Override Attributes** dialog, shown in Figure 4.20.6.4, is invoked. In this dialog, certain attributes of all lights in a symbol can be changed. This dialog is equivalent to the **Override Attributes** section in the **Light Group Attributes** dialog and the options work exactly in the same way. This is described in detail in section 2.10.4. As a matter of fact, one could think of the symbol instance as a light group, collecting the light it contains.
4.20.7 Exploding symbol instances

With this tool, symbol instances can be decomposed into the modeling objects that were used to create the symbol definition referenced by the symbol instance. Both the prepick and postpick methods can be used. When the prepick method is used, any number of symbol instances can be preselected, using the Pick tool. With the postpick method, only one symbol instance can be selected.

When a symbol is exploded, the instance is replaced by copies of the component objects that were used in the symbol definition. These objects are assigned the color and layer attributes of the symbol instance. The symbol definition currently stored in a library is not affected and remains as is.
4.20.8 Symbol definitions with nested symbol instances

Previously placed symbol instances may be nested in a symbol definition. That is, the entities selected for the creation of a symbol definition may also contain symbol instances that were previously placed. These instances must reference symbol definitions in the active library. In other words, the definition containing nested instances and the definitions referenced by these instances must exist in the same library.

When a scale factor is applied to a symbol instance which references a definition with nested instances, the result is different than it would be if the same components of the referenced definition were selected as individual objects when the definition was created. The scaling operation is applied at the lowest level of the symbol definitions and is relative to their local origin. Thus any nested instances are scaled before they are rotated, if a rotation was applied when they were placed. These differences are illustrated in Figure 4.20.8.1.

Recall the dining room set created in the example of section 4.20.5 (see Figure 4.20.5.4): the chairs around the table were placed by applying rotations of 45, 135, 215, and 305 degrees around Z. Assume that a symbol definition has been created that consists of the five instances in the dining room set (one table and four chairs). After this nested symbol definition has been created, it can be placed as any other instance. Figure 4.20.8.1(a) shows the dining room set symbol placed as a single instance, without applying any scaling factor.

Next, the symbol instance is edited using the Edit Symbol Instance tool, and the Y Scale factor in the Symbol Instance Edit dialog is changed to 2.0. Click on OK, and the chairs and the table are displayed as shown in Figure 4.20.8.1(b). All nested symbols have been scaled along the direction of their local Y axis.

Next, assume that the dining room set is exploded and a symbol definition is created by picking all its components as objects. This new definition has no nested instances. Place it after typing 2.0 in the Y Scale Factor field of the Symbol Place Options dialog. The result should be as shown in Figure 4.20.8.1(c). The different effect of the scaling operation, compared to the previous example, should be apparent.

Figure 4.20.8.1: Scaling a symbol instance, referencing a definition with nested instances.
Nested instances can be very useful in constructing symbol definitions with hierarchically structured entities. For example, as illustrated in Figure 4.20.8.2, a definition can be created representing a facade module. Each element of the facade module is a symbol instance referencing definitions with more nested instances. At the lowest level, the symbol definitions may contain basic building elements, such as frames, panels, and supports. Such a method encourages a modular assembly of larger pieces, gives you the flexibility to edit the basic definitions used in the higher level definition, and ensures that the assembly reflects all the changes that may be made to the lower level definitions.

Figure 4.20.8.2: Hierarchically structured (nested) symbol definitions facilitate modular design.
4.20.9 Working with symbols

This section will illustrate some of the practical applications of symbols that allow you to make global changes to already placed instances.

Example 1: Assume that the furniture arrangement shown in Figure 4.20.9.1(a) has been generated by placing each of the nine shown pieces as a symbol instance, and that the symbol definitions are picked from library “furniture.” In this example we shall replace all the “kitchen chairs” which are currently placed around the table, with “dining chairs,” another symbol definition that is also contained in our library.

• Preselect all the “kitchen chairs” instances. This can be done either by using the Pick tool to select each instance individually, or by using the Select By... command in the Edit menu and by setting the selection criteria for symbol instances in the Symbol Instance Selection Criteria dialog.
• Invoke the Symbol Edit Options dialog (double click on the Symbol Edit tool) and select the Edit Simultaneously option.
• With the Symbol Edit tool active, click on the graphics window. This invokes the Symbol Instance Edit dialog, whose Symbol Definition pop up menu displays the name “kitchen chair.”
• From the Symbol Definition menu select “dining chair” and click on OK.

As soon as you exit the dialog, the image on the screen is refreshed and the furniture arrangement is now displayed with different chairs, as shown in Figure 4.20.9.1(b). Since only the symbol definition reference was changed, the rotation, scale, and location of the selected symbol instances remained the same.

*Figure 4.20.9.1:* Changing the symbol definition reference of already placed symbol instances: (a) the original furniture arrangement, and (b) the arrangement after its chairs have been changed.
Example 2: In this example all the symbol instances will be changed by changing the library they reference. To do this we shall first need to load an additional library.

Assume that library “furniture2” is loaded, as shown in Figure 4.20.9.2. It contains three symbol definitions, namely “dining chair,” “table,” and “kitchen chair.” Note that these names are identical with names found in library “furniture.” Also note that the latter contains two additional symbol definitions that are not in “furniture2.”

- Preselect all the symbol instances.
- In the Symbol Edit Options dialog select the Edit Simultaneously option.
- With the Edit Symbol Instance tool active, click on the graphics window. Note that the Symbol Definition pop up menu, in the Symbol Instance Edit dialog that is invoked, displays “***” since instances referencing a variety of symbol definitions have been selected.
- The Library menu displays “furniture.” Change it to “furniture2,” and click on OK.

After the dialog is closed the previously displayed furniture arrangement (Figure 4.20.9.3(a)) is changed to the arrangement shown in Figure 4.20.9.3(b). Note that the chairs and the tables have been changed to the designs contained in “furniture2,” but the lamp has remained as in the previous display. This is because library “furniture2” does not contain a symbol definition with the name “lamp.” When switching libraries and such a situation arises, the reference of the previously placed instance for which there is no definition in the new library remains as is. To properly display such cases, all libraries that are referenced should be loaded.

Figure 4.20.9.2: Loading library “furniture2.”

Figure 4.20.9.3: Changing the definition reference of symbol instances from definitions in one library to definitions with the same name in another library.
Example 3: In this example the level of detail displayed by the symbol instances will be changed. Note that, as shown in Figure 4.20.9.4, library “furniture2” also contains level 2 symbol definitions for “dining chair” and “table.”

The instances shown in Figure 4.20.9.5(a) were initially placed with the Detail parameter (Symbol Place Options dialog) set to Level 1. We shall now change it to Level 2.

• Preselect all the symbol instances.
• In the Symbol Edit Options dialog select the Edit Simultaneously option.
• With the Edit Symbol Instance tool active, click on the graphics window and in the Symbol Instance Edit dialog that is invoked, from the Detail pop up menu, select Level 2 and click on OK.

After the dialog is closed, the symbol instances display the more detailed furniture representations of level 2 (rounded legs, etc.), as shown in Figure 4.20.9.5(b). Note that the lamp is now shown as a bounding box and recall that the lamp instance still references library “furniture,” which does not contain a level 2 definition for “lamp.” When such a case arises, that is, when a level that contains no symbol definition is referenced, a bounding box is displayed.

Figure 4.20.9.4: Level 2 symbol definitions of library “furniture2.”

Figure 4.20.9.5: Increasing the level of detail displayed by symbol instances: (a) Level 1 and (b) Level 2 symbol definitions.
Example 4: In this example the design of one of the symbol instances will be changed by exploding the symbol definition it references, making changes to the design of its components, and replacing the previous symbol definition with a new one.

- With the Explode tool active, click on one of the "dining chair" instances.

Even though no difference is visible in the display of the instance, it has actually been replaced with the modeling objects that correspond to its components. We will next refine some of the details of the objects, as shown in Figure 4.20.9.6(b).

- In the Symbols palette, switch the active level to 2 (select icon 2 if not already selected) (Figure 4.20.9.7(a)).
- Preselect the objects whose design you just refined.
- With the Create Symbol tool active click in the window.
- In the Name New Definition dialog that is invoked, enter “dining chair” and click on OK (Figure 4.20.9.7(b)).
- When the alert dialog appears (Figure 4.20.9.7(c)) and inquires whether the existing level of the definition “dining chair” should be overridden, answer “yes” by clicking on OK.

As soon as you exit the dialog, the objects you had selected are replaced with the newly defined symbol. All the other instances that reference symbol definition “dining chair” are also replaced with the new design, as shown in Figure 4.20.9.8.
4.20.10 Troubleshooting

A symbol definition cannot be created: There is no active library. Create a new library or load an existing library from file, using the New... or Load... command in the Symbol Libraries dialog.

A symbol instance only displays a bounding box: There are four conditions which cause a bounding box to be displayed for a symbol instance instead of the definition geometry. First, the definition referenced by the symbol instance exists, but the detail level set in the symbol instance does not exist in the referenced definition. Edit the symbol instance and set the detail to a level which exists. Undefined detail levels show a “◊” in front of the level name in the pop up menu. Second, the library in which the referenced definition is located exists, but the referenced definition does not exist in this library. When the symbol instance is edited, the definition name shows a “◊” at the beginning. Select a new definition name in the Symbol Instance Edit dialog, or create a new definition with the name of the non existing definition. Third, the library in which the referenced definition is located is not loaded. This can be determined by editing the symbol instance. A “◊” mark is shown in front of the library name in the pop up menu. Load the library file using the Load... command in the Symbol Libraries dialog. Fourth, the Show As Bounding Volume option in the Wire Frame Options dialog is checked, which causes all symbol instances to be displayed as bounding boxes, regardless of whether the referenced definition exists or not. Deselect this option to display symbol instances with the geometry of the referenced definitions.

Some objects of a definition displayed by a symbol instance do not draw, or are drawn ghosted: The Layer Display parameter of the symbol instance is set to Definition Layers. In the Layers palette, the display attribute of the layers referenced by the definition objects is set to “ghosted” or “not visible.” Change the layer attributes to “visible.”

The display attributes of certain layers are set to “ghosted” or “not visible” but objects in definitions located on those layers are still visible when displayed by symbol instances: The Layer Display parameter of the symbol instance is set to Instance Layer. Edit the symbol instance and change the Layer Display parameter to Definition Layers.

A symbol instance does not draw in the colors of the definition objects: The Color Display parameter of the symbol instance is set to Instance Color. As a result, the symbol instance is drawn in the color assigned to the instance when it was placed. Edit the symbol instance and change the Color Display parameter to Best Matching Color or Definition Color.
A symbol instance is placed, but does not draw in the current color: The Color Display parameter of the symbol instance is set to Best Matching Color or Definition Color. As a result, the symbol instance is drawn in the color of the definition objects. Edit the symbol instance and change the Color Display parameter to Instance Color.

An operation refuses to use a symbol instance as an operand: Certain operations require that an operand contains actual data. Symbol instances do not contain any geometric data, but merely display the geometry of the referenced symbol definition. Explode the symbol instance to make a copy of the referenced definition geometry.

A symbol instance cannot be placed using the Pick Origin And Uniform Scale option or Pick Origin And X, Y and Z Scale option: The handle point of the active definition may be identical with the definition origin. Edit the active definition using the Edit... button in the Symbol Libraries dialog and change the handle point. As a rule of thumb, it is best to locate the handle point at the corner of the definition bounding box opposite from the origin.

A symbol instance cannot be placed with any of the placement options: No active library or active definition exists. Load an existing library or create a new library using the Load... or New... command in the Symbol Libraries dialog. If a new library is created, at least one symbol definition in that library must first be created using the Symbol Create item.

A warning message appears, which indicates that a library has been modified from outside the project: Once created or modified, a symbol definition is immediately stored in the library file. Since it is possible that several projects use the same symbol library simultaneously, it is necessary that the changes in the library file made by one project are reflected in the definitions of that library in the other projects. This dialog indicates that at least one definition has been modified. The affected library and all the definitions in that library will be reloaded, and all symbol instances referencing definitions in that library will display the geometry of the updated definitions.
4.21 Editing 3D lines, attributes, and controls

This section discusses tools in four related categories, as follows:

• 3D line edit tools that include:
  • Break Line
  • Close Line
  • Trim Lines
  • Connect Lines
  • Join Lines
  • Fit Fillet/Bevel Lines
  • Insert Point
  • Insert Segment

• Topology adjusting tools that include:
  • Make First Point
  • Reverse Direction
  • Make One/Two Sided
  • Set/Clear Point Marker

• Wire to surface (and vice versa) transforming tool:
  • Cover

• Model type or personality converting operations that are executed with a single tool:
  • Convert

• Transforming the control lines of a parametric object to independent objects:
  • Extract
4.21.1 Breaking a line

This tool is used to break a line at a segment or a point. The segment may be straight or curved. Variations of the operation and parameters are selected from the **Break Line Options** dialog, shown in Figure 4.21.1.1, that can be invoked from the tool.

**At Click:** When this option is selected, the line is broken at the location on a segment or a point, where the cursor is when clicking.

**At Point When Within:** If the click is within a distance from a point that is less than the value entered in this tolerance field, **form-Z** interprets the click to be on the point. If the click is outside the tolerance value, it is interpreted to be on the segment. The default is 0’-1”.

Examples of lines broken at points and segments are shown in Figure 4.21.1.2. Note that offsets are also used, which are set as discussed below.

**With Line:** When this option is selected, one or more lines are broken at their points of intersection with another line. This operation can be executed in either prepick or postpick mode.

When using the prepick method, any number of lines are preselected with the Pick tool. Then with the Break tool active the break line is selected. The preselected lines are broken at the points of intersection with the break line. Examples are shown in Figure 4.21.1.3.

When using the postpick method, with the Break tool active you first click on the line to be broken and then on the break line. The line is broken at the point of intersection. If the two lines intersect at more than one point, they are all broken. An example of the latter case is shown in Figure 4.21.1.4.
Distance: When this option is on some space is added to the left and/or right of the break point. The size of this space can be determined by selecting one of the next two options.

Offset: When this option is selected, the space may be specified independently for the Left and Right sides of the break point.

Length: When this option is selected, the value entered in this numeric field determines the length of the break, where half of the specified length is placed to the left and half to the right of the break point.

The Break tool can also be used to break splines and arcs, as well as composite lines that may include portions of vector lines, splines, and arcs. The two variations of the Break operation (At Click and With Line) are illustrated in Figures 4.21.1.5 and 4.21.1.6, as they are applied to arcs and splines, respectively.
4.21.2 Closing an open line

This tool is used to close an open vector line, using one of three methods that is selected from the Close Line Options dialog (Figure 4.21.2.1), which can be invoked directly from the tool. Either the postpick or prepick selection method can be used to close the ends of an open line.

When you use the postpick method, with the Close Line tool active, click on the line you wish to close. The line should have at least two or three segments, depending on the type of closing operation you wish to apply. With the prepick method, use the Pick tool with topological level set to Object and preselect any number of open lines. Then activate the Close Line tool and click anywhere in the project window. All the lines you selected will be closed. The type of closing is determined by the following options:

Trim: This operation varies by type of line, as follows:
- With a vector line, this operation causes the two end segments of the line to be extended to their point of intersection where they are trimmed and joined (4.21.2.2(b)).
- With arcs, two new segments tangent at the end points of the arc are generated, extended to their point of intersection, trimmed, and joined (Figure 4.21.2.3(1b)).
- The operation is executed the same way for splines, but only after a new point is inserted between the end points of the spline, which effectively divides it into two portions. A single continuous spline cannot be closed (4.21.2.3(2b)).

Join: This operation also varies by type of line, as follows:
- With vector lines, the last point of the line is deleted and the point before it is connected to the first point of the line (Figure 4.21.2.2(c)).
- With arcs, the end of the arc is extended to its initial point. This operation essentially transforms an open arc to a closed circle (Figure 4.21.2.3(1c)).
- With splines, the last control point is joined with the first and the spline is regenerated as when it is drawn closed. Note that this operation causes the shape of the spline to change noticeably (Figure 4.21.2.3(2c)).

Connect: With this option, a new segment is added to the line, starting at its last and ending at its first point. This operation is executed the same for vector lines, arcs, splines, and combinations of these (Figures 4.21.2.2(d) and 4.21.2.3(d)).
4.21.3 Trimming lines

This tool is used to trim segments of lines to their point of intersection with other segments. Three variations of the operation may be selected from the Trim Lines Options dialog (Figure 4.21.3.1), invoked from the tool.

Trim Pairs Of Segments: When this option is selected the trim operation is applied to pairs of segments, which are trimmed to their line of intersection.

The operation is typically executed in postpick mode. With the Trim Line tool active click on two segments. They are immediately trimmed to their point of intersection, if one exists. With the postpick method, the topological level is ignored and the system always picks at the Segment level.

When using the prepick method, any number of segments can be preselected using the Pick tool, with topological level set to Segment. Then, with the Trim Lines tool click anywhere in the project window. The preselected segments will be used by the operation in pairs. That is, the first and second segments will be trimmed to each other, then the third and fourth, and so on until the set of prepicked segments is exhausted. If an odd number is picked, the last is ignored.
Where you click to select the segments to be trimmed is significant, since the click points determine the portions of the segments that will be retained after the trimming. Examples of segment trimming are shown in Figure 4.21.3.2.

When you trim segments that have open ends, you can leave them open, join, round, or bevel them. This is set in the Trim Lines Options dialog. When the ends of the trimmed segments are not open but connect to other segments, then these options do not apply and the trimmed segments preserve their original connections.

**Open**: When this option is selected, the open ends of trimmed segments remain open. This is the default option.

**Join**: When this option is selected, the open ends of trimmed segments are joined and become continuous.

**Fit Fillet**: When this option is selected, the open ends of trimmed segments are joined and the trim point is rounded. The rounding resolution is entered in the # Of Edges field. A length value may be interpreted either as a radius or as a distance, depending on whether Use Radius or Use Distance is selected. The fillet may be generated as a Facetted or Arc entity.

**Bevel**: When this option is selected, the open ends of trimmed segments are joined and the trim point is beveled, according to the following parameters set in the Trim Lines Options dialog.

**Offset: 1st, 2nd**: The values entered in these fields determine how far from the corner the two segments are beveled.

**Length**: When this option is on the value entered in its field sets the size of the bevel segment, whose ends are at the same distance from the corner.

Examples of the different options of segment trimming are shown in Figure 4.21.3.3.

**Trim Ends Of Lines**: When this operation is selected from the Trim Lines Options dialog, the Trim Lines tool can be used to trim two or more lines to the points where their end segments intersect. This operation can only be applied to open lines and can be executed using either the postpick or the prepick method.

With the postpick method, select the Trim Lines tool and, with topological level set to Object, click on the two lines you wish to trim. The two lines will be trimmed at the point where the last segment of the first line intersects the first segment of the second line, provided these segments are on the same plane and are not parallel. Note that, with this operation, the directions of the lines are significant for determining which segment is first and which last. Also, the pick order is significant.
With the prepick method, use the Pick tool to select any number of lines with the topological level set to Object. Then select the Trim Lines tool and click anywhere in the project window to execute the operation. The operation is applied to the lines in the order they were selected. That is, the first line is trimmed with the second, the second with the third, and so on, until the list of lines is exhausted. Note that this operation does not require the selection of only even numbers of lines.

The Trim Ends Of Lines operation may leave the trim points Open or may Join them, according to the selection of the respective options found under it. It is also affected by the Close Line Sequence option. When on, the last segment of the last line will also be trimmed to the first segment of the first line. Examples are shown in Figure 4.21.3.4.

Figure 4.21.3.4: Trimming lines selected in order shown: (a) and (b) with Close Line option on, and (c) off.

Trim Segments With Line: When this option is selected one or more segments are trimmed to a line, called the trim line. Recall that when the Trim Pairs Of Segments variation is used, pairs of segments are trimmed relative to each other and can be joined and rounded. In contrast, the Trim Segments With Line variation may be thought of as being one sided because the trim line acts as a guide only and is not trimmed to any line. This operation is typically useful for lining up the end points of many segments, which is achieved by using the prepick method.

When using the prepick method, any number of segments are selected with the Pick tool. Then, with the Trim Lines tool active you click on the trim line. When using the postpick method, with the Trim Line tool active you click on a segment and then on the trim line. While with prepick many segments can be trimmed, only one can be trimmed with the postpick method. Examples are shown in Figures 4.21.3.5 through 4.21.3.7.

Figure 4.21.3.5: (a) Before and (b) after trimming many segments to a line.

Figure 4.21.3.6: (a) Before and (b) after trimming many segments to a line.
Figure 4.21.3.7: Trimming many segments to a line.
(a) Segments selected where shown. (b) The results of the operation.

The Trim Lines tool can also be used to trim splines and arcs. Examples of the Trim Lines tool as it applies to arcs and splines are shown in Figure 4.21.3.8.

Figure 4.21.3.8: Trimming open (1) arcs and (2) splines by clicking where shown with the white bullets:
(a) Trim Pairs Of Segments, (b) Trim Pairs with Fit Fillet on,
(c) Trim Ends Of Lines with Close Line Sequence on, and (d) Trim Segments With Line.
4.21.4 Connecting lines

The open ends of two segments are connected by generating a new segment between the two open ends. Such operations can be executed with the following tool.

Connect Lines

This tool can be used to execute two variations of a connect operation, which are selected from the Connect Lines Options dialog (Figure 4.21.4.1), which is invoked directly from the tool.

Segments: When this variation is selected, the Connect Lines tool can be used to connect open ends of individual segments, using either the postpick or the prepick selection method.

When using the postpick method, with the Connect Lines tool active, click on the two segments you wish to connect. With the postpick method the topological level is ignored and the system always picks at the Segment level. With the prepick method, first use the Pick tool to select any even number of segments, with the topological level set to Segment. Then select the Connect Lines tool and click anywhere in the project window to execute the operation. If you preselect an odd number of segments, then the last one will be ignored. The Segments operation is applied to pairs of segments, in the order in which they are selected. That is, the first and the second segments are connected, then the third and the fourth, etc., until the list of preselected segments is exhausted.

Figure 4.21.4.2: Connecting segments: (a) Crossing segments, (b) partially crossing, (c) non crossing segments, and (d) parallel segments.
Where you click to select the segments to be connected is significant, because the click points determine which ends of the segments will be connected. The direction of the resulting line is always determined by the direction of the first segment in the pair of segments that is connected. Examples of the Segments operation are shown in Figure 4.21.4.2.

**Lines:** When this variation is selected, the Connect Lines tool can be used to connect the open ends of two or more lines. This operation can only be applied to open lines and can be executed using either the postpick or the prepick method.

With the postpick method, select the Connect Lines tool and, with topological level set to Object, click on the two lines you wish to connect. A new segment is created, which connects the last point of the first line to the first point of the second line. Note that, contrary to the Connect Segments operations, the directions of the connected lines never change.

With the prepick method, first use the Pick tool to select any number of lines with topological level set to Object. Then select the Connect Lines tool and click anywhere in the project window to execute the operation. The operation is applied to the lines sequentially, in the order they were selected. That is, the first line is connected with the second, the second with the third, and so forth, until the list of preselected lines is exhausted.

**Close Line Sequence:** When this option is selected, the last point of the connected lines is also connected to the first point of the line, resulting in a closed shape. Examples of the Lines operation are shown in Figure 4.21.4.3.

The Connect Lines tool can also be used with splines and arcs, as well as composite lines. Its two variations are illustrated in Figure 4.21.4.4.
4.21.5 Joining lines

The open ends of two segments can be joined when they coincide or they are close enough. They are joined by merging the two end points into one.

Join Lines

This tool can be used to join open ends of lines. Variations of the operation are selected from the Join Lines Options dialog (Figure 4.21.5.1), which is invoked directly from the tool.

**Join**: When this option is selected, the Join Lines tool is used to join two or more lines, whenever their end points coincide or are within a distance less than the value entered in the Tolerance field of the Join Lines Options dialog. This operation can only be applied to open lines and can be executed using either the postpick or the prepick method.

With the postpick method, select the Join Lines tool and, with topological level set to Object, click on the two lines you wish to join. If either endpoint of the first line is close to either one of the two end points of the second line, the two lines will be joined at that point. If the joined points are not exactly on top of each other, the end point of the second line will be moved to the position of the end point of the first line. Note that the direction of the first line prevails and the direction of the second line may be reversed.

With the prepick method, first use the Pick tool to select any number of lines with topological level set to Object. Then select the Join Lines tool and click anywhere in the project window to execute the operation. The operation is applied to the lines sequentially, in the order they were selected. That is, the first line is joined with the second, the second with the third, and so forth, until the list of preselected lines is exhausted. Each pair of consecutive lines is only joined when the two lines have touching end points. The direction of the joined sequence of lines is determined by the direction of the first line.

**Close Line Sequence**: When this option is on and the last point of the sequence coincides (or is close enough) with the first point, then the line is closed. Examples are shown in Figure 4.21.5.2.
**Join All:** When this variation is selected, the Join Lines tool can be used to automatically join all the lines whose end points touch or are close. The operation can be applied in two different ways. With the Join Lines tool active and topological level set to Object, you can click on a line. The system will automatically search and find the sequence of lines that have touching end points, starting with the line selected. If the **Close Line Sequence** option is on, the sequence will be closed if, at any stage, a point is found that coincides with the first point of the first line. The direction of the sequence is determined by the direction of the selected line.

Alternatively, with the Join Lines tool selected, you can click anywhere in the project window, away from any selectable entity. This will cause the system to search for all possible sequences of lines and join them at the points where their end points touch. The resulting lines may be open or closed, depending on whether **Close Line Sequence** is on and a closing point is found.

This operation can only be applied to open lines and whether two end points touch (coincide) is determined by comparing their distance with the **Tolerance** value entered in the respective field of the **Join Lines Options** dialog. How far the system will search for lines with coincident ends depends on the status of the **Visible Layers Only** option. When selected, the system will only search among the objects on visible layers. Otherwise, it will search the complete project. Examples of the **Join All** operation are shown in Figure 4.21.5.3.

![Figure 4.21.5.3](image-url)
The second method of applying the **Join All** operation is intended to be used in cases where data imported from other applications comes in as individual segments that need to be connected. By clicking away from any pickable entity you instruct the system to make all the possible connections. However, frequently ends of more than two lines coincide, which makes the decision about what to connect an ill-defined case. The system will connect to the first matching line it encounters, which may not be the desirable solution. Consequently, some data management may need to be applied to the data at hand in order to derive the desired results. This is illustrated with the example to the right (Figure 4.21.5.4).

The data shown in Figure 4.21.5.4 consists mostly of individual segments, which are also in different directions. Assuming that what is needed is to connect these segments in the horizontal and vertical directions, two adjustments will be needed.

First, the horizontal and vertical lines should be placed on separate layers. Secondly the directions should be adjusted to become consistent. Both adjustments are shown in Figure 4.21.5.4 (b) and (c).

After the data adjustments have been made, the **Join All** operation is executed twice, each time keeping only one layer visible. The result is shown in Figure 4.21.5.4 (d) and (e).

The Join Lines operation can also be applied to open arcs, splines, and composite lines. However, for the ends of these types of lines to be joined they need to be exactly on top of each other. In other words, no tolerance can be used with arcs and splines.

**Figure 4.21.5.4:** Two different ways to join the shown lines.
4.21.6 Filleting and beveling

The points of 3D vector line shapes can be rounded or beveled, one at a time, or all of them with a single operation.

**Fit Fillet/Bevel Lines**

This tool can be used to either fit arcs or bevel points of shapes. Which operation is applied and its parameters are selected from the **Fit Fillet/Bevel Lines Options** dialog, shown in Figure 4.21.6.1, that is invoked directly from the tool.

**Fit Fillet**: When this operation is selected, the Fit Fillet/Bevel Lines tool can be used to round the points (vertices) of a line. These points can be rounded one at a time or more than one point may be rounded with a single execution of the operation. That is, the **Fit Fillet** operation can be applied at the Point level, which rounds a single point, at the Segment level, which rounds the two end points of the segment, at the Outline, Face, or Object level, which rounds all the points in the outline, face, or complete line. Only connected points (as opposed to end points) can be rounded. Examples of fillet fitting are shown in Figure 4.21.6.2.

The **Fit Fillet** operation can be executed using either the postpick or the prepick method. With the postpick method, select the Fit Fillet/Bevel Lines tool, select the desired topological level, and pick the entity you wish to round (point, segment, outline, face, or line). With the prepick method, first use the Pick tool to select any number of entities at different topological levels. Then with the Fit Fillet/Bevel Lines tool click anywhere in the project window. All the points of the selected entities are rounded. The rounding parameters and resolution of the fillet are determined by the values entered in the numeric fields of this group of options.

**# Of Edges**: The integer number entered in this field determines how many points will be used for the derivation of the rounding arc.
Modeling • Editing 3D lines, attributes, and controls

4 - 611

**Fillet Method:** The two options available in this group determine whether the value entered in the **Fillet Size** field will be interpreted as a radius (**Size Is Radius**), which is the default, or as a distance (**Size Is Distance**). The distance is taken from the corner point and, for each segment, it determines the point where the fillet arc will be tangent to the segment.

**Fillet Is: Facetted / Arc (Parametric):** When **Facetted** is on, the fillet generated at the angle between two segments is a polygonal approximation. When **Arc** is on, the fillet is a parametric arc. Only connected points (as opposed to end points) can be rounded.

**Bevel:** When this option is selected, the Fit Fillet/Bevel tool can be used to bevel the points (vertices) of a line. This operation is executed exactly as the **Fit Fillet** operation, except that the magnitude of the beveling is controlled by the values entered in its own fields.

**Offset:** When this option is selected, the bevel distance from the point can be entered independently for the **Left** and **Right** portions of the bevel point. Left and right are relative to the direction of the line that contains the point. Left is the segment that ends at the point and right is the segment that starts at the point.

**Length:** When this option is on, the size of the bevel is defined by the number entered in this field, which is centered to the bevel point. Examples of beveling are shown in Figure 4.21.6.3.

The Fit Fillet/Bevel Lines operation can also be applied to points where arcs and/or splines meet (and are connected) at some angle. Such examples are shown in Figure 4.21.6.4.
4.21.7 Inserting points and segments

Points and segments can be inserted to both solid and surface objects, including 2D open and closed shapes. They can be inserted to both polygonal and smooth objects. Note however that, when inserting segments to open lines, they can only be inserted between their end points.

**Insert Point**

Clicking with this tool on a segment inserts a new point at the location of the click. The result can be made visible by turning on the Show Points option in the Wire Frame Options dialog. Inserted points can be moved freely, using any of the geometric transformations, as illustrated in Figure 4.21.7.1.

**Insert Segment**

Clicking twice with this tool on points or segments of the same face of an object, inserts a new segment between the two click points (Figure 4.21.7.2). Note that, if the click points are on different faces, the insertion fails. This tool does not require the insertion face to be planar, which is a condition for other insertion operations (see section 4.4).

The insertion of a segment effectively splits a face into two faces. If the original face is planar, the new faces will be coplanar. However, as already mentioned, planarity is not a necessary condition. This makes the insert segment operation ideal for working with non planar faces.

This tool can be effectively used to decompose a face whose planarity may have been disturbed by other operations, such as the geometric transformations. A non planar face can thus be even completely “triangulated”, manually, rather than using the Triangulate tool (see subsection 4.10.5), which does not allow much control over the exact shape of the triangulation. On the other hand, manual triangulation allows complete control over the shape of the triangulation. For an example see Figure 4.21.7.3.

Examples of point and segment insertions on smooth objects are shown in Figure 4.21.7.4.
4.21.8 Adjusting topological attributes

Tools are available that can be used to change topological attributes or characteristics of objects, such as the position of their first point and their direction, or to mark some points, or to switch meshes from one-sided to two-sided, and vice versa. These tools are discussed below.

**Make First Point**

When you click with this tool on a point of a closed surface object, it makes that point the first point of the shape. The operation is illustrated in Figure 4.21.8.1. Note that, even though this operation affects a point, it is executed at the Object level and is always executed with the postpick selection method. Also note that this operation has no effect on open vector lines since their first points can not be changed.

First points are significant because they typically become the base points for c-meshes and nurbz, and can be justification points for sweeps. Whenever shapes are used as control lines for c-meshes or nurbz, it is important that their first points line up, which can be done by using this tool. First points can be made visible as little white diamonds, in wire frame display by turning on the **Show First Point** option in the **Wire Frame Options** dialog.

**Reverse Direction**

When clicking with this tool on an object, it reverses its direction. This can be any type of object, ranging from open/close vector lines, meshed surfaces, to solids. Reversing the direction of the latter turns them inside out. This tool is always applied at the Object level and can be applied using either the postpick or the prepick selection method. The directions of edge segments can be made visible as little arrows in their middle, in wire frame display by turning on the **Show Directions** option in the **Wire Frame Options** dialog (Figure 4.21.8.1).

This tool can be useful for adjusting the directions of control lines, to avoid twists, as shown in Figure 4.21.8.2. The directions of a surface mesh can also be adjusted after it is created.

---

*Modeling • Editing 3D lines, attributes, and controls*
**Make One/Two Sided**

This tool is used to execute two opposite operations. They are selected from the **Make One/Two Sided Options** dialog (Figure 4.21.8.3) that can be invoked from the tool. **Make Surface Solid** turns a one sided surface object into a two-sided surface. **Make Surface Object** does the opposite.

**Set/Clear Point Marker**

Clicking with this tool on an unmarked point marks it; clicking on a marked point it clears the mark. Clicking on a line away from a point clears all the marked points on the line, if any, after issuing a warning. Marked points can be displayed as diamonds (larger than that used for the first point), when the **Show Marked Points** option is turned on in the **Wire Frame Options** dialog.

Marked points are specifically useful for the Skin operation, when the source profiles of the skin are not already placed on the paths (see subsection 4.9.2).
4.21.9 Converting types of objects

**Convert**

This tool can be used to convert one type of object to another, provided the object is convertible. It can be applied using either the prepick or postpick selection method and it is always applied at the Object topological level.

When using the postpick method, with the Convert tool you click on one object. As soon as you do, the Convert Options dialog is invoked, as shown in Figure 4.21.9.1. With the prepick method, you preselect any number of objects and then, with the Convert tool active, you click anywhere in the project window. The Convert Options dialog is invoked once for each of the objects you preselected, in the order you selected them. The Convert Options dialog mainly consists of two boxes labeled Current and Convert To.

**Current:** This box contains two information fields labeled Object Type and Model Type. They display the respective types of the object you click on. For example, the dialog in Figure 4.21.9.1 was invoked after clicking on a smooth sphere primitive.

**Convert To:** This box contains the same two options (Object Type and Model Type), which are now pop up menus from which different options can be selected. These menus are shown in Figure 4.21.9.2(a) and (b), respectively.

When a parametric object is converted to a plain object, it no longer contains the parameters used to create it. A conversion may be necessary to allow operations that are otherwise not allowed on parametric objects. For example, it is not possible to apply an independent scale factor to a parametric object of revolution. This becomes possible after the conversion.

Not all objects are convertible. For example, a plain faceted object can not be converted to a parametric and/or smooth object, after its initial generation. When such an object is selected with the Convert tool, the pop up menus in the Convert To box are dimmed and simply display what the object is or, in other words, repeat the information displayed in the Current box.
**Additional Facetting**: This option becomes available when *Facetted* is selected from the *Model Type* pop up menu. It is dimmed otherwise. Selecting it offers the option to apply certain adjustments to the way the facets are generated. The choices are made in the *Additional Facetting Options* dialog (Figure 4.21.9.3) invoked when clicking on the *Options...* button next to this option. Note that a different dialog is invoked for primitives and all the other objects.

*Figure 4.21.9.3*: The *Additional Facetting Options* dialog for (a) primitive and (b) all other smooth objects.

**Triangulate**: When this option is on, the facets are also triangulated, as shown in Figure 4.21.9.8(c).

**Optimize**: When this option is on the structure of the facets is optimized by joining faces at areas of low curvature. This is shown in Figure 4.21.9.8(b).

**Grid Type**: This pop up menu determines how the grid, if any, will be applied to mesh a surface.

- **No Grid**: When this item is selected, no grid is generated on the surface. In other words, the entire surface becomes a single face. This option works best with planar surfaces, since these surfaces can be converted to form-Z faces without meshing.

- **Interior**: When this is selected, the grid is applied only to the interior of the surface and does not continue all the way to the boundaries of the surface. The space between the interior grid and the boundaries is triangulated. This is shown in Figure 4.21.9.4.

- **To Edges**: When this item is selected, the grid continues all the way to the edges of the surface, as shown in Figure 4.21.9.5. While the *Interior* item generates a more evenly meshed zone along the boundaries of a surface, this option works best on surfaces with boundaries that align with the grid direction.

*Figure 4.21.9.4*: Mesh with *Grid Type* set to *Interior*.

*Figure 4.21.9.5*: Mesh with *Grid Type* set to *To Edge*.
Triangulation: This pop up menu contains six items that determine whether and what type of triangulation will be applied to the imported entities.

None: When this item is selected, no more triangulation is applied and the mesh is generated according to the option selected from the Grid Type menu.

Everywhere: When this item is selected, each and all the grid cells are triangulated.

1, 2, 3, or 4 Levels From Edges: When one of these items is selected, triangles are generated in grid cells that are 1, 2, 3, or 4 levels deep from the boundaries of the surface. Meshed surfaces generated with each of the triangulation options are shown in Figure 4.21.9.6.

Figure 4.21.9.6: Triangulation variations: (a) None, (b) Everywhere, (c) 1 Level, (d) 2 Levels, (e) 3 Levels, and (f) 4 Levels From Edges.
**Facet Adjustment**: This pop up menu determines whether the points of the mesh are adjusted to create triangles or quadrangles with sides which are as evenly long as possible.

**None**: When this item is selected, no adjustment is made.

**Triangle Points**: When this is selected, adjustments are made only to points that are completely surrounded by triangles.

**All Points**: When this item is selected, adjustments are made to all mesh points. If a mesh is composed of triangles and quadrangles, the **Triangle Points** option guarantees that planar quadrangles remain planar, whereas the **All Points** option may disturb their planarity. A meshed surface before and after adjustment is shown in Figure 4.21.9.7.

**Apply To All**: When this option is on and the Convert tool is used to operate on a number of preselected objects, after the conversion choices have been made in the dialog for an object, they will also be applied to all objects in the preselected group to which can be applied. The program will invoke the **Convert Options** dialog again only for objects in the preselected group that are different.

*Figure 4.21.9.7: Facet Adjustment* set to (a) **None** and (b) **All Points**.

*Figure 4.21.9.8: Converting a smooth ellipsoid to faceted with **Additional Facetting** (a) off, (b) on and **Optimize** on, and (c) **Triangulate** on.*

Shown (1) in wire frame and (2) in hidden line.
4.21.10 Covering and uncovering wire objects

This tool can be used to cover closed wires. A wire is a surface object that is not planar. Covering it means generating a meshed surface that has the wire as its boundary. Note that closed wires can also be triangulated, using the Triangulate tool, which “repairs” the wire’s lack of planarity. However, triangulation is not specifically sensitive to the curvature of a wire, which the covering operation is.

From a practical point of view, both the facetted and the smooth modeling operations have difficulties with non planar surfaces. While facetted modeling is frequently able to tolerate them (even though it will not guarantee proper results), smooth modeling is not. Smooth modeling will accept curved and flat surfaces, but no closed shapes that do not have all their points on the same plane. Such shapes will not even render, except as wire frames. To be able to do anything with them, they need to be covered using this tool.

**Cover**

This tool is used to cover wires or, in other words, to transform non-planar closed shapes to surfaces. It can also be used to do the reverse, that is to transform into a wire a non planar meshed surface. This operation is called **uncover**. It can be applied using either the prepick or postpick selection method and it is always applied at the Object topological level.

When using the postpick method, with the Cover tool you click on one object at a time. With the prepick method, you preselect any number of objects and then, with the Cover tool active, you click anywhere in the project window. There is no dialog for this tool. An example is shown in Figure 4.21.10.1.

---

**Figure 4.21.10.1:** (a) A closed wire (non-planar shape) (b) is covered (a meshed surface is generated between its boundaries). (c) Once covered, it can be rendered.
4.21.11 Extracting control objects

The Extract tool can be used to retrieve the control lines of controlled or parametric objects. For example, you can retrieve (by making them new independent objects) the source and path objects of a parametric sweep or the control lines from which a c-mesh or a nurbz surface are generated. This is done by selecting the following option:

This tool can be used either in prepick or postpick mode. In postpick mode, with the tool active you click on the parametric object whose controls you wish to extract. In prepick mode, you use the pick tool to preselect any number of parametric objects. Then, with the Extract tool you click anywhere in the window. The operation is executed for each of the objects you preselected.

The options that affect the execution of this tool are selected from the Extract Options dialog, shown in Figure 4.21.11.1, which is invoked directly from the tool.

**Group Multiple Objects**: When this option is on and there are two or more control objects, they are placed in a new group.

**Create Objects In Length (U) Direction, Create Objects In Depth (V) Direction**: When one of these mutually exclusive options is on, the control lines in the respective direction (U or V) are extracted and turned into objects.

**Create Objects As C-Curves**: When this option is selected, the length control lines of a nurbz or c-mesh object are converted into a c-curve with the same smoothing degree as the nurbz or c-mesh object. If this option is off, the control lines are converted to simple vector lines.

Note that the objects constructed from the control lines inherit the layer and surface style of the controlled object from which they are derived. Examples are shown in Figures 4.21.11.2 and 4.21.11.3.

![Figure 4.21.11.1](#)

*Figure 4.21.11.1*: The Extract Options dialog.

*Figure 4.21.11.2*: (a) A source shape is drawn and (b) revolved. (c) The object is edited and its form is changed, then (d) its source shape is extracted.
Figure 4.21.11.3: (a) Three control lines from which a (b) c-mesh is generated. (c) The c-mesh is edited and is changed, which also changes the shape of its control lines. The control lines with their new shape are extracted (d) as plain lines and (e) as c-curves.
4.22 Inquiring about objects, their attributes, and quantities

The following seven tools are discussed in this section:

- Query
- Query Attributes
- Query Parameters
- Measure
- Object Doctor
- Project Doctor
- Print Prep

After objects have been created, have been further manipulated by applying any variety of operations to them, and have been assigned attributes, you can request information about their status, structure, attributes, and parameters using the top three tools above. You can request information about complete objects or their parts (points, segments, outlines, and faces). You can also use the Query tool to change the parameters of parametric objects. The fourth tool is used for measuring lengths and distances.

The last three tools offer diagnostics and the ability to repair certain irregularities at either the object or the project level.
4.22.1 Querying objects

After objects have been created, have been manipulated, and have been assigned attributes, you can request information about their status, structure, and attributes using the Query tool. You can query complete objects or their parts (points, segments, outlines, and faces). You can also use this tool to change the parameters of parametric objects.

Query

The Query tool can be applied using either the postpick or the prepick method. To use the postpick method, activate the Query tool, set the topological level where appropriate, and then select the entity for which you desire information. The program presents different dialogs for the different topological levels. For example, when objects are selected, the Query Object dialog in Figure 4.22.1.2 is shown. This dialog is also accessible from all the other Query dialogs. For example, after information is issued about a point, information about the object to which the point belongs can also be requested.

To use the prepick method, preselect any number of entities, which may be at different topological levels, activate the Query tool, and then click anywhere in the graphics window. When prepicking more than one entity, the program parades through the entities selected, one at a time, and presents information about each one in the order in which they were picked.

Querying groups

Clicking on an object with the Query tool, while topological level is at Group, invokes the Query Group dialog (Figure 4.22.1.1), which contains information about the name, type, and members of the group.

When you click OK to close the Query Group dialog, the program parades through all the objects in the group, and offers information about each one individually. Each click on the OK button takes you to the next object in the group. Clicking on the Cancel button terminates the parade, and exits from the Query environment.

Figure 4.22.1.1: The Query Group dialog.
Querying objects

Clicking on an object with the Query tool invokes the **Query Object** dialog (Figure 4.22.1.2) that offers object level information.

**Object Type**: The object type is displayed in this field. If it is a parametric object, the **Edit** button next to it is active. Clicking on this button opens a preview dialog, which allows you to edit and change the parameters of the object. Depending on the type of the parametric object, a different preview dialog is opened. The **Edit** button is dimmed and inactive if the object is of the faceted type. The editing of parametric objects is discussed in the next subsection.

**Model Type**: The model type (**Facetted** or **Smooth**) is displayed in this field.

**Is Clone**: The **Yes** or **No** that is displayed in this field indicates whether the object is part of a clone family or not. Cloning is discussed in section 4.23.5.

**Display Resolution**: The two fields in this group contain information about the display of the object, that is, how many **Facets** and how many **Points**.

The remainder of the **Query Object** dialog is divided in two parts: one contains information about the topology and the other about the geometry of the object.

**Topology**: These fields contain information about the topology of an object.

- **Type**: This field displays the object type relative to its topological structure. It can be a point object, an open or closed surface object, a meshed surface, a surface solid, or solid.

- **Well Formed**: This field displays a “yes” or a “no,” depending on the topological completeness of the object in question. Recall that the term “well formed” is equivalent to “solid.”

- **Faces, Edges, Points**: These fields display how many of each of these entities are contained in the object. Note that reversely coincident (double) segments are only counted once.

**Geometry**: This group of fields displays geometric information about the object.

- **Non Planar Faces**: This is a count of how many non planar faces are in the object. If they are all planar, it displays 0. Next to this field is the **Strict Planarity** option. When on, the calculation of non planar faces is more sensitive to smaller planarity disturbances. Default is off.
**Surface Area:** The area of the surface of the complete object is displayed in this field, if it can be calculated. For example, an open vector line has no area. The units of measurement can be selected from the pop up menu next to this field, which is shown in Figure 4.22.1.3.

**Volume:** The volume of the complete object, when it is a solid, is displayed in this field. The units of measurement are selected from the pop up menu next to it, which is shown in Figure 4.22.1.4.

**Center Of Gravity:** These fields show the X, Y, and Z coordinates of the center of gravity of the object. For solids, the center of gravity is the point around which an object can pivot freely when suspended in space, assuming that the object is made from an equally dense material. For different types of objects the center of gravity is calculated as follows:

- Single point object: it is the single coordinate of the object.
- Open surface object (vector line): it is the weighted average of all mid-segment points.
- Closed surface object: it uses the center of gravity of the single face.
- Surface solid and meshed surface: it is the weighted average of all face centers of gravity.
- Solid: it is the volumetric center of gravity.

**Object Transformation:** These groups of fields display information about the position of an object’s local origin and the orientation of its axes.

**Origin:** These fields show the X, Y, and Z coordinates of the object’s local origin.

**Rotation:** These fields show the X, Y, and Z orientation of the object's local axes.

**Edit Object Axes...**: This button invokes the **Object Axes** dialog, shown in Figure 4.22.1.5, which contains options for editing the location of the origin.
Mass Properties...: Clicking on this button invokes the Mass Properties dialog, shown in Figure 4.22.1.7. It contains groups of options that display information about the object’s Axes, its Weight And Volume, and its Moments Of Inertia.

The units of measurement for weight and volume are selected from the pop up menu next to the respective fields. The Volume menu is shown in Figure 4.22.1.4 and the Weight menu in Figure 4.22.1.8.

Object Centroid: When clicking on an object with the Query tool, this pop up menu displays the type of centroid currently set for the object. This can be changed by selecting another item from the pop up menu (Figure 4.22.1.5). The available options are the same as those available for the Reset To menu (see above). If the object has no “natural” origin, then Per Object Type is dimmed.

Editing the object axes

The object axes can be edited throughout the Object Axes dialog, shown in Figure 4.22.1.5. This dialog shows the object with its axes and keypoints in a standard preview window and also lists the axes origin and rotation numerically. Through this dialog, the axes can be transformed relative to the object. For example, the user may want to align the object’s axes with a particular face, or set it back to the center of gravity. This can be achieved by either typing in specific values of the axes origin and rotation or by graphically moving and rotating the axes in the preview window.
**Origin, Rotation:** These fields show the values for the axes origin and rotation, which can be changed numerically.

**Reset To:** This pop up menu offers options for resetting the location of the origin to one of four choices: **Per Object Type**, **Center Of Gravity**, **Average Of Points**, and **Center Of Bounding Volume**. These options are the same as in the **Objects** dialog, that can be invoked from the Objects palette and are discussed in more detail in section 2.8.2.

The object axes can also be changed graphically. When moving the cursor over the object axes in the preview window of the dialog, it changes from the arrow to a crosshair. Clicking at that point invokes the object axes edit mode (Figure 4.22.1.6). This is the same editing mode that is also used by the Transform Object Axes tool. Clicking in one of the controls activates either a rotation or a translation of the object axes. Clicking a second time finishes it. While the axes are rotated or moved, the cursor is automatically snapped to points, midpoints, keypoints and segments of the object. The object axes edit mode is described in more detail with the Transform Object Axes tool (section 4.3.7) and Transform tool (section 4.23.2).
Querying points

When using the Query tool at the Point level, the program responds with the **Query Point** dialog shown in Figure 4.22.1.9. It provides information about the world coordinate position of the point selected. The X, Y, Z coordinate fields are also editable, and the values they contain can be changed. In other words, the **Query Point** dialog can also be used to reposition a point. The position will be changed as soon as you click on the OK button and exit the dialog.

Clicking on **Query Object**... invokes the **Query Object** dialog, which displays information about the object the selected point is on. Such a button is available in all Query dialogs for entities at levels other than object and group.

Querying segments

At the Segment level, the program invokes the **Query Segment** dialog (Figure 4.22.1.10), which provides information about the world coordinate positions of the two **Endpoints** of the segment, as well as its **Length**. The coordinate fields are editable and can be changed, as for individual points. The **Query Object**... button is also available for accessing information about the complete object.

Querying outlines

At the Outline level the program presents the **Query Outline** dialog, shown in Figure 4.22.1.11. It provides information about the **Outline Type** (closed or open), the **# Of Edges**, whether the outline is **Planar**, the length of its **Circumference**, and its **Surface Area**. The **Query Object**... button is also available for accessing information about the complete object.
Querying faces

The **Query Face** dialog (Figure 4.22.1.12), which is invoked when the Query tool is executed at the Face level, offers all the information found in the dialog for the outline, as well as information about the **Holes** that the face may contain, the **Area Of Holes**, and the **Center Of Gravity** of the face.

As with the other Query dialogs, the **Query Object...** button is available for invoking the respective dialog with information about the complete object.

*Figure 4.22.1.12: The Query Face dialog.*
4.22.2 Using Query to change parametric objects

When the object selected for the Query tool is a parametric object, in addition to retrieving information about it, the Query environment can also be used to change it. When the type displayed in the Object Type field is parametric, the Edit... button next to it is active. Clicking on it invokes a dialog for the respective object type, where the parameters of the object being queried are displayed and can be changed.

Querying arcs, circles, ellipses, splines, and composite curves

When an arc, circle, or ellipse is selected with the Query tool, the Query Object dialog displays Arc, Circle, or Ellipse respectively for type. Clicking on Edit... invokes the Arc/Circle/Ellipse Edit dialog (Figure 4.22.2.1). It allows you to both inspect the current parameters of the shape and to change them. Clicking on a spline and then the Edit... button invokes the Spline Curve Edit dialog. Clicking on a composite curve and then the Edit... button invokes the Composite Curve Edit dialog.

Querying c-curves

When a c-curve is selected, the Query Object dialog displays Controlled Curve for type and clicking on Edit... invokes the Query: C-Curve Options dialog (Figure 4.22.2.2) which is a variation of the C-Curve Options dialog invoked from the C-Curve tool. It shows all the parameters used for the generation of the c-curve. Changing any parameter causes the c-curve to be regenerated as soon as you click on the OK.
Querying c-meshes

When a c-mesh is selected, the **Query Object** dialog displays a **Controlled Mesh** for type and clicking on the **Edit...** button invokes the **Query: C-Mesh Options** dialog (Figure 4.22.2.3). It contains four tabs: **C-Mesh Options**, **Object Type**, **Length (U) Smooth**, and **Depth (V) Smooth**. They all contain options similar to those in the **C-Mesh Options** dialog invoked from the C-Mesh tool.

The **C-Mesh Options** tab contains the parameters that set the resolution of the net (Figure 4.22.2.3).

The **Object Type** tab, shown in Figure 4.22.2.4, contains the options that determine the type of the c-mesh to be generated.
The Length (U) Smooth and the Depth (V) Smooth tabs contain the options that set the type of curves to be generated for the length and depth of the c-mesh, respectively. These tabs are shown in Figure 4.22.2.5.

Recall that the curve parameters used for the length of the control net of a c-mesh may be set independently for each control line. In contrast, only one set of curve parameters can be used for the depth of a c-mesh. Consequently, the tab for the length contains two buttons, labeled Previous and Next, which allow you to parade through all the length control lines and change their parameters, if desired. These buttons are not available in the tab for the depth of the c-mesh. In a way, the Query environment allows you more control over the manipulation of the curve parameters than the C-Mesh tool, where the curve parameters can not be changed individually.

As with the c-curves, when any of the parameters of the c-mesh are changed in one of the tabs, the Query environment behaves as if the Adjust To New Parameters options were selected, and always regenerates the c-mesh.

![Figure 4.22.2.5: (a) The Length (U) Smooth and (b) the Depth (V) Smooth tabs of the C-Mesh Options dialog.](image-url)
Querying primitives

When a parametric primitive that has editable parameters is selected with the Query tool, the Query Object dialog that is invoked displays the type of the object and an active Edit... button. Clicking on the latter invokes the respective Edit dialog, where the objects parameters can be changed. A few examples are shown here.

When an analytic primitive is selected, the Query Object dialog displays Cube, Cone, Cylinder, Sphere, or Torus for type and clicking on Edit... invokes the respective Edit dialog. For example, the Cone Edit dialog is shown in Figure 4.22.2.6. This dialog displays all the parameters used for the generation of the analytic cone. Changing any parameter causes the cone to be regenerated.

When a parametric derivative, such as a revolved object, helix, screw/bolt, spiral stair, stair from path, or sweep is selected, the Query Object dialog displays the type name and clicking on the Edit... button invokes the edit dialog for the selected type of object. For example, Figure 4.22.2.7 shows the Revolve Edit dialog. Again, changing any parameter displayed causes the revolved object to be regenerated.
Similarly for all the other parametric objects, namely smooth mesh objects, displacement objects, rounded objects, metaformz, nurbz, spherical objects, and symbols. When selected, the **Query Object** dialog displays their type and clicking on their **Edit...** button invokes the respective edit dialog. The **Q-Subz Edit** and **Nurbz Edit** dialogs are shown in Figures 4.22.2.8 and 4.22.2.9.

![Figure 4.22.2.8: The Q-Subz Edit dialog.](image)

![Figure 4.22.2.9: The Nurbz Edit dialog.](image)

**Modeling** • Inquiring about objects, their attributes, and quantities
### 4.22.3 Inquiring about attributes and parameters of objects

In addition to retrieving information about the topological and geometrical characteristics of an object, you can also find out about its attributes, which is done with the following tool.

#### Query Attributes

This tool offers information about an object’s attributes, displayed in the **Query Object Attributes** dialog (Figure 4.22.3.1), or about a face’s attributes, displayed in the **Query Face Attributes** dialog (Figure 4.22.3.2). Which is invoked by the tool depends on the active topological level.

The information displayed consists of the object **name**, **layer**, **surface style**, **visibility**, **pickability**, and **snapability** attributes, **shadow casting** and **receiving**, wire frame and shaded **rendering** options.

The dialog also includes, in the lower portion, a scrollable list titled **Additional Attributes**. This list shows all the complex attributes offered by form•Z and the attributes installed by plugins.

The list has two columns. The left column shows the **Name** of the attribute. The right shows the **Status**. When an attribute is not used by the object or face, the status shows **Unused** or the attribute is not even shown in the list. Double clicking on an item with status **Unused** will add the attribute and invoke the respective edit dialog for the attribute, where the default settings can be changed. To remove an existing attribute, select the item in the list and click the **Remove** button, found under the list.

**Query Object...**: Clicking on this button invokes the **Query Object** dialog, where additional information about an object is presented (see subsection 4.22.1).
There are three different types of attributes that reflect the method by which they are added to an object or face. They are as follows:

- **Added Always**: This type of attribute is automatically added to every object when it is first created. When querying a newly constructed object, the attribute always shows up in the list with a status of **In Use**. form·Z does not offer an attribute of this type, but plugins do.

- **Added Through Query Dialog**: This type of attribute can be added through the **Query Object Attributes** or **Query Face Attributes** dialogs. It is not automatically attached to an object at the time it is created. When querying an object or face, this attribute will initially show as **Unused**. As already mentioned, double clicking on the attribute item in the list adds it to the object or face. The **Smooth Shading** attribute is an example of this type.

- **Added Through Other Method**: This type of attribute cannot be added through the **Query Object Attributes** or **Query Face Attributes** dialog. These dialogs will not show the attribute in their lists, unless it has been added through other means, such as by a dedicated attribute tool. The **Texture Map** attribute is an example for this type.

In addition to retrieving information about the attributes and the topological and geometrical characteristics of an object, you can also find out about its parameters, which is done with the following tool:

*Query Parameters*

This tool offers information about an object's attributes. Clicking on an object with this tool invokes an **Edit** dialog, displaying the parameters of the respective object type. This tool functions the same as selecting the **Edit**... button from the **Query** dialog.
4.22.4 Measuring lengths and distances

It is frequently necessary to know some lengths of entities as well as distances between entities, which is offered by the following tool.

This tool can be used to measure lengths and distances of open or closed surface objects or faces of solids, as well as measuring distances between points, segments, and surfaces. Which measurement is made is selected from the Measure Options dialog (Figure 4.22.4.1), which is invoked directly from this tool.

Lengths of shapes

Length Of Vector/Bounding Line: When this option is selected, the Measure tool can be used to measure the length of an open or closed shape, a face, an outline, or a segment.

With topological level set to Object and the Measure tool active, click on a surface object. The message dialog shown in Figure 4.22.4.2 is invoked and shows you the length of the boundary of the shape, which can be open or closed. If you set the topological level to Face, Outline, or Segment and you select an entity at the appropriate level, a message dialog posts the length of the entity you selected. When the topological level is at a level lower than Object, the respective entity can be on any type of object including solids. Note that executing this operation at the Point level is meaningless.
Distances between two points measured interactively

**Interactive Point To Point**: When this option is selected, the Measure tool can be used to measure distances between two points. These points can be anywhere on the reference plane or on an element. Any of the object snaps can be used to accurately place the mouse cursor at the point on an object whose distance you wish to measure.

After selecting this option, with the Measure tool active click on the first point. As soon as you do a line is rubber banded. It is anchored at your first point and follows the motion of the mouse. Simultaneously, a value is displayed in the Prompt palette, after the prompt “Distance:” (see Figure 4.22.4.3).

![Figure 4.22.4.3: Distance between two points reported in the Prompts palette.](image)

As you move the mouse, this value is continuously updated and gives you the length between your first point and the current position of the mouse. As soon as you click again, the distance is permanently recorded in the Prompts palette and no other message is issued. Note that this option ignores the topological level and always measures distances between points.

Distances between points and segments

**Point/Segment To Point/Segment**: When this option is selected, the Measure tool can be used for measuring the distance between two points, two segments, or a point and a segment. Note that the distance between two segments is only defined when the segments are parallel. If two non parallel segments are selected, an error message is issued.

With the Measure tool active, click on a point or a segment and then on another point or segment. Depending on what combination of points or segments you click on, one of the message dialogs in Figure 4.22.4.4 will inform you about the distance. Note that this operation requires no specific topological level to be set. It is able to recognize whether you click on a point or a segment and to calculate the distance. However, contrary to the **Interactive Point To Point** operation, the point you select for this option has to be on an object. It cannot be on the reference plane, away from an object.

![Figure 4.22.4.4: Distances between points and segments.](image)
Distances between points and surfaces

Point To Surface: When this option is selected, the Measure tool can be used to measure the distance between a point and a surface.

With the Measure tool active, click on a point and then select a face. How the face is selected (with one or two mouse clicks) depends on the option active in the Pick Options dialog. As soon as a surface is selected the message dialog shown in Figure 4.22.4.5 is posted with the distance. This operation requires no specific topological level to be set. It knows to select a point first and a face second.

Distances between points and the reference plane

Point To Reference Plane: When this option is selected, the Measure tool can be used to measure the distance of a point from the reference plane.

With the Measure tool active, click on a point. The message dialog shown in Figure 4.22.4.6 is posted with the distance. Note that this operation requires only one click.

Distances between surfaces

Surface To Parallel Surface: When this option is selected, the Measure tool can be used to measure the distance between two parallel surfaces.

With the Measure tool active, select two faces. How the faces are selected depends on the option active in the Pick Options dialog. As soon as the second surface is picked the message shown in Figure 4.22.4.7 is posted with the distance. Note that if the two picked surfaces are not parallel, an error message is issued.
4.22.5 Diagnosing and correcting problem conditions in an object

It is possible that certain irregularities may exist in an object that need to be detected and/or corrected. Such irregularities may have been accidentally caused by a user when executing other modeling operations, or they may have been generated when importing data from other programs, where the data may not be as clean as expected by form•Z. For example, one irregular condition would be the existence of zero length segments. Although regular modeling operations do not create such segments, given form•Z's ability to apply geometric transformations at any topological level, a user may have accidentally or intentionally moved the start point of a segment on top of the end point of the same segment. Such a condition will cause a number of modeling operations, such as the Booleans, to fail. Given that it is difficult to detect such segments and other irregularities by mere visual inspection, form•Z offers a tool that can be used to either simply detect and highlight the irregularities or to correct them, whenever possible.

**Object Doctor**

This tool can be used to either detect or correct certain types of irregularities that may exist in an object. It can be applied only at the object level using either the postpick or prepick selection method. When using postpick, one object is selected at a time and the program checks if the conditions that are turned on exist on that object. With the prepick method, several objects can be diagnosed and/or fixed at the same time.

The tool operates in three different modes: (1) To preview the problem conditions in a separate dialog. (2) To detect and highlight entities that present problem conditions in the project window. (3) To correct problem conditions that may exist in an object, whenever possible. These modes of operation are selected from the **Object Doctor Options** dialog that is invoked directly from the tool. They are described in more detail later in this section.

In the **Object Doctor Options** dialog, the irregular conditions which the Object Doctor tool can detect are organized in three tabs. The **General** tab contains the conditions that apply equally to both smooth and faceted objects. The **Facetted** and **Smooth** tabs list only the conditions that apply to faceted and smooth objects, respectively. Each condition is displayed with a check box. When that box is selected, the Object Doctor tool will test for and/or correct the respective condition. If the condition is off, it is ignored during the test. There is also a way to turn all the conditions of a tab either on or off. At the end of each tab there are two buttons: **All** and **None**. When selected they turn all the check boxes in that tab on or off, respectively.

Each condition in the tabs represents a certain irregularity that may occur in an object. Some of these irregularities can be corrected automatically by form•Z. When they can, the icon of a small hammer is shown next to the condition title. The irregularities that are not marked with a hammer may be corrected by the user, after they are detected by the program.
The General tab

This tab, shown in Figure 4.22.5.1, contains three conditions that apply to both facetted and smooth objects. The first two of the conditions can be corrected by form•Z.

Zero Length Segments: This condition tests for segments that do not have any length (have 0 length). In other words, the start and end point of the segment are on top of each other. form•Z corrects this problem by removing the offending segments.

Zero Area Faces/Holes: This condition tests for faces and holes that have no area. For example, a face may have collapsed to a line, because some of its points were moved manually, as shown in Figure 4.22.5.2. form•Z fixes this problem by removing the offending faces or holes.

Intersecting Faces: This condition tests for faces of an object that intersect each other, as shown in Figure 4.22.5.3. Certain operations, such as the Booleans, may refuse such objects. This condition cannot be corrected automatically by form•Z, but it is frequently correctable by a user.

Missing Faces: This condition tests for missing faces of an object. A missing face can be seen as an area in an object with an open boundary, which, if covered by a face, would close a hole surrounded by several faces. An example of such an object would be a cube, whose top face was deleted, as shown in Figure 4.22.5.4. While such a cube is easy to diagnose, other objects, with a dense mesh of faces, are not. form•Z fixes this condition by automatically creating the missing faces. In order to distinguish those areas that define a hole from those that do not, two options are available:
**Min # Of Edges** and **Max # Of Edges**: The values entered in these text fields define a range of edges such a hole should have in order to be covered with a face. An example is shown in Figure 4.22.5.5. The original object is a surface with a mesh of faces. In its middle there is a hole. At the same time, the outer boundary of the mesh may also be seen as a hole. Unless we wish to turn the whole object into a solid, the objective in this case would be to only cover the boundary along the inside of the mesh and not along the outside. As the number of edges along the outside of a mesh is usually larger than along the inside, specifying a smaller number for the **Max # Of Edges** will select only the inside hole.

The **Facetted tab**

This tab contains six conditions (Figure 4.22.5.6) that apply only to faceted objects. All, except the last condition, can be corrected by form-Z.

**Duplicate Points**: This condition tests for geometrically duplicate points and can be fixed. Importing from a number of other 3D programs may produce objects with such a condition. For example, another application may be representing a cube with six separate faces, each of which has four independent points, for a total of 24 points (6 times 4 points). Such faces form a cube by simply touching each other. However, such a collection of faces can not be treated as a solid object, although it looks like it is. In a properly formed solid cube, only 8 points are used, as shown in Figure 4.22.5.7. Correcting this object reduces its number of points from 24 to 8 and turns it into a proper solid. Another way to create such an object would be to construct six rectangular surface objects, arrange them to form a cube like shape, and join them together into a single object using the Join Volumes tool.

![Figure 4.22.5.5: A meshed object with an inside and outside open boundary.](image)

![Figure 4.22.5.6: The Facetted tab of the Object Doctor Options dialog.](image)

![Figure 4.22.5.7: A cube with and without duplicate points, shown in a slightly exploded view.](image)
**Collinear Segments**: This condition tests for segments of an object that are along the same line and in a sequence. form•Z corrects this condition by joining all collinear segments into one segment, as shown in Figure 4.22.5.8.

**Coplanar Faces**: This condition tests for faces of an object that are on the same plane and neighbor each other. For example, subdividing a face of an object using the Mesh tool creates a number of coplanar faces from a single face. When fixing this condition, form•Z reduces all the coplanar faces into a single face. This is in effect the inverse operation of the Mesh tool.

**Non Planar Faces**: This condition tests for faces of an object that are not planar. form•Z corrects this condition by triangulating the non planar surface. The triangulation options can be accessed by pressing the **Triangulate Options...** button found below this condition. This brings up the same dialog as the one invoked from the Triangulate tool.

**Irregular Face Boundaries**: This option tests for a number of problematic conditions of the boundary of a face. form•Z is able to detect such conditions, but it cannot automatically fix them.

- The first condition checked is edges of a face that intersect each other. An example of this condition is shown in Figure 4.22.5.9.

- The second condition is where the hole of a face is no longer contained inside the outer face boundary. Such a condition can be created, for example, by manually moving all points of a hole outside the face, as shown in Figure 4.22.5.10.

- The third condition checked, is where a hole of a face is located inside another hole of the same face, as shown in Figure 4.22.5.11. Again such a condition can be created by manually moving the points of a hole.
The Smooth tab

This tab contains seven conditions that apply to smooth objects only. It is shown in Figure 4.22.5.12. None of these conditions can be corrected by form•Z.

**Duplicate Points**: This condition checks for duplicate points in a smooth object. Two points are considered duplicate if the distance between them is less than 0.000001.

**Self-Intersecting Faces**: This condition tests for faces of a smooth object that intersect themselves. Such faces could cause problems in certain smooth operations.

**Self-Intersecting Edges**: This condition tests for edges of a smooth object that intersect themselves. Such edges could cause problems in certain smooth operations.

**Irregular Faces**: This condition tests for faces of a smooth object that are irregular. A face is irregular if its underlying surface geometry is scrunched up or twisted. Subdivision of such surfaces fails. Irregular faces could cause problems in certain smooth operations.

**Irregular Edges**: This condition tests for edges of a smooth object that are irregular. An edge is irregular if its underlying curve geometry is scrunched up or twisted. Subdivision of such curves fails. Irregular edges could cause problems in certain smooth operations.

**Discontinuous Spline Faces**: This condition tests for faces of a smooth object that have G0/G1/G2/C1 discontinuities. Such faces could cause certain smooth operations, like parallel and blend, to fail.

**Discontinuous Spline Edges**: This condition tests for edges of a smooth object that have G0/G1/G2/C1 discontinuities. Such edges could cause certain smooth operations, like parallel and blend, to fail.
Actions to be taken

At the lower portion of the **Object Doctor Options** dialog there is the **Action** group of options, as shown in Figure 4.22.5.13. These options determine what action will be taken when the Object Doctor tool is applied.

**Show Preview**: When this option is selected, the picked objects are shown in the **Object Doctor Edit** dialog, one at a time. As shown in Figure 4.22.5.14, this dialog has a standard 3D preview window, where the entities found by the diagnostics can be seen and where the fixes for the offending conditions can be viewed. The edit dialog is very similar to the **Object Doctor Options** dialog, except that the two remaining actions, **Add To Selection** and **Fix If Possible**, appear as buttons.
Add To Selection: This option is available as a radio button in the Object Doctor Options dialog and as a button in the Object Doctor Edit dialog. When selected in the former, the irregularities detected on the basis of the selected conditions are displayed highlighted in the project window. Clicking the button in the Edit dialog does the same but highlights the offending entities in the preview window.

Fix If Possible: This option too is available as a radio button in the Object Doctor Options dialog and as a button in the Object Doctor Edit dialog. When it is on in the former, the irregularities detected on the basis of the selected conditions are corrected, whenever form-Z can fix them. Similarly, clicking on the button in the Edit dialog fixes the selected conditions, if possible. In both cases the displays of the objects in the window are updated after the corrections are made.

Display Results: This option, found in the Object Doctor Options dialog, makes it possible to receive information about how many offending entities from each criterium were detected. When on, as soon as the Object Doctor tool is applied to one or more objects, it invokes the Object Doctor Results dialog, shown in Figure in Figure 4.22.5.15. This dialog contains the same three tabs the Object Doctor Options does, and each tab contains the same conditions. Next to each condition there is an alpha field where the number of irregularities detected is displayed. If none is found from a certain condition, 0 is shown.

Figure 4.22.5.15:
The Object Doctor Results dialog open at the (a) General and (b) Facetted tabs.
4.22.6 Diagnosing and correcting problem conditions in a project

The project doctor tool can be used to detect problems that occur between different objects in a project. Currently, there is only one such condition that is detected, which is whether two or more objects are duplicate.

![Project Doctor](image)

The Project Doctor tool can be executed only on the object level and works best in prepick mode. That is, several objects are first selected with the pick tool or all objects are selected with the Select All Unghosted menu command. Second, the Project Doctor tool is selected and the mouse is clicked anywhere in the project window.

The options that affect the Project Doctor tool are selected in the Project Doctor Options dialog, which is invoked directly from the tool.

**Duplicate Objects**: When this option is selected, all picked objects are checked whether they are identical in shape and are placed on top of each other. Such a condition can be generated by mistake quite easily, if, for example, an object is copied multiple times without moving it to a different place. Duplicate objects often cause unexpected results in renderings. After the mouse is clicked in the project window, the first object of a set of duplicate objects is deselected, while the remaining objects stay selected. Any preselected object which is not duplicate is also deselected. To remove all duplicate objects, one can now select the Delete tool and click in the project window. Note that form•Z determines that two objects are duplicate purely on a geometry basis. If the objects have different attributes, they are still considered duplicate.

**Add To Selection**: When this option is selected, the duplicate objects remain selected, while all other objects are deselected, as described above.

**Fix If Possible**: When this option is selected, the duplicate objects are automatically deleted.
4.22.7 Identifying potential problems before exporting to 3D printing

It is possible that certain objects may exist that will be difficult to print in 3D. Certainly point and line entities should not be exported to 3D printing. Very small objects will also be difficult to print, as they will have a tendency to break easily. While these objects are well formed and offer no problems in form•Z, the application offers a tool that can be used to detect and highlight these potential 3D printing problems. In other words, this tool allows the user to prepare the project for export to 3D printing.

Print Prep

This tool simply detects and highlights potential 3D printing problems. No attempt is made to correct these problems. In fact, the detected objects may or may not need to be corrected. In most cases there are several methods to correct a problem. For example, if an object is detected as too small for printing, the remedy may be to scale the object to be slightly larger. Another remedy may be to simply move the object to another layer so it will not be exported. Still another remedy may be to leave the object the same and simply adjust the scale factor with which the project is exported to the 3D printing file. In any case, there are many tools in form•Z that can be used to correct potential 3D printing problems.

The Print Prep tool applies to the entire project. Therefore prepicked objects will not affect this tool. Upon invoking the tool, the Print Prep Options dialog box, shown in Figure 4.22.7.1, will appear allowing the user to select one or more conditions. Once the condition(s) is selected simply click in the project window to conduct the test.

The results of this tool can be viewed in two different ways. (1) To highlight those objects which do not pass the selected condition(s). (2) To display a dialog where the number of detected objects for each condition is displayed. These methods are described in more detail later in this section.

In some of the conditions, a minimum and/or maximum distance are required. These distances are displayed in both world and print scale. The world scale represents a distance in the world; a wall is 6” thick, for example. The print scale represents the same distance in the scale of the printed model. For example, if the model were to be scaled by 12:1 where 1’ in the world equals 1” in the print model, then the same 6” wall would have a print scale of ½”. Therefore, modifying the world scale distance will also change the print scale distance. Modifying the print scale distance will also change the world scale distance.

The factor used to convert between world and print scales is determined by the scale parameter settings specified at the top of the Print Prep Options dialog. These same options can also be found in the Modeling Export Options: ZPR dialog.
Modeling

Inquiring about objects, their attributes, and quantities

The results of each test are determined by the selected conditions. One or more of the six possible conditions must be selected by checking the appropriate box. Each condition represents a potential 3D printing problem. When that check box is selected, the Print Prep tool will test for the respective condition. If the condition is off, it is ignored during the test. There is also a way to turn all conditions either on or off. Near the bottom of the options dialog box there are two buttons: All and None. When selected they turn all the conditions of the test on or off, respectively.

**Non Solids:** This condition checks each object to determine if it is a solid. Objects that are not solids will be counted. The “naked” or “untwined” segments of that solid will be highlighted.

**Inside Out:** This condition determines if the surfaces of a solid are inside out. All solid objects that need to be reversed will be highlighted.

**Printer Model Selection:** Use the drop down menu to select a printer model. This information is used to determine the size of the printer bed and may affect the scale conversion factor.

**Scale:** Models printed on Z Corporation printers are limited to a certain size, which may make it necessary to scale the model when it is exported. The Scale option will enable the scaling parameters.

**Scale Factor:** When this option is selected, the scale factor contained in its field will be uniformly applied in all directions to the model.

**Fit Box:** When this option is selected, the model is scaled to fit in a bounding box whose size is determined by the numeric values contained in the X, Y, and Z fields.

**Scale To Printer:** When this option is selected, the model is uniformly scaled to fit within the physical limits of the specified printer model.

![Figure 4.22.7.1: The Print Prep Options dialog.](image-url)
**Duplicate Surfaces**: This condition determines if overlapping surfaces exist in the project. Each object that contains a surface that occupies the same space as another surface will be highlighted.

**Small Objects**: This condition determines if each solid is large enough to print. The user is allowed to specify both the minimum and maximum thresholds. Each threshold is displayed in both world scale and print scale. Solid objects with at least one dimension of its bounding box smaller than the maximum threshold and larger than the minimum threshold are highlighted.

**Thin Objects**: This condition determines if each solid is too thin at some point on the object. For example, an hourglass shaped object might be too thin at its center. The user is allowed to specify the minimum threshold, which is displayed in both world scale and print scale. Solid objects with cross sections parallel to the reference plane smaller than this threshold are highlighted.

**Large Volume**: This condition determines if each solid is large enough so that a significant savings can be gained by hollowing that object. Hollowing avoids having to use unnecessary amounts of print material. The user is allowed to enter a minimum wall thickness in either world scale or print scale. A cube can be used to illustrate this condition. If the solid cube is offset toward the interior by the specified wall thickness and the volume of the remaining, smaller cube is significant than the solid is highlighted.

**All**: When this button is selected all the conditions will be turned on.

**None**: When this button is selected all the conditions will be turned off.

**Display Results**: This option makes it possible to receive information about how many offending objects there are for each selected condition. When on, it invokes the **Print Prep Results** dialog. Next to each condition there is an alpha field where the number of offending objects is displayed. If none are found for a certain condition, 0 is shown.
4.23 Geometric transformations

The geometric transformations affect the coordinate values of one or more points of an object. They have absolutely no effect on the topological structure of an object. They apply equally well to both surface and solid objects. They are the only operators that work in conjunction with two types of modifiers: the Topological Level and the Self/Copy modifiers. The former are discussed in subsection 4.3.1. The following Self/Copy modifiers are discussed in this section:

- Self
- One Copy
- Continuous Copy
- Repeat Copy
- Multi-Copy

Also discussed in this section are the following geometric transformation tools:

- Translate (Move)
- Rotate
- Independent Scale
- Uniform Scale
- Mirror (Reflect)
- Transform
- Repeat Last Transformation
- Macro Transformation 1, 2, and 3

A geometric transformation may be applied to a point, segment, outline, face, complete object, group of objects, or a hole/volume. The type of entity to which it is applied is determined by the currently selected Topological Level modifier. When it is applied to a point, its coordinate values are affected. When it is applied to a segment, the coordinate values of its two endpoints are affected. When it is applied to an outline or a face, all the points that are on their boundary lines are affected. All the points of an object are affected, when it is applied to a complete object.

When a geometric transformation is applied at the Object (or Group) level, it may be applied to the object itself, or to one or more copies of the object. This is determined by the active Self/Copy mode, listed above.
4.23.1 The Self/Copy modifiers

There are four modes that can be set through the Self/Copy modifiers:

**Self**

This tool sets the Self/Copy mode to Self. When this mode is active, all the geometric transformations are applied to the selected entity directly. The Self mode, as used with a move operation, is illustrated in Figure 4.23.1.1.

Double clicking on the Self icon invokes the **Self Options** dialog, which contains a few options specifying how clones should be treated when they are transformed. A complete discussion and illustrations of the options can be found in section 4.23.5.

**One Copy**

This tool sets the Self/Copy mode to One Copy. When this mode is active, all the geometric transformations first make a copy of an object, which is then transformed. The One Copy mode, as used with a move operation, is illustrated in Figure 4.23.1.2.

The **Copy Options** dialog can be invoked by double clicking on this and all the other Copy icons. It contains two options (**Return To Self After Copy** and **Make Clones**) that affect all the copy operations and a group of options that affect the multi-copy operations only. They are discussed with the Multi Copy options at the end of this section.

**Figure 4.23.1.1:** When Self is active, the object itself is transformed: (a) object in original position and (b) after it is moved.

**Figure 4.23.1.2:** When One Copy is active, a single copy of the object is made and is transformed: (a) original object and (b) object with moved copy.
**Continuous Copy**

This tool sets the Self/Copy mode to Continuous Copy. When this mode is active and the topological level is set to Object or Group, the geometric transformation is applied to copies of the picked object rather than to the object itself. One or more transformed copies may be derived by a single execution of the operation. The user may continue to make transformed copies by clicking the mouse at new positions.

The execution of the geometric transformation is progressive, which means that, each time a transformed copy is made, the new copy becomes the object from which the next transformed copy will be derived. The operation is terminated by a double click of the mouse. Continuous Copy can be applied at the Group level, and to any number of prepicked objects simultaneously. The copy mode is illustrated in Figure 4.23.1.3.

**Repeat Copy**

This tool sets the Self/Copy mode to Repeat Copy. When this mode is on and the topological level is set to Object, one or more transformed copies of the picked object may be derived, as with the Continuous Copy mode. The difference is that with this mode each next copy is derived by repeating the transformation parameters that were established by the first application of the geometric transformation.

For all subsequent applications of the operation, the location of the cursor where the mouse is clicked is insignificant and is only used to signal the system that another transformed copy is desired. The operation is terminated by a double click of the mouse. Like the Copy mode, the Repeat Copy mode can also be applied to a group of objects, and to any number of prepicked objects simultaneously. The repeat copy mode is illustrated in Figure 4.23.1.4.
Multi-Copy

This tool sets the Self/Copy mode to Multi-Copy. When active and topological level is at Object, a number of transformed copies are automatically generated, by \( n \) repetitive applications of the geometric transformation. The parameters of the transformation are established by the first and second input points. The value of \( n \), the number of copies made, is set in the Copy Options dialog (Figure 4.23.1.5) that is invoked directly from this modifier. The default for \# Of Copies is 6.

The Multi-Copy modifier can be applied in two different ways, which is selected from the dialog:

**Even Increment**: When this option is selected (default) the parameter (distance, angle of rotation, etc.) established by the first two input points is repeated \( n \) number of times.

**Divide Distance**: When this option is selected, the distance between the two positions established by the two input points, is subdivided by \( n \) to derive between positions, where the copies are placed.

**Big Angle For Rotation**: This option is only available when Divide Distance is on and Rotate is applied. Between the two click points there are two angles, depending on whether the angle is seen in a clockwise or counterclockwise direction. When this option is on, the larger angle is used and divided by \# Of Copies.

Multi-Copy applied to a move operation is illustrated in Figure 4.23.1.6.

**Make Clones**: When this option is on, the copies made are placed into a family of clones. The clones and their behavior are discussed in section 4.23.5.

The prepick method makes it possible to pick entities at different topological levels. When this is the case and one of the Copy modes is active, the geometric transformations will produce copies for all the picked objects, but not for entities lower than the object level. For the latter, the geometric transformations will simply follow the norms of the progressive execution.

**Return To Self After Copy**: This option in the Copy Options dialog, which can be invoked by all the Copy modifiers, affects all the Copy modifiers. When on, as soon as a copy transformation is completed, the Self modifier is activated automatically.
4.23.2 The Transformation tools

The geometric transformations may be executed using either graphic or numeric input. However, given their nature, graphic input is generally superior, because its interactive and dynamic character allows us to observe the changing state of the object and to visually evaluate.

The following paragraphs discuss the graphic input method. To use numeric input, respond to the prompts and use the keyboard to enter the required parameters in the Prompts palette.

Translate (Move)

This tool is used to move a point, a segment, an outline, a face, a complete object, a group of objects, or a hole/volume to a new position. The move can be applied to entities of either facetted or smooth objects and is relative to the active reference plane or relative to the object’s local axes. Which option applies is selected from the Translate Options dialog (Figure 4.23.2.1) that can be invoked from the tool. It contains three tabs, as follows:

The Options tab

Relative To Reference Plane: When this option is on (default), the transformation is applied relative to the active reference plane.

Relative To Object Coordinate System: When this option is on, the transformation is applied relative to the local axes of the first object selected, according to the following additional selection:

XY Plane, YZ Plane, ZX Plane: With one of these options on, an entity is moved parallel to a plane defined by two of the local axes of an object, namely, XY, YZ, or ZX. If the Perpendicular switch is on, the motion is perpendicular to these planes, which is along the Z, X, or Y axes, respectively.

When using the post pick method, with the Move tool you click on the entity to be moved and then on the new position. After the first click, the entity is rubber banded and follows the mouse. It is moved to its new position with the second mouse click (Figure 4.23.2.2).
When using prepick, any number of entities, possibly at different topological levels, are first selected with the Pick tool. Then, with the Move tool active, you click the mouse twice to define the distance of the move. After the first click, all the prepicked entities are rubber banded and follow the mouse. The entities are repositioned with the second click. Moving entities of facetted and smooth objects is shown in Figures 4.23.2.3 and 4.23.2.4, respectively.

**Figure 4.23.2.3:** Moving a point, segment, outline, face, and object, simultaneously: Prepick the entities and with the Move tool click 1 and 2.

**Figure 4.23.2.4:** (a) A smooth point, a smooth segment, a smooth outline, and a smooth face prepicked. (b) Moving them all simultaneously by clicking on 1 and 2.
The **Self Options** and **Copy Options** tabs

Whether the **Translate Options** dialog will contain a **Self Options** or a **Copy Options** tab depends on which Self/Copy modifier is active at the time the dialog is invoked. The content of both of these tabs is shown in Figure 4.23.2.5. Note that the content of these tabs is identical to the **Self Options** and **Copy Options** dialogs invoked directly from the Self and Copy modifiers, as mentioned in the previous section. Their options are discussed in section 4.23.5.

![Figure 4.23.2.5:](image)

**Figure 4.23.2.5:**
(a) The **Self Options** and (b) **Copy Options** tabs of the **Translate (Move) Options** dialog.
The **Smooth Topology Manipulation** options

These options are at the lower portion of the **Translate (Move) Options** dialog, which is shown in Figure 4.23.2.1.

**Facetted Objects**: This pair of options determines how facetted objects will be treated, when transformed. The options are **Keep Facetted** or **Make Smooth**.

**Constrain To Boundaries**: When moving an entity of a smooth object, such as a point or a segment, the whole surface to which that entity belongs is affected. This option determines how any holes that may be contained in the surface are treated. If this option is off, which is the default, the boundary of a hole follows the surface it is on. When this option is on, the boundary of a hole retains its original position and the surface is adjusted from there on. This option is illustrated in Figure 4.23.2.6.

**Convert Arcs To Spline**: When moving points of an arc that are part of a smooth object, when this option is off (default), the arc remains as an arc and is simply scaled up or down to pass through the new position of the point. When this option is on, the arc is converted to a spline, which is extended or shrunk to the new position of the point. Examples of this option are shown in Figure 4.23.2.7.

**Extend Adjacent Faces**: When moving a face of a smooth object and this option is off, the face moves freely without restrictions. When this option is on, the movement of the face is constrained by its adjacent faces and only moves along extensions of the neighboring faces. When all the adjacent faces are perpendicular to the moving face, the face retains its original size. If one or more adjacent faces meet the moving face at non orthogonal angles, the moving face is by necessity scaled up or down as it moves. Examples of this option are shown in Figure 4.23.2.8.

**Interactive**: When this option is on, the motion of an entity is rubber banded as the move operation is executed. It is not otherwise. While normally this option should be on, a movement might involve a very dense object resulting in some very slow rubber banding. In such cases, it is best that this option be turned off.
Figure 4.23.2.7:
Moving a point and a segment of a smooth object:
(a) The original object with a point and segment picked.
   With Convert Arcs To Splines off,
   (b) a point and (c) a segment are moved.
   With Convert Arcs To Splines on,
   (d) a point and (e) a segment are moved.

Figure 4.23.2.8:
Moving a face of a smooth object. (1) An object
   with parallel faces and
   (2) an object with non-parallel faces.
   (a) Original object and picked faces.
   Moving the faces with the Extend Adjacent
   Faces (b) off and (c) on.
Rotate

This tool rotates a point, a segment, an outline, a face, a complete object, a group of objects, or a hole/volume about an axis. What axis is used may vary and depends on the option selected from the Rotate Options dialog, shown in Figure 4.23.2.9. This dialog contains three tabs similar to the Translate (Move) Options dialog. The contents of the Options tab is discussed here. The content of the other tabs is as for the Translate (Move) tool.

Relative To Reference Plane: When this option is on (default), the rotation is about an axis perpendicular to the reference plane, at a location according to the selection from the Axis Through group of options:

Click Point: With this option on, the rotation is about an axis through the point clicked for center (first click after the selection of an entity).

Object Centroid: With this option on, the rotation is about an axis through the centroid of the object or the first object if more than one object is selected.

Object Origin: With this option on, the rotation is about an axis through the local origin of the first object selected. If the origin coincides with the centroid, this option has the same effect as the previous.

Note that the last two options require fewer clicks, since there is no need to select a center of rotation.

Relative To Object Coordinate System: When this option is on, the rotation is relative to one of the local axes of the first object selected, which also needs to be selected: X, Y, or Z Axis.

With postpick, with the Rotate tool active you select the entity to be rotated, then click to establish the center of rotation, click to start the rotation, and click again to complete the rotation. The picked entity is rubber banded after the second click, and the operation is completed with the last click (Figure 4.23.2.10).
With prepick, any number of entities are first picked. Then with the Rotate tool you click three times to establish the center, to start, and to finish the rotation. With prepick, entities at different topological levels can be rotated. Examples with faceted and smooth objects are shown in Figures 4.23.2.11 and 4.23.2.12, respectively.

**Figure 4.23.2.11:** Rotating simultaneously a point, segment, outline, face, and object. The entities are first picked. Then the mouse is clicked on points 1 (center), 2 (base), and 3.

**Figure 4.23.2.12:** (a) A smooth point, a smooth segment, a smooth outline, and a smooth face prepicked. (b) Rotating them all simultaneously by clicking on 3, 4, and 5.
Independent Scale

This tool can be used to scale either dynamically through graphic input or through size parameters. Dynamic execution can be relative to the reference plane or to an object’s local axes. The method to be used is set in the Scale Options dialog (Figure 4.23.2.13) that can be invoked directly from this tool. This dialog has the same three tabs as the Translate Options dialog.

Independent Scale can be applied to all the plain and some parametric objects of form Z, such as c-curves, c-meshes, and nurbs, including nurbs derived from primitives, sweeps, revolutions, and skins. It cannot be applied to analytic primitives, metaballs, text, etc. When you try to do so, a warning message is posted telling you that the object will be converted to plain before executing the operation.

**Scale By Percentage (Dynamic):** When this option is on (default), the Independent Scale tool is used dynamically to scale a point, a segment, an outline, a face, a complete object, a group of objects, or a hole/volume by applying independent scaling factors in the directions of the axes of the active reference plane or the local axes of the object.

When using the postpick method, with the Scale tool active, you click up to four times. The first click selects the object, the second click establishes the base of the scale, the third click starts the scaling, and the fourth click completes the scaling operation. The picked entity is rubber banded after the third click and follows the motion of the mouse. When the prepick method is used, any number of entities are first selected with the Pick tool. Then with the Scale tool you click three times. The Independent Scale operator is illustrated in Figure 4.23.2.14.

Different options are available under the Scale By Percentage option, which determine the entity Relative To which the scale is applied, as follows:

**Reference Plane:** With this option on, the scaling operation is executed relative to the reference plane and its Base At:

**Click Point:** With this option on (default), the base point is at the click point. Note that this is the only option that requires a click for setting the base. All the others do not, which means that they require one less click to execute the operation.

**Object Centroid:** With this option on, the base point is positioned at the centroid of the first object selected. Note that this option is of a hybrid character (it is partially relative to the reference plane and partially relative to the local coordinate system of an object).

**Object Origin:** With this option on, the base point is positioned at the local origin of the first object selected. If it coincides with the centroid, this option is the same with the previous.
Object Coordinate System: When this option is on, the scaling operation is executed relative to the local coordinate system. The base point is always at the local origin of the first object selected. Along which local axis or axes the scaling is executed is determined as follows:

XY, YZ, ZX Plane: Selecting one of these options sets the direction of the scale along the respective pair of local axes. If Perpendicular switch is on, the scaling operation is applied along the single axis that is perpendicular to these pairs of axes, that is, Z, X, and Y, respectively.

Scale By Absolute Value: When this option is on in the Scale Options dialog, the Independent Scale tool is applied through size parameters, rather than dynamically through graphic input. Clicking on an object or another entity with the tool invokes the Modeling Independent Scale dialog (Figure 4.23.2.15), where scaling factors can be entered.

X, Y, Z: When the Modeling Independent Scale dialog is invoked, these fields display the current dimensions of the object’s bounding box in X, Y, and Z. You can change any of these values and, after clicking on OK, the object is resized to the dimensions you entered in the X, Y, and Z fields. This scaling operation is always executed relative to the centroid of the bounding box of the selected entity. You can use this method to resize complete objects, faces, outlines, and segments, but not single points, because they have 0 dimensions. At all other topological levels the displayed dimensions are of the bounding box.

For example, when a segment is positioned parallel to one of the axes of the coordinate system, its bounding box has a single dimension, and the others are 0. When it is parallel to a coordinate plane, but not to an axis, it has two dimensions, and the third is 0. Finally, when it is not parallel to any of the planes, it has three dimensions. Only the dimensions that are non-zero can be changed in the Modeling Independent Scale dialog.

🔒: When this option is on, the X, Y, Z fields are locked to the same value, which makes the scale uniform. When this option is off, different values can be entered in the X, Y, Z fields.

Apply To Each Selected Entity Individually: This option only has meaning when applying scaling to more than one entity, which is done when using the prepick method. When Scale is applied to multiple entities, the values displayed in the dialog are of the bounding box of the group of entities. When this option is off, the values you enter in the X, Y, Z fields are applied to the bounding box of the group. When it is on, they are applied to each individual entity independently. Note that significantly different results are produced when this option is on than when it is off.

Relative To Reference Plane: When this option is selected the directions of the bounding box, which is the basis of the scaling operation, are relative to the reference plane. When this option is off, which is the default, the directions of the bounding box are relative to the world coordinate system. Note that, when relative to the reference plane, the meaning of the X, Y, Z fields is also relative to the axes of the reference plane.
Uniform Scale

This tool is similar to the Independent Scale tool except that it locks all three directions to a single scaling factor. It can also perform a scaling operation dynamically through graphic input or through size parameters. The method to be used is again selected from the Scale Options dialog, shown in Figure 4.23.2.13, which can be invoked directly from the Uniform Scale tool.

**Scale By Percentage (Dynamic):** When this option is on, which is the default, the Uniform Scale tool is used dynamically to scale a point, a segment, an outline, a face, a complete object, a group of objects, or a hole/volume, by applying a uniform scale factors in all three directions (X, Y, Z). The postpick and prepick methods as well as all the suboptions work identically with the Independent Scale. Note, however, that, when the Object Coordinate System is used, the XY, YZ, and ZX Plane options are ignored, and the scaling is always applied to all the directions uniformly. The Uniform Scale tool, as executed relative to Reference Plane and with Base At Click Point, is illustrated in Figure 4.23.2.16.

![Figure 4.23.2.16: Scaling uniformly: With the Uniform Scale tool select the cube (1), then click on 2, 3, and 4.](image)

**Scale By Absolute Value:** When this option is on in the Scale Options dialog, the Uniform Scale tool is applied through size parameters, rather than dynamically through graphic input. Clicking on an object or another entity with the tool invokes the Modeling Uniform Scale dialog (Figure 4.23.2.17), where the scaling factor can be entered.

![Figure 4.23.2.17: The Modeling Uniform Scale dialog.](image)

Rather than the three fields for X, Y, and Z that are available in the Independent Scale Options dialog, this dialog has a single Scale Size field. Other than this the two dialogs have the same options and the two scale operations work the same way.
Mirror (Reflection)

This tool reflects a point, a segment, an outline, a face, a complete object, a group of objects, or a hole/volume relative to an axis of reflection. All types of objects can be reflected except text objects.

The Mirror (Reflect) Options dialog, shown in Figure 4.23.2.18, is invoked directly from the Mirror tool by double clicking on it. From the dialog, the mirror method to be used can be selected. The default is Dynamic. About A Point, About A Segment, About A Surface/Plane, and About Local Plane reflect an object about the respective entity. Any one of these methods can be executed using either the postpick or prepick method.

Dynamic: With the postpick method, select the Mirror tool, then select the entity to be reflected, followed by a mouse click to establish the base point of the reflection. The picked entity is rubber banded together with the axis of reflection, and they both follow the motion of the cursor. The axis of reflection is the perpendicular bisector of the line which connects the base point to the current position of the cursor. When the mouse is clicked again, the mirror operation is completed.

When the prepick method is used, any number of entities are first picked, followed by a click of the mouse to set the base of the reflection. From this point on, the prepick method works as the postpick method. The Dynamic Mirror function is illustrated in Figure 4.23.2.19.
**About A Point, About A Segment, and About A Surface/Plane**: With the postpick method, select the Mirror tool, click on the object to be reflected, and then pick a point, a segment, or a face, depending on the reflection type you are using. The operation is executed immediately, and contrary to the dynamic reflection, it involves no rubber banding.

With the prepick method, first select any number of entities using the Pick tool, then activate the Mirror tool and pick a point, a segment, or a face. All the prepicked entities are reflected in one step. Reflections about a point, segment, and face are shown in Figure 4.23.2.20.

**Relative to Reference Plane**: Reflections are normally executed in all three directions (X, Y, and Z). With this option on, the reflection is executed in only two directions, which are the directions of the currently active reference plane.

**About Local Plane**: When this option is on, the reflection is executed about one of the local coordinate planes of the first object selected, as follows:

**XY Plane, YZ Plane, ZX Plane**: Selection of one of these options reflects an entity about the respective local plane. Note that this option requires a single click to execute a reflection. In postpick mode, the click is on the object to be reflected; it selects the object and executes the operation. In prepick mode, after the object has been preselected, one click anywhere in the project window executes the operation.

With reflections, the Self/Copy modifiers apply to a full extent only with the **Dynamic** option, which allows interactive repositioning of the axis of reflection every time the operation is repeated. With all the other options, a single copy is made for all the Copy modifiers.
The Transform tool is used to change the location, scale, and rotation of an object by transforming the object’s axes graphically. The object can be edited one at a time or multiple objects can be edited simultaneously. Both the prepick and postpick methods can be used with this tool. To transform a single object, with the Transform tool active, the object is picked. To transform more than one object, the Pick tool is used first to preselect them. Then, with the Transform tool, click anywhere in the project window. The selected objects are placed in a special edit mode, where the object’s axes are displayed as transform controls, as shown in Figure 4.23.2.21. Next, one control must be selected with another click. Between this click and the next, the object is transformed interactively. Which type of transformation is executed depends on which control was selected. A second click finishes the current transformation. The process of picking a control and transforming the object may be repeated continuously, without having to exit the edit mode. To exit the transform edit mode, click away from the object in the project window, instead of picking a control.

The following controls are available in the transform edit mode.

**Move center:** When this control is selected, the object is moved parallel to the current reference plane, until the mouse is clicked a second time. An example is shown in Figure 4.23.2.22. Instead of clicking a second time, x, y, and z values for the translation may also be entered in the Prompts palette. These values may represent an absolute or relative translation, depending on whether the “A” (Absolute) check box is selected in the palette.

**Move axes:** These are the line segments that start at the center and extend to the rotation rings. They can be used to move an object along their respective directions. To do so, click the mouse on a move axis. The object moves along the line until the mouse is clicked again. An example is shown in Figure 4.23.2.23. In the Prompts palette, the translation value may also be entered as a relative distance along the selected axis. Note that, unlike the translation parallel to the reference plane described above, the status of the “A” option in the palette is irrelevant, as the shown prompt value is always relative.
**Rotation rings:** There are three rings drawn in the transform edit mode. One of the rings is always drawn in a solid color. It is the active plane of rotation. The other two rings are drawn dimmed. Clicking anywhere on the active ring starts a rotation of the object around the axis perpendicular to the center of the plane of the ring. A second click finishes the rotation. The rotation angle can also be entered in the Prompts palette. Clicking anywhere on a ring that is not active does not start a rotation, but selects that ring as the active ring. The rings and their perpendicular axis are also color coded. In the **Project Color** dialog, which is invoked from the **Options** menu in the main menu bar, the colors titled **Control Editor Color 1**, **Control Editor Color 2**, and **Control Editor Color 3** define the colors for the YZ, XY, and ZX plane of the object’s coordinate system.

Along the active ring, where it intersects the axes, are four point controls. Clicking in any of those four controls also starts the rotation. The start point for the rotation, which is used to measure the angle of rotation, is placed exactly on the respective axis. This is important, when using snaps to align objects precisely. This process is illustrated in Figure 4.23.2.24. The objective is to translate the object to a line and rotate it so that the object’s X axis is aligned with the direction of the line. In step (a), the object is translated by clicking in the center control. (b) Using point snap, the center point of the object is placed precisely on one of the end points of the line. In step (c), the control point on the active ring that is at the end of the object’s X axis is selected. Next, the user switches to segment snapping. Placing the cursor near the line snaps the X axis exactly to the line.

**Figure 4.23.2.22:** Translating the object by clicking the center control.

**Figure 4.23.2.23:** Translating the object by clicking on the axis.

**Figure 4.23.2.24:** Rotating the object by clicking on the ring control.
Resize arrows: These are the portions of the axes that are beyond the rotation rings. Clicking on them scales the object relative to its center, until the mouse is clicked for a second time. The scale transformation is uniform, if the **Uniform Scale** option is selected in the **Transform Options** dialog (see below). If this option is off, the scale transformation is non uniform. Examples of both options are shown in Figure 4.23.2.25

![Figure 4.23.2.25: Scaling the object by clicking on the arrow with Uniform Scale (a) off, and (b) on.](image)

While the controls of the Transform tool appear at a default size, they can be resized. This can be done after clicking on an object with the Transform tool and the controls appear. While at this state, shift clicking anywhere in the project window starts the resizing process. Moving the mouse cursor to the right makes the controls larger. Moving the mouse to the left makes the controls smaller. A second click terminates the resizing action.

Options affecting the Transform tool can be set in the **Transform Options** dialog (Figure 4.23.2.26) that is invoked directly from the tool. It contains two tabs, **Options** and **Self Options** or **Copy Options**, depending on which self/copy modifier is active at the time the dialog is invoked.

![Figure 4.23.2.26: The Transform Options dialog.](image)
The Options tab

Transform All: When a number of objects are prepicked and transformed, if this option is on, they will all be transformed as a group. An example is shown in Figure 4.23.2.27.

Transform Individually: When multiple objects are prepicked and transformed, if this option is on, each will be transformed individually. An example is shown in Figure 4.23.2.28.

Uniform Scale: When this option is on and a scale transformation is applied to an object, it will be applied uniformly, rather than non-uniformly, which is done when this option is off. Examples are shown in Figure 4.23.2.25.

Lock To World Axes: When this option is on, transformations can only be applied in directions parallel to the directions of the world axes.

The Self / Copy Options tab

The options in these tabs are as for all the transformations and are discussed in section 4.23.5.
Repeat Last Transformation

This tool repeats the most recently executed geometric transformation. When the postpick method is used, the Repeat Last Transformation tool is selected first, followed by a pick of the entity to which the transformation repetition will be applied, followed by a mouse click, which establishes the base point of the operation. The repeat operation is executed immediately.

When the prepick method is used, any number of entities are picked first. Then the Repeat Last Transformation tool is selected, followed by a click of the mouse which establishes the base of the operation. The repeat operation is executed immediately.
4.23.3 Transforming and copying holes

Holes/volumes can only be transformed (moved, rotated, scaled, or mirrored) on the face where they are located. When the geometric transformations are executed in conjunction with the appropriate Self/Copy modifier they can be used to produce copies of a hole/volume. Examples are shown in Figures 4.23.3.1 and 4.23.3.2.

Holes can only be transformed when the outlines that correspond to holes of faces are parallel. The prepick method of selection can also be used to move a number of holes/volumes simultaneously. In the latter case all the prepicked holes should be on the same “wall” and all the outlines that correspond to holes of faces should be parallel. Examples of holes that cannot be transformed are shown in Figure 4.23.3.3.

![Figure 4.23.3.1](image1)

*Figure 4.23.3.1:* Moving and copying holes: (a) A 3D enclosure with an opening on one of its walls. (b) The Move tool is used to move the hole. (c) Moving with Repeat Copy on makes multiple copies.

![Figure 4.23.3.2](image2)

*Figure 4.23.3.2:* Rotating holes: (a) A quarter-circle shaped opening is generated on a wall. (b) The Rotate tool is used with Repeat Copy on. (c) Moving with Copy on makes a copy of the window.

![Figure 4.23.3.3](image3)

*Figure 4.23.3.3:* Holes that cannot be moved, because (a) have non parallel outlines, (b) are on different walls.
4.23.4 Macro transformations

A macro transformation is a composite transformation, consisting of a sequence of geometric transformations, which can be applied and generally treated as a single operation.

The steps involved in defining and using a macro transformation are as follows:

- A macro transformation is first defined using either the keyboard (numeric input) or through graphic input. When saved, the macro transformation is also assigned a name.

- A macro transformation is assigned to one of the three Macro Transformation tools available on the eleventh row of the modeling tool palette. Macro transformations can be assigned to all three tools and these assignments can be freely changed at a later time.

- A Macro Transformation tool is used to execute the macro transformation assigned to it.

The Macro Transformations dialog

At the right end of the Geometric Transformations tool palette, there are three icons marked with the numbers 1, 2, and 3. They are the Macro Transformation tools, which are used to execute macro transformations that have been assigned to them. The Macro Transformations dialog, shown in Figure 4.23.4.1, can also be invoked from them.

The Macro Transformations dialog, in addition to containing a list of the names of the currently defined macro transformations, can be used to perform any one of three functions:

- Assign a macro transformation to a Macro Transformation tool.

- Invoke additional dialogs for defining a new macro transformation through keyboard input.

- Edit and change the definition of a previously defined macro. Also, the parameters of a graphically defined macro transformation can be inspected.

The left part of the dialog lists the names of the macros currently defined. Any one of them may be selected by clicking on it. To the right are the three Macro Transformation icons with the names of the macro transformations currently assigned to them displayed. At the lower right are four buttons used to define new, revise existing, change names, or delete macro transformations. Three of them invoke additional dialogs.
Defining macro transformations through the keyboard

After the Macro Transformations dialog has been invoked, the New... button command can be used to define a new macro transformation. The process starts by entering a new name in the macro transformation names list. Once a name has been entered, it can be selected and the Edit... button can be used to define the content of the macro transformation, by entering its parameters through the keyboard. This button can also be used to revise the parameters of a previously defined macro. Selected names can also be changed or deleted using respective button commands.

**New...**: This button invokes the **Name New Macro Transformation** dialog shown in Figure 4.23.4.2. It appears with a default name that consists of the string “Tmac” followed by a numeric index. You can use the default name or type in your own. As soon as you click on the OK button, the name is entered and appears on the name list of the Macro Transformations dialog. These names are listed in alphabetical order.

**Rename**: This button invokes the **Rename Macro Transformation** dialog (Figure 4.23.4.3) which is similar to the Name New Macro Transformation dialog, but which performs a different function. To be able to execute Rename, a macro name must be selected. Otherwise Rename appears dimmed and is inactive. When invoked, the Rename dialog contains the name of the currently selected macro. You can type a new name and click on the OK button, which changes the name in the macro list.

**Delete**: This button deletes the currently selected macro. A macro name needs to be first selected before executing this operation.

**Edit...**: This button invokes the **Macro Transformation Edit** dialog shown in Figures 4.23.4.4 and 4.23.4.5. If the macro is new, the dialog will appear empty. If the macro exists, then the transformations which define it appear in the order in which they are executed. The definition dialog allows you to inspect the exact content and the parameters of the transformations, which can be particularly helpful when the macro was originally defined through graphic input.
In addition to the editable window, the Macro Transformation Edit dialog also contains a pop up menu labeled Type, and button commands New and Delete.

A new entry is made by clicking on the New button. The program uses the transformation operation selected in the Type menu, and produces an entry for that operation. The applicable parameters are also listed and are assigned “identity” parameters, which have no effect when the transformation is applied. The editable numeric fields are shown in red, and can be selected and revised in the usual manner. If no entry is selected when the New button is clicked, a new entry will be made at the end of the list. If one is selected, the new entry will be made right after it. The sequence of transformation entries can also be revised by selecting a field and dragging it to another position in the list. Recall that the sequence in which the geometric transformations are applied is significant, and different results will result when the order of the transformations is changed. The transformations will be executed in the order in which they appear in the Macro Transformation Edit dialog.

Eight transformations are available for constructing macros: move, rotate around each of the three axes, scale, and mirror relative to each of the three orthogonal planes. Any one can be selected by clicking the mouse on the Type menu, then dragging it to the desired transformation, and releasing the mouse. When an entry is selected in the list window, the type field is automatically updated to reflect the operation in the entry. Picking another operation while the entry is selected, changes the operation in the entry. The revised operation appears with default parameters that may need to be set to the proper values. For each operation these parameters are as follows:
Move
Distance: x y z: Distance of motion in each of the three directions.

Scale
Center: x y z: Coordinate position of the center of scale in 3D space.
Scale: x y z: Scaling factor for each of the three directions. Assigning different factors to the three directions corresponds to the Independent Scale operation. Assigning the same factor corresponds to the Uniform scale.

Rotate Around Z
Center: x y: Coordinate position of the Z axis (axis of rotation) on the XY plane.
Angle: Angle of rotation in degrees.

The rotation parameters are illustrated in Figure 4.23.4.6(a). Rotate Around X and Rotate Around Y are as Rotate Around Z with the center positioned on the YZ and ZX planes, respectively.

Mirror in XY
Base: x y: Coordinate position of the base point of the reflection on the XY plane.
Distance: Distance between the base point and the axis of reflection.
Angle: Angle (slope) of the axis of reflection in degrees, measured from the horizontal direction.

The mirror parameters are illustrated in Figure 4.23.4.6(b). Mirror in YZ and Mirror in ZX are as Mirror in XY with the base point positioned on the YZ and ZX planes, respectively.

Figure 4.23.4.6: The parameters of (a) the Rotation Around Z and (b) the Mirror in XY.
Defining macro transformations graphically

Macro transformations can also be defined using a graphic method. This process consists of the graphic execution of a sequence of transformations, which are recorded by the system as they are executed. That is, the system produces entries in its macro definition records according to the tool used and the positions of the mouse. After the completion of the macro definition, the parameters entered through graphic input can be inspected and even revised as for any macro transformation defined through numeric input, by executing the Edit... command in the Macro Transformations dialog.

To define a macro transformation graphically, an entity that will be used for the definition of the macro needs to be preselected. The macro recording process is activated by selecting the Define Macro Transformation tool, which is in the same tool palette as the Self/Copy modifiers.

Define Macro Transformation

This tool turns the macro recording mechanism on, which causes all the subsequently executed geometric transformations to be recorded in a macro definition.

Upon selection of the Define Macro Transformation tool, the Name New Macro Transformation dialog (Figure 4.23.4.2) is presented for entering the name of the macro you are about to define. After entering the name and clicking on the OK button the system returns to the modeling window, and you are ready to proceed with the macro definition process.

You can execute any number of geometric transformations in the usual manner. This is done by selecting the desired transformation tool and applying the transformation with the prepick method. Note that none of the Self/Copy modes may be selected while the recording mechanism is on, and that all the transformations are executed in Self mode. Also note that the positions of the base points used for the execution of the different transformations are significant, since they become part of the macro definition. Of particular significance is the base point of the first transformation, since it is used as the reference point for the macro when the macro is applied.

When the desired transformations have been entered into the macro definition record, the recording mechanism is turned off by deselecting the Define Macro Transformation tool. This can be done either by selecting one of the Self/Copy modifiers (found in the same row as the Define Macro Transformation tool), or by unpicking the entity used for the definition of the macro (select the Pick tool and click the mouse anywhere in the graphics window away from any pickable entity). Note that selecting the Define Macro Transformation tool again indicates the beginning of a new macro definition.
The process of defining a macro transformation is illustrated in Figure 4.23.4.7. It involves the following steps:

(a) Invoke the **Grid Snap Options** dialog from the Snap tools in the window palette and set the **Grid Snap Module** to 0'-8" for all directions. Turn the Grid Snap switch on. Use the **Custom...** item in the **Heights** menu to enter a height of 0' - 8". Set the scale to 1/2" = 1' - 0". Generate a 4' x 16" x 8" rectangular extrusion. The exact shape is unimportant. It is simply used to execute the transformations of the macro. Make a copy of the object on top of itself. This is not a requirement, but it is convenient to have a reference of the original position as the object is transformed.

(b) Prepick the object (which of the two copies is picked is insignificant). Select the Define Macro Transformation tool ( ) and, when the **Name New Macro Transformation** dialog appears, type “Spiral Stair” and click on the **OK** button.

(c) Turn the Perpendicular switch on ( ). Select the Move tool ( ) and click the mouse on points **c1** and **c2**, which moves the object up by 8 inches.

(d) Turn the Perpendicular switch off ( ). Select the Rotate tool ( ) and click the mouse on points **d1** (center), **d2** and **d3**. This rotates the object. Unpick the object to complete the macro definition process.

*Figure 4.23.4.7:* The definition of the “Spiral Stair” macro transformation.
Executing a macro transformation

Macro transformations are assigned to the Macro Transformation tools through the **Macro Transformations** dialog (Figure 4.23.4.1). To do so, select the name of the macro transformation in the names list, and then click on the icon of the tool that appears in the dialog. The name of the macro transformation appears in the text field next to the icon, indicating that the macro has been assigned to that tool.

When no names have been explicitly assigned to the macro icons, the system will automatically assign the first three macros to the icons, in the order in which they are defined. After all three icons have names assigned to them, any additional macros that may be defined are simply entered in the names list, and require user action to be assigned to one of the icons. After a macro transformation has been assigned to a Macro Transformation tool, that tool can be selected to execute the macro transformation assigned to it in the usual manner.

These tools apply the macro transformations that have been assigned to them. If a Macro Transformation tool is selected which has no macro assigned to it, the system will beep and cancel the execution of the operation.

When the prepick method is used, one or more entities are selected, then the Macro tool is selected, and the base point is entered by a click of the mouse. The macro transformation is executed immediately. When the postpick method is used, the Macro tool is selected first, a single entity is picked, and the base point is established by a click of the mouse. The macro transformation is executed immediately.

Figure 4.23.4.8 illustrates the generation of a spiral stair, using the macro transformation defined in the previous subsection (Figure 4.23.4.7). Assuming that the macro transformation has been assigned to the Macro Transformation 1 tool, the generation process involves the following steps:
(a) Using a relatively large scale, the shown “step” is generated, using a 4” height. Note that the position of the step in the coordinates system is significant.

(b) The step is prepicked, Self/Copy is set to Repeat Copy (△), the Macro 1 tool (○) is selected, and the mouse is clicked on point p (the origin of the coordinates system). This produces a second step, as shown.

(c) Repeated clicks of the mouse (anywhere on the graphics window) generate the spiral stair, shown both in wire frame and hidden line plots.

Figure 4.23.4.8: Generating a spiral stair by applying the previously defined macro transformation.

Note that the macro transformations are independent of the level of the entity used for their definition. An entity at any topological level may be used. However, using an object is typically the best practice. Once defined, a macro transformation can be used equally well with entities at any topological level, as can the other transformations.

Note also that spiral stairs can also be generated directly using the Spiral Stair tool.
Using macros to define other macros

A previously defined macro transformation can be used to define other macro transformations. This is illustrated in the examples of Figures 4.23.4.9 and 4.23.4.10.

Working in scale 1/8"=1'-0", with height set to 0' - 8", generate an extruded diamond, as shown in Figure 4.23.4.9(a). Prepick it, activate the Define Macro tool and enter a name for the new macro in the **Name New Macro Transformation** dialog. Select the Scale tool and click the mouse on points 1, 2, and 3 (Figure 4.23.4.9(a)). The result should be as shown in Figure 4.23.4.9(b). Next, select the Macro 1 tool (the macro “Spiral Stair” should still be assigned to it), and click on point 4 (Figure 4.23.4.9(b)). The result should be as shown in Figure 4.23.4.9(c). Select the Pick tool and click the mouse anywhere on the window to clear the pick and to complete the definition of the macro transformation. Because this is only the second definition of a macro transformation, it is automatically assigned to the Macro 2 tool. This process has defined a new macro transformation by using a previously defined macro transformation. The new macro can now be applied in the usual manner, as illustrated in Figure 4.23.4.10.

*Figure 4.23.4.9:* The definition of a macro transformation using the scale operator and the “Spiral Stair” macro.

*Figure 4.23.4.10:* Applying the new macro:
(a) to a diamond by placing the reference point at center, (b) to a hexagon by placing the reference point at center, (c) to a diamond and the reference point on a boundary point, and (d) to a hexagon and the reference point on a boundary point.
4.23.5 Cloning

As already discussed in previous sections, form\-Z provides a number of methods for creating copies of objects. For example, the Move tool can be combined with the Copy modifier to make copies as the object is moved. After a copy is made, the new object is independent from the original object. That is, subsequent operations on the new object do not affect the original and vice versa. It is also possible to make copies that maintain a link between the copied object(s) and the original. When this happens, the original and copied objects are referred to as clones and the process of creating them is called cloning. All the objects that are derived in a single copy operation from one original object are called a clone family. The behavior of a clone family is such that, if an editing change is made to one member of the family, the same change is applied to all the members of the same family.

For example, as shown in Figure 4.23.5.1, an object, representing an architectural column, may be copied 3 times using the Move tool with the Multi Copy modifier and the Make Clones option in the Copy Options dialog turned on. The original object and the 3 copies are a clone family. Next, one of the 4 objects may be selected and a base may be unioned to it. As soon as the operation is applied to one object, it is also applied to all four members of the clone family, all of which now have the same base.

*Figure 4.23.5.1:* (a) A column (b) is cloned 3 times, then (c) a base is unioned to one of the clones. (d) The same base is automatically unioned to all the members of the clone family.
Creating clones

The tools that have the ability to create clones, when copying objects, are: Move, Rotate, Uniform Scale, Independent Scale, Mirror, and Transform. Cloning is done through an option found in the Copy Options dialog (Figure 4.23.5.2) and in the Copy tab of the tool palette of each of the above tools (Figure 4.23.5.3).

Make Clones: When this option is on, clones are created whenever objects are copied. If it is off, plain copies are made, which are independent from each other.

Note that, if the selected object is not part of a clone family and the Make Clones option is on, all new objects resulting from a single execution of a Geometric Transformation tool with a Copy modifier selected, and the original object are formed into a new clone family. If the selected object is already part of a clone family, the copied objects are automatically added to the family of the selected object, regardless of whether the Make Clones option is on.
Transforming clones and tracks

How a clone is geometrically transformed in Self mode, when using the Move, Rotate, Uniform Scale, Independent Scale, or Mirror tool, is controlled by settings in the **Self Options** dialog, shown in Figure 4.23.5.4.

**Transform Clones**: When this option is off, only the selected cloned object is transformed, and no other member of the clone family is affected. That is, the clone behaves as if it were an independent object. If this option is on, the clones are transformed as clones. Selection of one of the two options that follow it determines exactly how the clones will behave.

**Individually**: When this option is selected, the transformation that is applied to the picked clone is also applied to each member of the clone family in a relative manner. For example, if the selected object is rotated about one of its points, each clone of the family is rotated about its corresponding point and by the same angle of rotation. This is equivalent to picking each clone object individually and repeating the rotation manually, while the **Transform Clones** option is off.

**As A Family**: Selecting this option causes all members of the same clone family to be transformed as a group. This is equivalent to first selecting all members of the clone family and then transforming them with the **Transform Clones** option off.

Examples are shown in Figure 4.23.5.5.

**Transform Animation Tracks**: When this option is on and a transformation is applied to an animated entity, it is applied to the entire animation sequence, rather than to a single frame only. This option is discussed in more detail in section 7.2.9.

---

**Figure 4.23.5.4**: The **Self Options** dialog.

**Figure 4.23.5.5**: (a) Five clones. Rotating the right most object with **Transform Clones** (b) off, (c) on and **Individually** on, and (d) on and **As A Family** on.
Working with clones

Once a clone family has been created, a variety of operations, including the editing operations, when applied to a clone affect all the members of the same clone family. In general, there are two groups of tools that affect cloned objects in different ways. The tools in the first group always operate on the original object, rather than producing a new object. Examples of this group would be the Triangulate or Insert Hole tools. When one of these tools is applied to a clone, all the members of the clone family are updated with the same changes.

The tools in the second group create new objects, which are returned as the result of the operation. The original operands are handled according to the settings in the Status Of Objects dialog. That is, the input objects are either left alone, ghosted, or deleted. The Boolean Union or Sweep operation are examples of this group of tools. These tools are further classified as to whether they update clones, if one of the operands is a clone or not. For example, if two objects are unioned, a new object is created and the two input objects may be deleted. If the first object is a clone, all members of the clone’s family are updated with the result of the union operation. If the first object is not a clone, but the second one is, the clone’s family is updated likewise. However, note that only when the clone is a significant factor in an operation, are all the clones updated accordingly. For example, with the Terrain tool, if one of the contour lines is a clone, the clone’s family is not updated with the shape of the resulting terrain object, since a single contour line is typically a very small portion of the overall terrain modeling operation.

Figure 4.23.5.6 contains a list of tools that create new objects and also update clones. Tools that are not listed do not update clones.

Copy / Cut / Paste

When copying one or more clones of the same family with the Copy command in the Edit menu, the relationship of the clones is automatically maintained in the form-Z Clipboard. When the clipboard content is pasted into a new project, the clone links are maintained. If the clipboard is pasted back into the same project from which the clones where initially copied, a new clone family that consists only of the pasted clones is created.
<table>
<thead>
<tr>
<th>Tool</th>
<th>Which clone is updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curve</td>
<td>source object</td>
</tr>
<tr>
<td>Point Cloud</td>
<td>source object</td>
</tr>
<tr>
<td>Loft</td>
<td>first or second object</td>
</tr>
<tr>
<td>Revolve</td>
<td>source object</td>
</tr>
<tr>
<td>Sweep</td>
<td>source object</td>
</tr>
<tr>
<td>Draft Sweep</td>
<td>source or path object</td>
</tr>
<tr>
<td>Helix</td>
<td>source object</td>
</tr>
<tr>
<td>Stair From Path</td>
<td>path object</td>
</tr>
<tr>
<td>Parallel</td>
<td>source object</td>
</tr>
<tr>
<td>Fillet</td>
<td>source object</td>
</tr>
<tr>
<td>Blend</td>
<td>first or second object</td>
</tr>
<tr>
<td>Cage</td>
<td>source object</td>
</tr>
<tr>
<td>Projection</td>
<td>source object</td>
</tr>
<tr>
<td>Controlled Rounding</td>
<td>source object</td>
</tr>
<tr>
<td>Patch Derive</td>
<td>source object</td>
</tr>
<tr>
<td>Patch Grow</td>
<td>source object</td>
</tr>
<tr>
<td>Patch Attach</td>
<td>source object</td>
</tr>
<tr>
<td>Nurbz</td>
<td>source object, if it is a primitive (cube, sphere, cone, cylinder or torus)</td>
</tr>
<tr>
<td>Curve On Surface</td>
<td>surface object</td>
</tr>
<tr>
<td>Untrim Nurbz</td>
<td>nurbz object</td>
</tr>
<tr>
<td>Extend Curve</td>
<td>nurbz curve object</td>
</tr>
<tr>
<td>Boolean Union</td>
<td>first or second object</td>
</tr>
<tr>
<td>Boolean Intersection</td>
<td>first or second object</td>
</tr>
<tr>
<td>Boolean Difference</td>
<td>first object</td>
</tr>
<tr>
<td>Boolean Split</td>
<td>first object</td>
</tr>
<tr>
<td>Line of Intersection</td>
<td>first or second object</td>
</tr>
<tr>
<td>Trim/Split</td>
<td>first or second object</td>
</tr>
<tr>
<td>Stitch</td>
<td>first object</td>
</tr>
<tr>
<td>Trim Lines</td>
<td>first or second object</td>
</tr>
<tr>
<td>Connect Lines</td>
<td>first or second object</td>
</tr>
<tr>
<td>Cover/Uncover Wires</td>
<td>source object</td>
</tr>
</tbody>
</table>

**Figure 4.23.5.6:** List of operations that return new objects and with them they update all the clones.
4.24 Relative geometric transformations

The operations that transform the geometry of one or more objects relative to another object are called relative geometric transformations and are discussed in this section. Specifically, the following tools are discussed:

• Attach
• Extend
• Place
• Align/Distribute
• Replace
4.24.1 Attachments

The attachments are special types of composite or relative geometric transformations and can be applied to either faceted or smooth objects. Objects or parts of objects are geometrically transformed by attaching one of their points, segments, or faces to a point, a segment or a face of another object, or to a reference plane. You execute an attachment by indicating which part of an object is to be attached to what part of another object. The system automatically figures out what geometric transformations need to be applied in order to achieve the requested result. These transformations typically involve a combination of translations, rotations, and scalings.

The attachments typically work in conjunction with the Topological Level and the Self/Copy modifiers. The former determines what level of an entity will be used for the attachment. The latter determines whether a copy will be made or whether the object itself will be attached. When applied to Smooth objects, the attach operation by definition works properly only with straight segments and planar faces. While applying it to curved segments and faces will not be refused, the results may be unpredictable.

**Attach**

This tool is used to attach an entity to another entity or to the reference plane. There are three variations (or Kinds) of the Attach operation that are selected from the **Attach Options** dialog, shown in Figure 4.24.1.1. This is invoked directly from the Attach tool.

**Attaching entities of the same level**

**Per Topological Level**: When this option is on, the attachment is executed according to the active topological level and the options selected from the **Attach Type** and **Adjust To** groups of options. That is, complete objects or parts of objects at different topological levels, the original objects or copies of objects can be attached.

**Attach Type**: This pair of options controls whether complete objects or parts of objects will be attached.

- **Object Part**: When this type of an attachment is selected a point, a segment, or a face of one object is attached to a point, segment, or face of another object, while the other parts of the object remain in their original positions (are not affected by the motion of the attached entity).

- **Entire Object**: When this type of an attachment is selected (default) a point, a segment, or a face of one object is attached to a point, segment, or face of another object in such a way that the object whose part is being attached follows the motion of the attachment.
The attachments require that two entities be picked. The first entity picked will be attached to the second entity picked. (The first is the entity that will be attached, and the second is the entity that will be attached to.) Usually, the two entities belong to two different objects; however, the two entities may belong to the same object.

Both the prepick and the postpick methods are available, but their effects are identical. Given the nature of the attachments, the prepick method can only be used to execute one attachment at a time. When the postpick method is used, the Attach tool is selected first, followed by the selection of the two entities involved in the operation. The operation is executed immediately. When the prepick method is used, the two entities are picked first, followed by the selection of the Attach tool, followed by a click of the mouse anywhere on the graphics window, causing the operation to be executed. If more than two entities are prepicked, the system uses the first two only and ignores the others.

The active Topological Level mode determines which level of entities are picked for the operation. Both entities are required to be at the same topological level. If that condition is not satisfied, the system will beep, issue an error message, and cancel the execution of the operation.

The active Self/Copy mode determines whether the original object will be attached, or whether a copy of the object will first be made and then attached. These are the only two possibilities. The Repeat Copy and Multi-Copy modes have the same effect as the Copy mode. Note that even though the Attach tool involves two objects, it only affects the object picked first. When any Copy mode is active, only a copy of the first object is made.
Point-to-point attachments

Point-to-point attachments are applied when either one of the attachment types is executed with the topological level set to Point. The first point picked is attached to the second point. The Attach Object Part operation only affects the position of the selected point. The Attach Entire Object type affects all the points of the object to which the selected point belongs. Point-to-point attachments involving faceted and smooth objects are illustrated in Figures 4.24.1.2 and 4.24.1.3, respectively. The examples have been executed with the Self/Copy mode set to Copy. For point-to-point attachments, there is only one way in which the points can be picked, and only one way in which the two points can be mapped to each other. This is not true for attachments at the other topological levels.

Figure 4.24.1.2: Point-to-point attachments:
(a) Point 1 is attached to point 2.
(b) The result of Attach Object Part.
(c) The result of Attach Entire Object.

Figure 4.24.1.3: Point-to-point attachments:
(a) Point 1 is attached to point 2.
(b) The result of Attach Object Part.
(c) The result of Attach Entire Object.
Segment-to-segment attachments

Segment-to-segment attachments are executed when two segments are picked. The first segment picked is moved and attached to the second segment picked. Segment-to-segment attachments involving facetted and smooth objects are shown in Figures 4.24.1.4 and 4.24.1.5, respectively.

With segment-to-segment attachments, where the segments are picked is very significant. The locations of the pick points affect the manner in which the two segments are mapped to each other. For each of the segments picked, one of its points is marked as the initial and the other as the terminal point. The initial is the point which is closest to the pick point.

Figure 4.24.1.4: Segment-to-segment attachments:
(a) Point 1 is picked first, then 2.
(b) The result of Attach Object Part.
(c) The result of Attach Entire Object.

Figure 4.24.1.5: Segment-to-segment attachments:
(a) Segment 1 is attached to segment 2 using the Midpoint of Segment option and the two segments are equal. (b) The result of Attach Object Part. (c) The result of Attach Entire Object.
In Figure 4.24.1.6, the objects were selected at pick points 1 and 2. The initial points \((a1)\) of the first segments selected are mapped to the initial points of the second segments selected \((b1)\), and the terminal points of the first segments selected \((a2)\) to the terminal points of the second segments selected \((b2)\).

Note that for every pair of segments that may be picked for the attachment, two distinct results are always possible, depending on where the pickpoints are positioned when the segments are picked. When you execute an **Attach Entire Object** operation, and you are not careful to pick the segments where appropriate, you may derive a result other than what you had in mind. When you execute an **Attach Object Part** operation, if you do not position the pickpoints properly when you select the segments, you may produce a twisted object. While there may be cases where such a result is desirable, in general twisted objects indicate that the segments were not properly picked. The results derived by different positions of the pickpoints are illustrated in Figure 4.24.1.7.
Face-to-face attachments

Face-to-face attachments are applied when either one of the attachment types is executed and the topological level is set to Face. The first face picked is moved and attached to the second face picked. Examples of face-to-face attachments are shown in Figures 4.24.1.8 and 4.24.1.9.

How the two faces are mapped to each other depends on the way in which they are picked. Which segments are picked first when picking the faces, is significant. The segment picked first on the first face is mapped to the segment picked first on the second face. The exact position of the pick point on the segment of the face is of no significance. When the single point pick option is used, then the segment that is closest to the pick point is used as the first segment.

Figure 4.24.1.8: Face-to-face attachments. (a) Points 1, 2, 3, and 4 are picked in this order. (b) The result of Attach Object Part. (c) The result of Attach Entire Object.

Figure 4.24.1.9: Face-to-face attachments: (a) Points 1, 2, 3, and 4 are picked in this order and Center Of Face is on. (b) The result of Attach Object Part. (c) The result of Attach Entire Object.
The faces are always attached in a way such that their positive sides face each other. The number of ways in which two faces can be oriented when they are attached to each other depends on the number of sides of the two faces and the number of possible combinations. For example, as illustrated in Figure 4.24.1.10, there are four distinct ways in which a triangular face can be attached to a rectangle. As with the segment-to-segment attachments, certain ways of picking the faces produce twisted objects when the Attach Object Part type is applied. Such cases are illustrated in Figure 4.24.1.11.

Figure 4.24.1.10: There are four distinct ways in which a triangular shape can be attached to a rectangle.

Figure 4.24.1.11: Face-to-face attachments of object parts: four objects are attached by picking segments (a) in parallel positions and (b) in opposite positions on the objects.
Object-to-object attachments

Object-to-object attachments are applied when the Attach tool is executed with the topological level set to Object. This attachment is given a special interpretation. The first face of the first object is attached to each of the faces of the second object. The object-to-object attachment is always executed in the Copy mode. While both of the attachment types may be used for object-to-object attachments, Attach Object Part will rarely derive meaningful results. Where the mouse is clicked when the objects are selected is of not significant. Examples of object-to-object attachments are shown in Figure 4.24.1.12.

Figure 4.24.1.12: Object-to-object attachments: (a) the objects involved in the attachments: (b) a cuboid attached to a hexagonal extrusion and (c) a cuboid attached to a pyramid.
The justification options

A number of options are available for selecting the manner in which the segments and faces are positioned relative to each other by the attachment operations. They are set in the Attach Options dialog (Figure 4.24.1.1) which is invoked directly from the Attach tool. There are no options for the point-to-point attachments. The options for the object-to-object attachments are the same with the face-to-face attachments.

For segment-to-segment attachments the options are:

**Endpoint Of Segment**: The beginning point of the first segment is positioned on the beginning point of the second segment (Figure 4.24.1.13(a)).

**Midpoint Of Segment**: The midpoint of the first segment is placed on the midpoint of the second segment (Figure 4.24.1.13(b)). This is the default option.

**All Points**: Both endpoints of the first segment are made coincident with the endpoints of the second segment (Figure 4.24.1.13(c)). This option effectively scales the first segment.

**Scale**: The scaling factor that makes the size of the first segment the same as the second segment is applied to the entire first object (Figure 4.24.1.13(d)).

*Figure 4.24.1.13: The Adjust options for the segment-to-segment attachments: (a) Endpoint, (b) Midpoint (default), (c) All Points, and (d) Scale.*

*Modeling • Relative geometric transformations*
For face-to-face attachments the options are:

**Endpoint Of Segment**: The beginning point of the segment that is picked first on the first face, is positioned on the beginning point of the segment that is picked first on the second face (Figure 4.24.1.14(a)).

**Midpoint Of Segment**: The midpoint of the segment that is picked first on the first face is placed on the midpoint of the segment that is picked first on the second face (Figure 4.24.1.14(b)).

**Center Of Face**: The center of gravity of the first face is positioned on the center of gravity of the second face (Figure 4.24.1.14(c)). This is the default option.

**All Points**: All the points of the bounding box of the first face are made coincident with the points of the bounding box of the second face. This option effectively scales the first face (Figure 4.24.1.14(d)).

**Scale**: The scaling factors that make the size of the bounding box of the first face the same as the bounding box of the second face are applied to the entire first object (Figure 4.24.1.14(e)).

*Figure 4.24.1.14*: The Adjust options for the face-to-face attachments: (a) **Endpoint**, (b) **Midpoint**, (c) **Center Of Face** (default), (d) **All Points**, and (e) **Scale**.
Attaching to all faces

**Selected Face To All Faces:** When this option is selected, you use the Attach tool to first select a face and then an object. Copies of the first object are attached through the face you selected to all the faces of the second object. Note that this produces the same result with an object-to-object attachment, except that you now have control of which face of the first object is attached to the faces of the second object. This is illustrated in Figure 4.24.1.15(2). How the two objects were selected for the execution of the operation is shown in Figure 4.24.1.15(2a). Note that the Click On Edges option of face picking is used (two clicks are used to select the face on the first object).

Attaching to the reference plane

**Selected Face To Reference Plane:** When this option is selected, an object can be repositioned relative to the active reference plane, by placing the selected face exactly on that plane. To execute this kind of Attach, with the tool active, select the face of an object and then click somewhere on the reference plane. The point on the reference plane you click on and the Adjust To option selected for Face in the Attach Options dialog determine how the object is positioned. This is illustrated in Figure 4.24.1.16.

For example, clicking on 1 and 2 positions the first object as shown. Assuming the one point face selection option is used, the first click selects the face to be attached and the relative position of that click on the face determines the orientation in which the face will be attached. That is, the segment closest to which we click is placed parallel to the horizontal orientation of the reference plane. Assuming Center Of Face is selected in the Attach Options dialog, the centroid of the attached face is positioned on the click point.

Figure 4.24.1.15: (1) Per Topological Level, object-to-object attachment.
(2) Selected Face To All Faces attachment.

Figure 4.24.1.16: Attaching Selected Faces To Reference Plane:
(a) Clicking as marked to select a face
(b) results in the positions shown.
4.24.2 Aligning and distributing objects

Objects scattered in the 3D space can be aligned along the X, Y, or Z directions and can be distributed at equal distances using the following tool:

Align/Distribute

An alignment, distribution, or a combination of both operations can be executed depending the options selected from the Align/Distribute Options dialog (Figure 4.24.2.1) which is invoked directly from the tool.

The Alignment/Distribution operation requires a minimum of two objects to be executed, but it is typically applied to a large number of objects. When only two objects are aligned and/or distributed, the post pick method can be used. With the Align/Distribute tool active click on the two objects. The operation is executed immediately according to the parameters set in the Align/Distribute Options dialog.

The Alignment/Distribution operation is normally executed with the prepick method. Using either the Pick tool or one of the Select menu commands (Select All, Select By...) preselect the objects you wish to align or distribute. With the Align/Distribute tool active, click anywhere in the window. The operation is executed immediately.

The Align/Distribute Options dialog contains identical parameters for each of the three Cartesian directions (X, Y, and Z). Any combination of these options can be used. For example, the objects can be aligned in one direction, but distributed in another. The dialog also includes a preview window where the effects of the different operations and their combinations can be viewed in one of two modes.

When the dialog is invoked without any objects selected, three fictitious (nonexistent) objects are used in the preview to display the effect of the currently selected combination of parameters. As soon as a different option is selected in the dialog the preview is adjusted to reflect the change. If objects from the project are already preselected when entering the dialog (at least two are required) then these objects, as transformed by the selected operations, are shown in the preview. Again, changing a selection in the dialog is immediately displayed in the preview.

Note that under the preview window are the usual tools for manipulating the view in the window.
No Adjust/Distribution

No operation is applied in the respective direction (Figure 4.24.2.2(a)).

Align At Minimum

The objects are aligned along the minimum of the respective direction (Figure 4.24.2.2(b)).

Align At Center

The objects are aligned along the centers in the respective direction (Figure 4.24.2.2(c)).

Align At Maximum

The objects are aligned along the maximum of the respective direction (Figure 4.24.2.2(d)).

Figure 4.24.2.2: Top views of objects and (a) no alignment applied; aligned in X at (b) minimum, (c) center, and (d) maximum.
Note that in each direction, the alignments are executed independently, but the final result may reflect alignment applied in each direction. For example, if **Align At Center** is applied in all directions, all the objects will be positioned at the center of the project. The center of the project is halfway between the minimum and maximum points of all the objects in the project, for each of the three cartesian directions. To observe the effects of the alignment operations in each direction, you should execute them in isolation. That is, align in one direction while you apply no alignment/distribution to the other orientations. Note that the center is the average of the minimum and maximum in each direction. It can be thought of as the centroid of a box that bounds all the selected objects.

In each direction, the distributions are executed in two modes, as follows:

**Between Ends**: When this option is selected for a direction (X, Y, or Z), the objects at the minimum and maximum positions are determined and all the other objects are placed (distributed) between these two at equal distances. Whether these distances are taken at the minimum, maximum, or center points of the objects depends on the type of distribution operation executed (see below).

**At Distance**: When this option is selected for a direction, the object at the minimum, center, or maximum position of the modeling screen is determined and the others are sequentially placed at distances specified in the numeric field of this option. The objects are placed in the order they were selected. When selected by frame or using the **Select All** menu item, the order in which they were generated becomes their pick sequence. The order of distribution is illustrated in Figure 4.24.2.3. The numeric value shown represents the order in which these objects were selected. After the objects are aligned horizontally, the minimum object 5 and the maximum 3 retain their positions. Then, objects 1, 2, and 4 are distributed left to right, in that order.

To describe how the different distributions are applied, we need to define a few terms:

The **minimum point** of an object relative to a direction is the point with the smallest coordinate value for that direction. The **maximum point** is exactly the opposite. The **center point** of an object relative to a direction is a point whose value is the average of the minimum and maximum values in that direction. The **minimum object** relative to a direction in a modeling scene is the object with the smallest minimum coordinate in that direction. The **maximum object** is exactly the opposite.

---

*Figure 4.24.2.3:* Top views of objects and (a) no alignment/distribution applied, (b) aligned along center in Y. Recall that in **form•Z**'s top view, X is the horizontal and Y is the vertical direction.
and **Between Ends**: Distribution between minima of ends:

With this operation the objects are placed at equal distances between the minimum point of the minimum object and the minimum point of the maximum object. The distance of these two points is divided by the total number of objects that need to be placed plus one. The distances between the placed objects is measured at their minimum points (Figure 4.24.2.4(1a)).

and **At Distance**: Distribution from minimum:

With this operation the distribution distances are measured from the minimum point of the minimum object and is given the value entered in the numeric field (Figure 4.24.2.4(1b)).

and **Between Ends**: Distribution between centers of ends:

With this operation the center points of the objects are placed at equal distances between the center points of the minimum and maximum objects (Figure 4.24.2.4(2a)).

and **At Distance**: Distribution from center:

With this operation the basis of the distribution is the center of the modeling scene. Half of the objects can be placed under it and half over it, at equal distances measured at the centers of the objects (Figure 4.24.2.4(2b)).

and **Between Ends**: Distribution between maxima of ends:

With this operation the objects are placed at equal distances between the maximum point of the minimum objects and the maximum point of the maximum object. The distance of the placed objects, is measured at their maximum points (Figure 4.24.2.4(3a)).

and **At Distance**: Distribution from maximum:

With this operation the basis of the distribution is the maximum point of the maximum object (Figure 4.24.2.4(3b)).
and **Between Ends**: Distribution between ends:

With this operation the objects are placed at equal distances between the maximum point of the minimum object and the minimum point of the maximum object. The objects are placed in the order they were picked (Figure 4.24.2.4(4a)).

and **At Distance**: Distribution from center:

With this operation the base point is the center of the modeling scene, which is the average point of the maximum point of the minimum object and the minimum point of the maximum object. Half of the objects are placed under and half above the center at equal distances specified by the value in the numeric field. Note that with this operation, none of the objects stays in its current position and they are all redistributed (Figure 4.24.2.4(4b)).

![Figure 4.24.2.4: Top views of objects aligned along center in Y and distributed in X.](image-url)
4.24.3 Extending segments

Segments or groups of segments as they end to a face can be extended to another face, using the following tool.

![Extend tool]

This tool is used to extend segments to surfaces or another segment. Variations are selected from the Extend Options dialog (Figure 4.24.3.1), invoked directly from the tool. The Extend tool can be used in both prepick and postpick mode and can be applied to facettted and selected cases of smooth objects.

The options in the Extend Options dialog are organized in two columns, labeled Segment and Face. When an option from the first column is selected, the segments to be extended are selected as individual segments. When an option from the second column is selected, the segments to be extended are selected through the face on which they have their end points.

When the prepick method is used with an option from the Segment column, any number of segments are selected with the Pick tool and the topological level set to Segment. Then with the Extend tool active you click on a face, or on a segment, or anywhere in the graphics window, depending on which type of extension is executed. For example, when extending to a face, then this face needs to be selected. When extending by a percentage relative to the length of a segment then no additional entity needs to be selected. You simply click in the window to activate the operation. An example is shown in Figure 4.24.3.2.

After generating the two objects ("roofs") roughly as shown in (a), with topological level at Segment and the Pick tool active, click on points 1, 2, and 3, to prepick these segments. Then, with the Extend tool active click on points 4 and 5 to select the face to which the segments will be extended (assuming that Clicking on Edges is on in the Pick Options dialog). The result is as in (b). The operation has been described using the prepick method. The same result can be derived by using the postpick method and executing the operation three times, once for each segment.

![Figure 4.24.3.1: The Extend Options dialog.]

![Figure 4.24.3.2: Extending segments to the a face.]

When the postpick method is used with an option from the Segment column, with the Extend tool you click on a segment. If you are extending To Face or To Segment, that face or segment needs to be selected next. For all other options (By percentage or At distance) the single click completes the operation. When the postpick method is used with an option from the Face column, the first (and possibly only) click with the Extend tool is on a face. In this case you are actually selecting for extension all the segments whose end points are on the face you select.

**To Face**: When this option is selected the segment(s) selected for extension, either directly or through the face to which they end, are extended to a face, which is also selected before the operation is completed. The To Face options are illustrated in Figure 4.24.3.3. Extending a face of a smooth object to the face of another smooth object is shown in Figure 4.24.3.4.

**By p %**: When this option is selected the segment(s) selected for extension, are extended by percentage $p$, which is relative to the length of the extended segment. By this option, when multiple segments of different lengths are selected, they are extended by different sizes. Examples are shown in Figure 4.24.3.5.

**At d From Point**: With this option the selected segment(s) are extended by distance $d$, which is an absolute value measured from their end points. With this option all the segments are extended by the same distance, regardless of whether they have equal or different sizes. Examples are shown in Figure 4.24.3.6(2).

**At d From Face**: This option is similar to the previous, except that the distance $d$ is measured from the face to which the selected segment ends. If a segment has an open end (does not end to a face), this option has no effect. When a segment ends to the face at a right angle, this option has the same effect with the previous. Examples are shown in Figure 4.24.3.6(3).
To Segment: Only available in the Segment column, this option allows you to extend a segment to a segment, provided the two segments are on the same plane. If they are not a message is posted. An example is shown in Figure 4.24.3.7.

Adjust Directions: When extending To Face, it is possible to reverse the directions of a solid. This option instructs the program to warn you when such cases occur and ask you if you desire to reverse the directions. An example is shown in Figure 4.24.3.8.

Alert For Intersections: When extending To Face, it is possible to produce self-intersecting objects. With this option on, the program warns you when such cases occur and asks you if you wish to keep the object. An example is shown in Figure 4.24.3.9.

Figure 4.24.3.6: (a) Segment and (b) Face: (1) Picking, (2) At 2' From Point, and (3) At 2' From Face.

Figure 4.24.3.7: Segment To Segment Extension: (a) picking and (b) the result.

Figure 4.24.3.8: A Face To Face Extension reverses directions: (1) Picking, (2) wire frame, and (3) quick paint displays. (a) Without and (b) with the reverse directions corrected.

Figure 4.24.3.9: A Face To Face Extension generates self-intersecting object.
4.24.4 Placing shapes and objects

2D shapes or 3D objects can be placed on other 2D shapes or 3D objects, using the Place tool. While this tool's functionality is useful for a variety of operations, it offers some major conveniences for the nurbs mesh generating procedures. It can be used to place a selected shape on segments or points of a vector line. The object that is placed is called the **source object** and can be an open or closed shape, a meshed surface, or a solid. The object where the source shape is placed is called the **placement object** and can also be an open or closed shape or a solid object. When lines are used, the placement line is not required to be planar and can have any position in 3D space. This is also true for the source shapes. However, use of planar shapes as sources is recommended when c-meshes will be generated, since they allow better control of the c-mesh generation process.

**Place**

This tool is always executed at the Object level and can be used in both prepick and postpick mode, to place objects on other objects. It is affected by the Self/Copy modifier. You can place the source object itself, individual copies, or multiple copies. Which variation to use is selected from the **Place Options** dialog, shown in Figure 4.24.4.1, which is invoked directly from the Place tool. The orientation in which these objects are placed is also selected from the dialog.

**Perpendicular To Line**: When this option is selected, the plane of the source shape is placed perpendicular to the placement line.

For this and the following options, when the object placed is a solid rather than a 2D shape, the plane of the first face is used for determining the orientation of the source object. When an object is placed on a 3D object, the edge we click on becomes the placement line. When multi-placements are executed, all the edges of an object become placement lines.

**Parallel To Line**: When this option is selected, the source shape is placed parallel to the plane of the placement line. While the plane of the placement line is unambiguously defined when the placement line is an open or closed 2D shape that consists of more than one segment, when it is the edge of a solid it will be one of the two faces that touch at that edge. Because of this, this option is more meaningful when placing on lines rather than on solids.
**Along Line:** When this option is selected, the source shape is placed along the placement line and perpendicular to its plane. Again, when the placement line is the edge of a 3D object, perpendicular will be relative to one of the faces that touch on that edge.

**At Normal:** When this option is on, the plane of the source is placed perpendicular to the normal at the placement point. The normal of a surface is a line perpendicular to it. The normal at an edge is the average of the normals of the two faces that end at that edge. The normal at a vertex (point) is the average of the normals of the three or more faces that share that point. Note that this option is more meaningful when placing objects on 3D objects rather than on lines.

**Perpendicular To Ref Plane, Parallel To Ref Plane:** When one of these options is selected, the source shape's orientation is determined by the reference plane, to which it can be perpendicular or parallel. When the latter, it is placed along the X axis of the reference plane.

It should be noted that, while these options offer a variety of possibilities in orienting the placed shapes or objects, some work better with lines and other are more meaningful when placing on solid objects. Which is the appropriate orientation option will depend on the task at hand.

*Figure 4.24.4.2:* Placing shapes and objects:
(1) **Perpendicular To Line,** (2) **Parallel To Line,** (3) **Along Line,** and (4) **At Normal;**
(a) 2D shapes on line, (b) shapes on solid, (c) solids on line, and (d) solids on solid.
Alignment of the source shape and snapping to points

How the source object is aligned relative to the placement point depends on the Alignment option selected from the Place Options dialog. When the source is a 2D shape the alignment options refer to that shape. If it is a 3D object, including a solid, its first face is aligned and placed, and the rest of the object follows. These options are illustrated in Figure 4.24.4.3.

**Centroid**: This option places the centroid of the placement shape on the line. The centroids of open shapes are calculated as if they were closed. This is the default option.

**Reference Point**: This option places the first marked point of the placement shape on the placement point, if a marked point exists. If not, this option defaults to the next one.

**First Point**: This option places the first point of the source shape (or of the first face of a solid) on the placement point.

**Middle Of Open Ends**: This option applies to open source shapes only. It finds the midpoint of the segment that connects the last and first points of the open shape and places it on the line. For closed shapes this option is equivalent to **First Point**.

**World Origin**: When this option is selected, the world origin relative to which the source shape was drawn is placed on the placement point. Note that this is the only “relative” option, which aligns to a point that is not on the placement object.

**Point Snap Within**: When placing the source, if you click on a segment of the placement object, the placement point is in that segment. If you click on a point, that point becomes the placement point. The Snap To Points option is available for facilitating the placement on points. It is complemented by a numeric field where the snapping tolerance is entered. It means that if you click somewhere within a distance from a point that is equal or less to the tolerance, the position of the click is adjusted to the position of that point. When the snap is on, snapping will occur not only when shapes are placed individually, but also when repeated and multiple copies are placed automatically by the system (see below). If any of the positions calculated by the program are found to be within the snap tolerance from a point, that position is adjusted to that point.

![Figure 4.24.4.3:](image_url) (a) Source shapes as drawn and (b) how they are aligned for each option.
Placing copies and multiple copies

The Place tool is affected by the active Self/Copy mode, which determines whether to place the selected shape itself, a copy of the shape, repeated copies of the shape, or multiple copies.

To execute the Place operation (using the postpick method) in Self mode, after the tool has been activated, click the mouse on the source shape and then click on the placement object at the point where you wish the source to be placed. The source is immediately transformed and placed on the line. The One Copy mode works the same way, except that the original source stays in its position, and a copy is placed.

In Continuous Copy mode, execute the operation the same way but, after the first placement, continue to click the mouse on other positions of the placement object. Each click causes a new copy to be placed. You terminate the Continuous Copy mode by double clicking the mouse or by selecting another operator.

The Repeat Copy mode works as the Continuous Copy mode, except that after the first two placements the position where the mouse is clicked is insignificant. Clicks subsequent to the first two simply signal the program to repeat the distance established by the first two placements. This mode is terminated by a double click. Self, Continuous Copy, and Repeat Copy are shown in Figure 4.24.4.4.

The Multi-Copy mode requires one or two clicks after the selection of the source object, depending on the selected method of placement. The options available to the Place tool for multi-placement are independent from the options available for other operations. That is, the Place multi-copy options are not controlled by those in the Copy Options dialog (accessible from the Copy icons), but from options selected from the Place Options dialog.

There are two basic methods by which Multi-Placement can occur:

**Between Selected Points**: This method uses the two positions where the mouse is clicked to determine a distance. It then uses that distance either as an increment that is applied a given number of times (Even Increment), or as the total distance, which it subdivides by the given number to determine the positions where copies of the source will be placed (Divide Distance). For both options, the number of desired placements is entered in the # Of Placements field. This method of placement is similar to that available in the Copy Options dialog.
**End-To-End:** This method places copies of the source from one end of the placement line to the other. This can be done in one of four ways. If the **Use Increment** option is selected, the distance between two mouse clicks is used as an increment and the length of the placement line is subdivided by that increment to determine where between the two ends copies of the source shape will be placed. The number of placements depends on the size of the increment and the length of the placement line. When the **Use # Of Placements** option is selected, the number entered in the respective field is used and that many, evenly distributed copies are placed along the placement line. If the **On Point Bisectors** option is selected, a shape is placed on all the points of the placement line, including the two endpoints of open lines. When the **On Mid-Points** option is selected, one copy of the source shape is placed at the middle of each segment. Note that in this case no shapes are placed at the endpoints of open placement lines. Examples of the multi placement methods are shown in Figure 4.24.4.5.

![Figure 4.24.4.5: Between Selected Points](image)

*Figure 4.24.4.5: Between Selected Points: (a) Even Increment and (b) Divide Distance. End-To-End: (c) Use Increment, (d) Use # Of Placements, (e) On Point Bisectors, and (f) On Midpoints.*

Examples of how these options work with solids are shown in Figures 4.24.4.6 and 4.24.4.7. In the 4.24.4.6, six copies of a small cube are placed on a large cube. The operation is executed twice. The first time, we select the **Even Increments** option and click first on the small cube and then on points 1 and 2 on the large cube. The result is as in (b). The second time we select **Divide Distance** and after we click on the small cube we click on 3 and 4. The result is in (c).

Note that when we multi-place on a solid the sequence of the placements follows the direction of the boundaries of the face. That is, only one face is involved in the operation.
**Figure 4.24.4.6:** (a) Picking for Multi-Placement of 6 copies: (b) at Even Increments and (c) with Divide Distance.

Figure 4.24.4.7 illustrates End-To-End multiplacement. The result in (a) is derived by selecting the On Point Bisectors option and then clicking on the little cone and on the cube. The result is a cone placed at each point of the cube, lined up with its bisector (average normal). The result in (b) shows one more execution of the multiplacement operation. With the On Midpoints option on, we click on the small cylinder and on the cube. A cylinder is placed at the middle of each edge, lined up with the average normal.

The last two options in the End-To-End group, On First Point Of Segs and Per Length Of Seg, are of special interest to the stairs, but can also be useful in other types of models. They are meaningful only when placing solids. The former option places the base of the solid on the first point of a segment and lines up its height along the segment. The latter option does the same and, in addition, it adjusts the height of the placed object to the size of the segment. How the wire columns of stair railings can be turned into solid entities is illustrated in Figure 4.24.4.8.

**Figure 4.24.4.7:** End-To-End placements with (a) On-Point Bisectors and (b) On Midpoints selected. Shown in wire frame and hidden line.

**Figure 4.24.4.8:** Per Length Of Seg Place: (a) Stair generated with wire columns. (b) Source solid (c) placed on wire columns.

**Make Clones:** When this option is on, the copies of the source shape that are placed become clones and members of the same clone family. Note that, contrary to other operations that produce clones, the original source object is not included in the clone family. How clones behave when certain operations are applied to them is discussed in section 4.23.5.

*Modeling • Relative geometric transformations*
Orienting the source shape when placing it

The Place operation always starts by determining the plane of placement, which is perpendicular to either a segment or to the normal of a placement point. It also needs to determine how the plane’s X and Y axes will be positioned. These then are used for adjusting the position of the placement plane according to the Orientation option selected in the Place Options dialog.

The placement plane position and axes are calculated using the plane that is defined by the selected position and the previous and next (non-linear) points of the placement line. The X axis is positioned on the intersection line of the placement plane (the plane of the source shape) and the latter plane. The Y axis is placed perpendicular to the X. This method produces uniform positions and directions for the axes when the placement line is planar or when it makes relatively small turns in 3D space. This is illustrated in Figure 4.24.4.9, where, even though a non planar placement line is used, the directions of the placed shapes turn smoothly.

When a placement line makes abrupt turns, different groups of the source shape will have different orientations. An extreme case is illustrated in Figure 4.24.4.10, where the placement line turns in all directions. Note that the Show Directions and the Show First Point options (Wire Frame Options dialog) are on, which allows us to see how the shapes are positioned. A close inspection reveals that in at least two places the source shapes take radical turns, which would result in a twisted mesh. The only method available in this case is to manually adjust the directions and/or the position of the first point to achieve the desired result. The program will generate positions based on the geometry of the lines in the area where a source shape is placed. The rest must be controlled manually.

Figure 4.24.4.9: The placement of multiple copies of a source shape on a non planar placement line results in smoothly turning directions and a smoothly turning mesh.

Figure 4.24.4.10: When the placement line turns abruptly in all directions, then the directions and/or positions of first points require manual adjustment in order to produce a smooth mesh.
Adjusting the source shape when placed on points

When source shapes are placed on points, their dimensions are adjusted proportionally. This makes it possible to turn the corners smoothly when meshes are constructed. The exact factors used for the adjustment depend on the angle of the point and the orientation of the placed source shape. These adjustments are illustrated in Figure 4.24.4.11.

Generating a nurbz object after placement

The Place tool facilitates the positioning of 2D shapes on a line whose shape outlines a form that needs to be generated. After the placement of a number of shapes, which may be copies of a single shape or different shapes, they can be selected and used as control lines by either the C-Mesh or the Nurbz tool. Given the significance of these operations, the Place Options dialog contains the Generate Nurbs (Copy Modes) option that instructs the system to actually generate a nurbz as soon as the placement is completed. This option affects all of the Copy modes that place more than one copies and requires at least as many source shapes as the depth degree (specified in the Nurbz Options dialog) to be placed before a mesh can be generated. With Continuous Copy and Repeat Copy, the nurbz is generated as soon as the mouse is double clicked to terminate the placement. With Multi-Copy, the nurbz is generated as soon as the placement is completed. The nurbz objects are generated using the parameters currently selected in the Nurbz Options dialog.

Applying a macro transformation when placing

The Place Options dialog also contains an option that allows you to associate a previously defined macro transformation with the execution of the placement. That is, your source shape may be scaled, rotated, moved, or any combination of these geometric transformations, as it is placed on the placement line. Next to the Apply Macro Transformation option, there is a pop-up menu that allows you to select the name of the macro transformation you wish to apply. This must be an already defined macro. Only one macro transformation can be used for a single placement operation. If more than one is desired, then the placement process must be broken down into a number of partial placements. The option Transform First Copy allows you to control whether or not the first copy placed is transformed or retains its original state. The Apply Macro Transformation option affects all the Self/Copy modes.
Examples

The use of the Multi-Copy mode in combination with macro transformations makes it possible to generate an extensive variety of 3D forms, as illustrated in Figure 4.24.4.12. Note that in (a) and (b) the meshes were generated directly by the placement operation. In (c), two placements were made, each using a different macro transformation. The shapes were then picked in the proper order and the Create C-Mesh tool was applied.

The examples in Figures 4.24.4.13 and 4.24.4.14 further illustrate the placement of shapes on non-planar lines. Generating the example in 4.24.4.13 is straightforward. The creation of the twisted cable in Figure 4.24.4.14 required that the directions of the control shapes be manipulated individually after their initial placement on the line with the Place tool. In fact, a few trials were needed before deriving the model shown.
4.24.5 Replace

The Replace tool allows the user to replace one or more objects with another object. The object that will be replacing is referred to as source. The object being replaced is referred to as the destination. This operation is particularly useful when multiple copies of a complex object need to be placed in a modeling scene. Temporary low density objects may be first positioned as place holders during modeling and are then replaced by the real objects at the very end of the process.

This tool may be applied in either prepick or postpick mode. In postpick mode, with the tool active, you first click on the source and then on the destination. The operation is executed immediately and the source replaces the destination object. The source object is moved and rotated in a way such that its axes coincide with the axes of the destination object. When using the prepick mode, more than one object can be replaced with one operation. The multiple destination objects are selected first using the Pick tool, then, with the Replace tool active, you click on the source. All prepicked objects are replaced with the source object.

The Replace tool is affected by settings selected from the Replace Options dialog that is invoked directly from the tool. The dialog is shown in Figure 4.24.5.1.

- **Scale**: When this option is selected, the shape of the source object is also scaled, so that the bounding box of the source object fits onto the bounding box of the destination object.

- **Uniform Scale**: When this option is on, a uniform scale is applied (same scale factor in x, y and z). If this option is off, the source shape is scaled so that the bounding box of the source takes exactly the size of the destination bounding box, in each direction. This typically results in non uniform scaling.

- **Make Clones**: When this option is on, the source and destination objects are formed into a clone family, which works as discussed in section 4.23.5. When multiple objects are replaced, they all become members of the clone family, when this option is on. If the source object is already a clone, the destination objects join the clone family of the source object, even when this option is off.
4.25 Assigning and inquiring about attributes

The following tools that control the assignment of attributes are discussed in this section:

- Color
- Smooth Shade
- Render Attributes
- Texture Map
- Decal
- Set Attributes
- Get Attributes
- Copy Attributes
- Ghost
- Unghost
- Set Layer

The nature of the attributes that these tools handle is different. The first three deal with aspects of color and rendering effects. The next two actually deal with high end rendering and are only available in form•Z RenderZone Plus and are discussed in the respective manual. They are simply mentioned here for the sake of completion.

The pair of Get and Set Attributes handles the whole spectrum of attributes; they are mostly useful for transferring the attributes of one object to another. The Copy Attributes tool directly copies the attributes of one object to one or more other objects.

The Ghost/Unghost pair deals with setting objects inactive and then active again. While ghosting is not a “true” attribute, it is treated as one and is used rather extensively.

Finally, while the layer again is not a “true” attribute, it is treated as one. It is also of prime significance, as it offers very practical methods for organizing the data of a project.
4.25.1 Assigning colors

The modeling objects of form•Z or their faces can be colored using any one of the surface styles contained in the Surface Styles palette. Surface styles are assigned at the Face or Object topological levels only. Each object always has a surface style assigned to it at the Object level, but not necessarily to each of its faces. When an object is rendered in color, the faces that have been assigned surface styles individually are painted with their own color. The faces that have no surface style assigned to them are rendered with the surface style assigned at the Object level. Note that the terms color and surface style are used here in the same sense. While the low end rendering types display objects in a solid “color,” RenderZone, OpenGL, and QD3D can display additional surface properties, such as textures and highlights. The palette which contains the “colors” is called the Surface Styles palette, and the items in the palette are surface styles. However, in the context of this section and throughout this manual, the term color is used also to describe surface display properties.

This tool is used to assign the active color to an object or to its individual faces. Whenever color assignments have been made to individual faces and a new color is assigned to the object, whether or not this color will override all the previously assigned colors depends on the option selected from the Surface Style Options dialog (Figure 4.25.1.1), which is invoked directly from the Color tool.

Keep Current Surface Style Of Faces: When this option is selected, the new color is applied to the object only, and the previously individually colored faces keep their color.

Clear All Face Surface Styles: When this option is selected, all the colors assigned to individual faces are cleared, and the active color is assigned to the entire object.

These options have the same effect when the Color tool is applied at the Face level. This is the most usual way to color individual faces. However, recall that colors are also assigned to individual faces by the Boolean operations. To assign a single color to such an object, the Color tool should be used at the Object level with Clear All Face Surface Styles selected.

When using the postpick method, with the Color tool active select an object or a face, and the active surface style is assigned to it. With the prepick method, use the Pick tool to select any number of objects and/or faces and then, with the Color tool, click anywhere in the window.
4.25.2 Assigning smooth shade to surfaces

All objects are initially created without the smooth shade attribute. When rendered by a display mode that supports smooth shading, such as RenderZone*, the project level smooth shading settings are used to smooth shade these objects, as described in more detail in section 3.6.6. Individual objects can be set to override the project level settings. This is done by applying the Smooth Shade tool with the Override Project Smooth Shading check box selected. The object level smooth shading parameters are the same as those in the Project Rendering Options dialog, with the addition of the All Faces option in the Facetted Objects section.

Smooth Shade

This tool is used to turn on/off the smooth shaded attribute of an object, depending on the selection made from the Smooth Shade Options dialog, shown in Figure 4.25.2.1. This dialog is invoked from the Smooth Shade tool.

As already mentioned above, the Smooth Shade Options dialog has the same content as the Smooth Shading tab in the Project Rendering Options dialog. In the Facetted Objects section it also allows to smooth shade all faces of an object by selecting the All Faces option.

If the Override Project Smooth Shading check box is selected, the options in the dialog will override the options selected in the Project Rendering Options dialog. However, this mechanism only takes effect, if the Allow Object And Layer Override option in the Project Rendering Options dialog is also selected.
4.25.3 Assigning shadow attributes and display types

Render Attributes

This tool is used to turn on and off the shadow casting, shadow receiving, and other rendering attributes of an object. The assignment to be applied or removed is selected from the **Render Attributes Options** dialog that can be invoked directly from this tool and has two tabs. The **Options** tab is shown in Figure 4.25.3.1.

**Casts Shadows**: When this option is selected, the shadow casting attribute of the object to which this tool is applied is turned on.

**Does Not Cast Shadows**: When this option is selected, the shadow casting attribute of the object to which this tool is applied is turned off.

**Receives Shadows**: When this option is selected, the shadow receiving attribute of the object to which this tool is applied is turned on.

**Does Not Receive Shadows**: When this option is selected, the shadow receiving attribute of the object to which this tool is applied is turned off.

Note that it is not enough to turn an object’s shadow attributes on in order for the object to cast or receive shadows. The **Shadow** option in the dialogs, that support the shadow casting and receiving rendering methods and the **Shadow** attribute of the light must also be turned on.

**Object Renders As Wireframe**: When this option is selected, the wireframe type is assigned as an attribute to the selected object. Such objects are rendered as wireframes in **Shaded Render*** and **RenderZone*** display modes.

**Line Width **\( n \) **Pixels**: The value entered in this field determines the line width at which an object is displayed of the **Object Renders As Wireframe** option is selected.

**Object Renders As Shaded Surface**: When this option is selected, the shaded surface type is assigned as an attribute to the selected object. Such objects are rendered as shaded surfaces in **Shaded Render*** and **RenderZone*** display modes.
Object Is Rendering Backdrop: This option is used only by the RenderZone* display mode, when raytracing procedures are invoked. By default this attribute is off. When assigned to an object, it is treated by the raytracing procedures as geometry, which is assumed not to obscure any other objects. As a result raytracing rendering times may be reduced.

It is very important to note that this attribute should be applied only to a few selected objects in a scene. For example, a single large surface object, acting as a ground plane for a building would be a good choice, if that ground plane does not obscure any other objects in the scene. If objects are tagged with this attribute and they obscure other objects, the rendered result will be incorrect. Likewise, if objects with too many faces are assigned as backdrop, the rendering times may actually increase. Typically, the number of faces of all objects with the backdrop attribute would be less than 20. Appropriate and inappropriate use of the backdrop attribute is illustrated in Figure 4.25.3.2.

Figure 4.25.3.2: Use of the backdrop attribute.
(a) Correct: The object is a large ground plane with just a few faces.
(b) Incorrect: The object obscures other objects in the scene. The rendered image is wrong.
(c) Incorrect: Although the object does not obscure other objects in the scene, the object has too many faces. While the rendering is correct, the rendering time went up.
Receives Ambient Occlusion: When this option is on, an object is rendered showing the effects of ambient occlusion. If the option is off, the object will not show any ambient occlusion effects.

Causes Ambient Occlusion: When this option is on, an object causes ambient occlusion on any other object. If the option is off, the object will not causes ambient occlusion. Ambient occlusion is discussed in more detail in section 6.1.5

Note that the wireframe and shaded surface display types can be used individually or in combination with each other. That is they are not mutually exclusive. The Shaded Render\* and RenderZone\* modes display the wireframes as one pixel wide solid lines in the wireframe color. In RenderZone Plus, this color can be edited in the Surface Style Parameters dialog. Figure 4.25.3.3 shows examples of objects rendered as wireframe only, as shaded surfaces only, and as both wireframe and shaded surfaces. Note, that objects rendered as wireframes can also cast shadows. Another example is shown in Figure 4.25.3.4.

Figure 4.25.3.3: Objects in same scene rendered as (a) wire frames only, (b) shaded surfaces, and (c) both wire frames and shaded surfaces.

Figure 4.25.3.4: Parts of a building rendered as wire frames, shaded surfaces, and mixed wire frames and shaded surfaces.
4.25.4 Rendering tools

- Texture Map
- Decal
- Render Textures
- Radiosity Attributes
- Radiosity Bounding Box

These tools are available only in **form•Z RenderZone Plus** and are discussed in the respective User's Manual.
4.25.5 Changing all the attributes of objects

Set Attributes

This tool can be used to set all or as many of an object’s attributes as desired with a single operation. Which attributes will be adjusted, and to what settings, is determined by the selections in the Set Attributes Options dialog shown in Figure 4.25.5.1, which is invoked directly from this tool. The attributes that can be set in this dialog correspond to those that are set through the individual attribute assignment tools, discussed elsewhere in this section.

All: Clicking on this button turns all the attributes in the dialog on.

Get Attributes

This tool can be used to retrieve all or some of the attributes assigned to an object and place them in the Set Attributes Options dialog. Which attributes are retrieved is determined by the selections in the Get Attributes Options dialog, shown in Figure 4.25.5.2. This dialog can be invoked directly from the Get Attributes tool.

All: Clicking on this button turns all the attributes in the dialog on.
4.25.6 Copying attributes from one object to other objects

The Copy Attributes tool is used to transfer attributes from one object to one or more other objects. The object from which the attributes are copied and the object to which they are copied will be referred to as source and destination objects, respectively.

Copy Attributes

This tool can be applied in either the prepick or postpick mode. In postpick mode, with the Copy Attributes tool active, you click on the source object and then on the destination. The attributes of the former are transferred to the latter as soon as the operation is executed.

In prepick mode, you first use the Pick tool to preselect any number of objects. Then, with the Copy Attributes tool active, you click on the source object. The attributes of the source are transferred to all the prepicked objects.

Which attributes are copied is determined by the options selected in the Copy Attributes Options dialog, shown in Figure 4.25.6.1. This dialog consists of a scrollable list, which contains all the attributes currently available to an object. These are attributes always provided by form·Z, attributes provided by plugins that may or may not be installed, and attributes created by the user. User defined attributes are described in more detail in section 3.7.4.
The scrollable list consists of four columns:

- The first column displays a check mark ✓ in front of an attribute, when the attribute is to be copied. If no check mark is shown, the attribute will not be copied. Clicking on a check mark turns it off and clicking on an empty spot in the first column turns the check mark on.
- The second column displays the name of the attribute.
- The third and fourth columns are only used with optional attributes. These are attributes that may or may not exist with an object. For example, an object may or may not have a texture map control attribute. With optional attributes, the icons displayed in the third and fourth columns indicate what actions will be taken relative to those attributes. With required attributes nothing is shown in the third and fourth columns.

One of the attributes in the list is always highlighted and is called the **current attribute**. If the current attribute is an optional attribute, which can be recognized by the contents of the 3rd and 4th columns, its options are shown in more detail in the Optional Attributes section of the dialog. These options apply when an attribute is not present in the source or the destination object, each of which is a separate scenario, as follows:

**In Source, Not In Destination**: This first scenario applies when the attribute exists in the source but not in the destination and contains two options:

- **Copy To Destination** (✓ → ○): If this option is on, the attribute will be created in the destination and then copied from the source.
- **Don’t Copy** (○ → ○): If this option is on, the attribute will not be copied. This option is intended to allow attributes that were assigned to certain objects for a specific reason to not propagate to other objects. For example, an object that denotes a window may have an attribute assigned to it, which specifies the manufacturer of the window. It may not make much sense to copy that attribute to other, non-window objects.

**Not In Source, In Destination**: The second scenario is when the attribute does not exist in the source, but it exists in the destination.

- **Delete From Destination** (○ → ☐): When this option is on, the attribute is deleted from the destination.
- **Don’t Delete** (○ → ○): When this option is on, the attribute remains as is. This option is intended to keep attributes that were assigned specifically to certain types of objects.

The icons that represent each option and are displayed in the 3rd and 4th columns to mark the selected option, are shown above in parenthesis. Clicking on an icon in the 1st, 3rd, or 4th column toggles the option from one state to the other. Clicking in the title bar of the 1st, 3rd, or 4th column toggles the respective option for all the attributes.

*Modeling - Assigning and inquiring about attributes*
4.25.7 Ghosting and unghosting objects

The Ghost and Unghost tools are used to ghost and unghost objects and can be applied using both the prepick and postpick methods. When an object is ghosted, it becomes inactive. Objects may also be ghosted automatically by the system. This occurs when objects are used in an operation and **Ghost** is selected in the **Status Of Objects** dialog (see section 4.5).

When wire frame plotting is used, ghosted objects are shown in a different color, which is defaulted to light gray. This color can be changed in the **Project Colors** dialog. Ghosted objects can be made completely invisible by selecting the **Hide Ghosted** option in the **Wire Frame Options** dialog, invoked from the **Wire Frame** item in the **Display** menu. When any of the other plotting/rendering types are used, ghosted objects are not displayed.

A ghosted object retains its existence, but can not be picked and can not be used by any operator other than the Unghost tool. The only exception is provided by the **Select All Ghosted** and **Select By...** items in the **Edit** menu (see subsection 3.2.4). A ghosted object recovers its complete status as soon as it is unghosted.
4.25.8 Placing objects on layers

The layers represent a method by which objects can be organized in separate groups. Once this is done, objects on one layer may be treated independently from objects on another layer. The layers are discussed in section 2.7.

Set Layer

The Set Layer tool allows entities to be moved from one layer to another, after their initial placement. Recall that a newly generated entity is initially placed onto the layer that is active at the time it is created. After its generation it can be moved to another layer, which can be the active layer or another layer. This is determined by the selection in the Set Layer Options dialog (Figure 4.25.8.1), that is invoked directly from the Set Layer tool.

**Active Layer:** When this option is selected, an entity is placed into the active layer.

**Layer:** When this option is on, the entity is placed into the layer selected from the pop up menu next to it.

This operator can be executed in both prepick and postpick mode. To use it with the postpick method, make the desired selection of the target layer in the dialog, and with the Set Layer tool active click on the object whose layer is to be changed. To use the prepick method, preselect any number of entities, then select the Set Layer tool, and click the mouse anywhere in the graphics window. The Set Layer tool can only be applied at the Object or Group level. Since objects cannot be partially on one and partially on another layer, topological levels lower than Object cannot be handled. The system beeps when the operator is used with such topological levels. In addition to individual layers, groups of layers can also be defined, and the layers can be structured hierarchically at multiple levels. These options as well as the Layers palette and Layers dialog are discussed in detail in section 2.7.
4.26 Delete operations

This section discusses the following three tools for deletions:

- Delete
- Delete Topology
- Delete Geometry

The first tool is applied exclusively at the object level and causes complete objects to be erased from memory. It is equivalent to the delete key. The other two, Delete Topology and Delete Geometry, are used to delete parts of objects: either parts of their topology or their geometry. These can be either individual or groups of points, segments, outlines, or faces.

The Delete Topology and Delete Geometry tools are essentially editing tools that allow you to manipulate the form of 2D shapes or 3D objects. In some ways, they are opposites or counterparts of the insertion operations (see section 4.4). They also relate to the 3D line editing operations (see section 4.21).
4.26.1 Deleting objects, groups, and holes

Complete objects, groups, or holes/volumes can be deleted using the Delete tool, which is equivalent to the delete key on the keyboard.

![Delete](image)

This tool can only be applied at the Object, Group, or Hole topological levels, using either postpick or prepick. Its effect is to clear the object, group, or hole from the project memory.

When using the postpick method, set the topological level to the proper entity (Object, Group, or Hole) and with the Delete tool active click on an object, group, or hole, which is deleted immediately. If the topological level is set to an entity lower than Object, it is interpreted as Object and the whole object is deleted. When using the prepick method, use the Pick tool to preselect any number of objects, groups, or holes. Next, with the Delete tool active, click anywhere in the project window. All the preselected objects, groups, and/or holes are deleted. Any entities preselected at a level lower than Object will be ignored.

When using the postpick method to delete entities with the delete key, they can be at any topological level. When they are prepicked at levels lower than Object, they can be deleted either as topology or geometry. To determine which is desired, as soon as the delete key is pressed after they have been picked, a dialog is invoked (Figure 4.26.1.1) from which you can select to Delete Topology or Delete Geometry.

The Delete tool, when executed at the Object level, resembles the Cut and Clear items in the Edit menu.

Recall that a hole/volume is picked by clicking on a segment of an outline, which is an inner outline of the face to which the hole/volume attaches. This assumes that Clicking On Edges is on in the Pick Options dialog. Examples of hole deletions are shown in Figure 4.26.1.2. The holes/volumes are picked as shown in (a) and the results are shown in (b). Note that attempt to delete the “hole” of the 2D shape failed. However, this 2D hole can be deleted as an outline, using the Delete Topology tool (see next subsection).

![Figure 4.26.1.1: The dialog for selecting Delete Topology or Geometry.](image)

![Figure 4.26.1.2: Deleting holes/volumes with Delete: (a) Picking and (b) the result.](image)
4.26.2 Deleting topological entities

**Delete Topology**

This tool deletes the topology of an entity, which can be a point, segment, outline, or face. Recall that the term topology means the connections of the coordinate points. Deleting the topology of an entity means deleting its connections with other entities. When this tool is applied at the Object, Group, or Hole/Volume levels, it deletes the complete object, group, or hole.

The Delete Topology tool can be applied using either postpick or prepick mode. When postpick is used, with the tool active, you click on the entity to be deleted. The operation is executed immediately. When using the prepick method, you select any number of entities with the Pick tool, you then activate the Delete Topology tool and click anywhere on the project window.

**Deleting topology of points**

The topology of points can be deleted from both incompletely formed surface objects and well formed solids. A point is topologically deleted by eliminating all the connections that end to it. When a point is deleted from a surface object, the deletion breaks the shape, unless the deleted point is an end point of an open shape. Point deletions from surface objects are shown in Figure 4.26.2.1.

When the topology of a point of a well formed solid is deleted, all the connections of that point are deleted. Contrary to the surface objects, when a point’s connections are deleted from a solid object, other connections are rerouted, which preserves the solidity of the object. However, the operation typically produces non-planar faces. For an example is see Figure 4.26.2.2.

For both surface and solid objects, the deletion of a point should not cause the outline to which it belongs, to intersect itself or another outline of the object. If it does, the program beeps, issues an error message, and cancels the operation.

**Deleting the topology of segments**

Segments can be topologically deleted from both incompletely formed surface objects and well formed solids. A segment is by definition the connection of two points. Thus, deleting the topology of a segment deletes the segment itself. Note, however, that the endpoints of a segment survive as geometric entities, after the segment is deleted. When applied to surface objects, segment deletions break a shape, except when the segment is an end segments of an open shape (Figure 4.26.2.3).
When topological segment deletions are applied to well formed solids, all segments of which are double, both parts of a double segment are deleted and the ends of these segments are rerouted, which preserves their solidity. However, such deletions tend to result in non-planar faces (Figure 4.26.2.4).

For both surface and solid objects, the Delete Topology operation must not cause any curves of the object to intersect one another. When they do, the program beeps, issues an error message, and cancels the delete segment operation.

**Deleting the topology of outlines**

The topology of an outline is deleted by deleting all its connections. Only inner outlines (holes) can be deleted as outlines. Outer outlines can only be deleted by deleting the face which they delineate. Deleting the topology of outlines opens “holes” to an object (Figure 4.26.2.5).

The delete outline operation must not cause intersecting faces. If it does, the system beeps, issues an error message, and cancels the execution of the operation. Examples of outline deletions are shown in Figure 4.26.2.5.

**Deleting the topology of faces**

The topology of a face is deleted by deleting all its connections. This operation does not preserve the solidity of an object, but rather opens “holes” to it. Examples of face topology deletions are shown in Figure 4.26.2.6.

The Delete Topology of face operation must not cause intersecting faces, and should not result in face structures that have no meaning, such as the hole of a face being positioned outside the boundaries of the face. If such a condition occurs, the system beeps, issues an error message, and cancels the execution of the operation.

While complete objects are deleted with the Delete tool, it is also possible to cause the deletion of a complete object while executing a topology deletion at a level lower than the object. For example, a 2D shape consists of a single face and a single outline. If that face or outline is deleted, the complete object is deleted. Similarly, if an object consists of a single point or a single segment and these entities are deleted, the whole object is deleted.
4.26.3 Deleting geometric entities

Delete Geometry

This tool deletes the geometry of an entity, which can be a point, segment, outline, or face. The term geometry means the coordinates of a point entity. When such an entity is deleted, any connection to it cannot exist either. Thus, under certain circumstances, geometry deletions also cause topology deletions.

The Delete Geometry tool can be applied using either the postpick or prepick selection method. When postpick is used, set the desired topological level and, with the Delete Geometry tool active, click on the entity you wish to delete. The operation is executed immediately. When the prepick method is used, any number of entities are first selected at appropriate levels, the Delete Geometry tool is activated next, followed by a click of the mouse anywhere on the project window. When this tool applied at the Object, Group, or Hole topological levels it deletes the complete object, group, or hole; in other words it works as the Delete tool.

Deleting the geometry of points

Points can be deleted from both incompletely formed surface objects and well-formed solids. A point is deleted by connecting its previous point to its next. If the point is the first or the last point of an open surface object, that point is eliminated without rerouting any connections. Geometry deletions of points from surface objects are shown in Figure 4.26.3.1.

When the point of a well-formed solid object is deleted, all the occurrences of that point are deleted. This typically affects more than one face and frequently disturbs the closure condition of the solid, which may now be missing a face. When this condition occurs, form-Z automatically generates a face to fill the “hole.” Thus, for well-formed objects, the delete point operation is implemented as a composite operation that first deletes all the instances of a point and then creates a new face, if needed. Geometry deletions of a point from a solid is illustrated in Figure 4.26.3.2.

For both surface and solid objects, the deletion of a point should not cause the outline to which it belongs, to intersect itself or another outline of the object. If it does, an error message is posted.
Deleting the geometry of segments

Segments can be geometrically deleted from both incompletely formed surface objects and well formed solids. Such deletions are executed by deleting the two endpoints of the segment, which can be a double (belonging to two faces) or single segment. The latter is also called an end segment, which is a segment that does not have a reversely coincident segment as a pair (or a segment that belongs to only one face). Deletions of such segments are illustrated in Figure 4.26.3.3.

When applied to well formed solids, which have only double segments, geometric deletions of segments are executed by deleting all the points at the two ends of a segment. When this operation disturbs the closure of a solid (it results in a solid that is missing a face), a new face is generated so that the object may retain its well formed status. This is illustrated in Figure 4.26.3.4.

For both surface and solid objects, the Delete Geometry operation must not cause any curves of the object to intersect one another. When they do, the system beeps, issues an error message, and cancels the delete segment operation. Deletions of segments from 3D solids are illustrated in Figure 4.26.3.5.

Figure 4.26.3.3: Deleting geometry of segments from: (a) closed and (b) open shape. (c) Deleting first segment of open shape.

Figure 4.26.3.4: Deleting geometry of segment from solid: (a) Picking and (b) the result.

Figure 4.26.3.5: Deleting from 3D solids: (a) a segment of coplanar faces; (b) a segment that creates new face.
**Deleting the geometry of outlines**

An outline is geometrically deleted by deleting all its points. Only inner outlines (holes) can be deleted as outlines. Outer outlines can only be deleted by deleting the face which they delineate. Deleting the geometry of outlines preserves the solidity of an object.

The delete outline operation must not cause intersecting faces. If it does, the system beeps, issues an error message, and cancels the execution of the operation. Examples of outline deletions are shown in Figure 4.26.3.6.

**Deleting the geometry of faces**

A face is geometrically deleted by deleting all its points and all their instances. This operation preserves the solidity of an object.

The Delete Geometry of face operation must not cause intersecting faces, and should not result in face structures that have no meaning, such as the hole of a face being positioned outside the boundaries of the face. If such a condition occurs, the system beeps, issues an error message, and cancels the execution of the operation. Examples of face deletions are shown in Figure 4.26.3.7.

While complete objects are deleted with the Delete tool, it is also possible to cause the deletion of a complete object while executing a geometric deletion at a level lower than the object. For example, a 2D shape consists of a single face and a single outline. If that face or outline is deleted, the complete object is deleted.

---

**Figure 4.26.3.6:**
Geometric deletions of outlines:
(1) Picking and the result in (2) wire frame and (3) surface render.
(a,b) Deletions fail (only inner outlines can be deleted)
(c,d) Deletions succeed.

**Figure 4.26.3.7:**
Geometric deletions of faces:
(1) Picking and the result in (2) wire frame and (3) surface render.
All points of picked faces are deleted; solidity of objects is preserved.
4.26.4 Applying multiple topology or geometry deletions

To delete multiple entities from the same object, use the Pick tool to preselect the entities, which can be at different topological levels. Then, with either the Delete Topology or Delete Geometry tool, click anywhere in the window. All the selected entities will be deleted in one step. Multiple entities on multiple objects can also be preselected to be deleted with a single operation.

Note that it is not possible to mix the types of deletions, when preselecting, since the two types (topology and geometry) are executed by independent tools.

Examples of multiple deletions are shown in Figure 4.26.4.1. The illustration displays the results of geometric and topological deletions for the same entities next to each other, which makes a useful comparison. The examples also illustrate how the closure of objects is affected by the different types of deletions and topological levels. That is, whether a deletion produces a hole or not. These results are also summarized in the following table, where √ indicates that the closure remains intact and o indicates that a hole is produced.

<table>
<thead>
<tr>
<th></th>
<th>geometry deletions</th>
<th>topology deletions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>point   segnt outln face</td>
<td>point   segnt outln face</td>
</tr>
<tr>
<td>** solids**</td>
<td>√ √ √ √</td>
<td>√ √ o o</td>
</tr>
<tr>
<td>** meshed surfaces**</td>
<td>o o o o</td>
<td>√ √ o o</td>
</tr>
</tbody>
</table>

*Figure 4.26.4.1:* (a) Selection of multiple entities, (b) geometry and (c) topology deletions applied to (1) a meshed surface and (2) a solid.
5.0 Introduction

The drafting module and its tools are provided primarily in support of the modeling operations. After 3D models have been created, they may need to be communicated for presentation or construction purposes. This frequently requires the production of drawings that contain information in addition to that which can be extracted directly from 3D models. This additional information may take the form of varying line weights and types, notes attached to elements or their parts, dimensioning information, or highlighted textured areas. To further articulate the graphic representations of 3D models, created in the modeling environment, views of these models may be transported into the drafting environment where they can be further refined using the drafting tools.

When modeling views are transported into the drafting environment, they are in a format that can be freely edited and revised, if necessary. Thus the drafting tools can be used to do more than just the production of drafted drawings. They can be used to refine a design. Furthermore, they can be used by themselves to produce new designs. Even though the drafting module's intended function is to facilitate the presentation of the models generated in the modeling environment, you are not restricted from using the drafting tools independently or even exclusively. Design solutions can actually be developed from scratch in the drafting environment, be refined, drafted, and printed, without ever using the modeling environment.

Alternatively, drawings produced in the drafting environment may be transported as surface objects into the modeling environment, to become the base for the generation of 3D models. Even though modeling supports the option to work in a two dimensional space, some users may feel more comfortable with the simplicity of the drafting tools, when they design in two dimensions.

There are functions and operators that are available in the modeling but not in the drafting environment, and vice versa. As you work in the drafting environment, you may use certain modeling tools to support drafting tasks. You do this by producing a shape in the modeling environment and then importing it into the drafting environment.
5.0.1 Drafting elements and their representation

form•Z offers three main types of drafting elements: delineators, annotations, and symbols. Also available in the drafting environment are two special types of elements, the area, which is used to manipulate other elements and does not appear in the final drawing, and the panes, which are essentially windows into a drafting project that allow you to lay out drawings.

There are two categories of delineators: polylines and arcs. Semantically speaking, the polylines and the arcs may be considered as the only real elements of drafting. They are the only ones that can be used for the delineation of physical entities. However, the polylines and the arcs are not restricted to the single role of delineating physical entities, and can also be used to draft any type of an image or notation. The remaining types of elements play a support role.

The polylines and the arcs are distinguished by the way they are represented internally, which also affects their behavior when operations are applied to them. A polyline is an ordered and connected sequence of line segments, which may be open or closed. A polyline may also consist of a single point or a single segment. The internal structure of a polyline resembles the boundary representation used in modeling for the representation of surface objects. Given that a drafting element is, by definition, always a 2D entity, the internal structure of a polyline does not carry a value for Z. In addition, the polylines are not subject to many of the restrictions imposed by the boundary representation, as used in the modeling environment. The direction of a polyline, positive or negative, is of no significance and reversely coincident pairs of segments are not required. Since the world of the polylines is strictly two-dimensional, they are always planar. Examples of polylines are shown in Figure 5.0.1.1.

The topological levels of polylines are the point, the segment, the element, and the compound. They are illustrated in Figure 5.0.1.2. The element level corresponds to the outline level of the modeling environment, which is what a polyline actually is. Structures such as faces that may consist of more than one outline are inapplicable and unavailable. Therefore, the structure of the polylines does not provide for explicit holes, but holes may be inferred by the system. When hatch patterns are applied, certain areas delineated by polylines may behave as holes, depending on their position relative to the areas delineated by other polylines.
A special type of a closed polyline is the rectangle. It has all the characteristics of a polyline, but its response to operations is constrained in ways that guarantee the preservation of its rectangular shape. Rectangles are shown in Figure 5.0.1.3.

Polylines can also be used to draft circular shapes such as complete or partial circles or ellipses. However, form\textsuperscript{Z} also provides the arc as a distinct type of a drafting element. An arc can be open, which corresponds to a partial circle, or can be closed, which corresponds to a complete circle or ellipse. The arcs are internally represented by the parameters that are used to generate them, rather than as a sequence of line segments. The latter is actually the difference between a circular shape that is generated as a polyline and a circular shape that is generated as an arc. The first will be referred to as polyarc and the second simply as arc. As shown in Figure 5.0.1.4, the two may look exactly the same when they are displayed on the screen, but they are represented differently internally, which affects their response to operations.

In contrast to the polylines, the arc has only one topological level: the Element. Therefore, operations such as geometric transformations can only be applied at the Element level. The implication of this is that, unlike the polyarcs, the circular shape of an arc can never be disturbed. Note that the modeling environment does not include an object type analogous to the arc. Instead, circular shapes are approximated by connected sequences of segments, which is analogous to the polylines. This allows them to be used for the generation of planar faces, as required by the boundary representation.

Annotations are symbolic representations that are used to complement the graphic representations of physical entities, with additional information about their attributes and character. There are three types of annotations: dimensions, leader lines, and text. They are illustrated in Figure 5.0.1.5.

Dimensions are used to communicate numerically the size or distance of selected drafting elements. They are generated automatically by the system, after the user has identified the points that need to be dimensioned. A complete dimension element consists of four parts: a dimension line, extension lines, terminators, and text. The dimensions of form\textsuperscript{Z} may be associated, non-associated, or mixed. This means that whenever the shape to which the former are attached is changed, they are also adjusted automatically to reflect the current state of the drawing. The dimensions may be linear, angular, radius/diameter, or arc dimensions. Linear dimensions may be horizontal, vertical, or at any slope.
Leader lines are connected sequences of segments, which may include terminators at their endpoints. They are typically used to relate two or more other elements, or to attach some textual information to an element. One or both of their endpoints may be associated, which is an option controlled by the user.

Text elements consist of one or more alphanumeric strings of characters and can be generated using either outline or bitmap fonts.

Symbols are drafting elements that facilitate repetition. They are of a derivative character and are constructed from polylines, arcs, and text. As they are constructed, they are placed into a library of symbols displayed in a Symbols palette from which they may be selected and repeatedly plotted in a drawing, as many times as necessary.

The different types of drafting elements behave according to their intended functionality when operations are applied to them. To preserve their specific character, they are also restricted from certain operations. For example, the parts of a dimension cannot be moved as individual segments or points. However, all the drafting elements can be exploded to become polylines. When this is done, their displayed images remain as they were before they were exploded, but their internal representations take the form of the polylines, which makes them receptive to all the operations that can be applied to the polylines.
The area is a special type of a drafting element. It is temporary and does not appear in the final drawing. It is used for the manipulation of other drafting elements. By definition, it is always a closed shape and is generated by the same drawing procedures used for the generation of the delineators. The area divides the universal space into two portions: that inside and that outside its boundaries. It can thus be used to identify the elements that are inside, outside, or crossing its boundaries (Figure 5.0.1.6), and supports a special pick function which facilitates many drafting operations. The area can also be used to cut or clip whatever other drafting elements may be crossing its boundaries. After the clipping operation is applied, the portions of the drafting elements that are inside or outside may be further manipulated (Figure 5.0.1.7).

The drafting environment operates in two distinct modes: draft space and layout space. While the former is where most of the drawing work occurs, the latter is where layouts for printing/plotting purposes may be composed. Panes are a special category of draft element used exclusively in the layout space. A pane is a "window" into a project's drafting content (draft space). Any number of panes can be generated and used for displaying details at different scales and orientations. The panes are generated using the usual drawing tools available in the drafting environment and can take any closed shape.

There is also a special image element which is created by importing images from any of the available layer formats. Such elements can be captured from within the drafting or the modeling environment, or in applications external to form•Z.
5.0.2 The drafting tool bar

The default layout of the drafting tool palette is similar to the modeling tool palette. Its default position is also on the left edge of the screen and arranged vertically. It consists of two columns, each containing thirteen rows of icons that can be pulled out to the left or right and can be torn off to become stand alone tool palettes that can be positioned anywhere on the screen. These palettes are shown in Figure 5.0.2.1.

As with the modeling tool bar, the drafting tool bar and its palettes can be completely redesigned and rearranged, including hiding tools that are not needed. This is done through the Customize Tools... item in the Palettes menu (see section 1.9). As in modeling, drafting tool icons marked with a little dot in their upper right corner have a dialog associated with them. You can invoke that dialog by double clicking on the tool. In addition, on the Macintosh you can invoke the dialog by option clicking on the tool. On Windows you can invoke the dialog by right-clicking on the tool or clicking while pressing ctrl+shift.

The positions where the drafting tools are placed in the default tool bar are determined by two factors. The first is their correspondence to the modeling tools, whenever there is a counterpart. The second is the order in which they are expected to be used when drawings are produced. However, note that the drafting elements are less uniform than the modeling objects and their behavior is less sequential.

The presentation of the material in this chapter follows the order in which the tools appear in the default tool bar. Due to the less uniform character of the drafting operators, the presentation is not completely progressive, and some material may be mentioned before it is fully explained. In these cases a reference to where the material is discussed is included. The practice of making each section as self-contained as possible is continued. Consequently, summary discussions of some material have also been included in places other than where the main discussion of that material is provided. However, wherever drafting operations are identical to operations available in modeling and are discussed in detail in Chapter 4, a reference is made to that discussion, rather than repeating it here. In general, this Chapter on drafting assumes that the material on modeling has already been read.
Figure 5.0.2.1: The torn off palettes of the drafting tool bar.
5.0.3 Types of drafting tools

The default drafting tool bar is structured similarly to the modeling tool bar, except that it has one less row. It contains **modifiers** and **operators**. The modifiers are shown in turquoise, and the operators in black.

The drafting modifiers correspond to three of the types of modifiers found in the modeling environment. There are no counterparts to the modeling environment's Insertion modifiers.

**Drafting modifiers**

Three groups of modifiers are positioned in four rows of the drafting tool bar.

**Element type**

These modifiers occupy the first row, left side of the drafting tool bar and determine the type of delineator element that will be generated by the drawing operators.

**Area and pane**

These modifiers occupy the first row, right side of the drafting tool bar and determine the type of special drafting element that will be created. It works in conjunction with the previous palette. That is, only one of the four types can be available at a given time.

**Topological levels**

These modifiers occupy the fourth row, left side of the drafting tool palette and establish the topological level at which a variety of subsequently executed operators will be applied.

**Self/Copy**

These modifiers occupy the tenth row, left side of the drafting tool palette and determine whether a geometric transformation will be applied to the element itself, to a copy of the element, or to multiple copies of the element.
Drafting operators

The remaining rows contain drafting operators. As in modeling, the default drafting tool bar has been structured so that the location of tools corresponds to the inherent order in which operations are expected to be executed. Modifiers only affect tools that are in a lower row and never a tool that is above them.

Polygons, circles, and ellipses

These tools occupy the second row, left column of the drafting tool palette and are used to interactively generate polygonal polylines, or complete circular and elliptical arcs.

Arcs

These tools occupy the second row, right side of the drafting tool bar and are used to generate parametric arcs. These tools generate different types of elements than those in the third row, which generate polyarcs.

Point and segment elements

These tools occupy the third row, left side of the drafting tool bar and are used to generate single point, or single segment elements.

Polylines, splines, and polyarcs

These tools occupy the third row, right side of the drafting tool bar and are used to interactively generate lines, splines, and polyarcs.
Picking

These tools occupy the fourth row, right side of the drafting tool bar. The first operator is used to select elements and/or their parts. The active topological level (fourth row, left side) determines whether a point, a segment, a complete element, a compound, or the area element will be picked. This operator allows elements, and their parts, to be prepicked, so that an operation may subsequently be applied to them. The second operator is used to activate a pane in the drafting layout space environment.

Enclosure

This is the only tool on the fifth row, left side of the drafting tool bar and it generates a derivative enclosure from a previously drawn element.

Parallel offset

This is the only tool on the fifth row, right side of the drafting tool bar and it applies to polylines, including rectangles, leader lines, and arcs. Given a selected element, it creates a new element that is parallel to the selected element.

Boolean operations

These tools occupy the sixth row, left side of the drafting tool bar. The drafting Boolean operations are similar to their 2D modeling counterparts. They can only be applied to closed shapes at the element or compound topological levels.

Compound operations

This is the only tool on the sixth row, right side of the drafting tool bar and it allows you to group or link other previously created elements, including other compounds.
Line editing

These tools occupy the two tool palettes on the seventh row of the drafting tool bar and execute line editing operations such as joining, closing, connecting, filleting, beveling, and inserting.

Text

These tools occupy the eighth row, left side of the drafting tool bar and generate a special type of drafting elements, text.

Symbols

These tools occupy the eighth row, right side of the drafting tool bar and generate, place, edit, and explode symbol instances.

Dimensions

These tools occupy the ninth row, left side of the drafting tool bar and are the operators for placing a variety of dimensions and leader lines.

Dimension attributes

These tools occupy the ninth row, right side of the drafting tool bar and are operators for changing the attributes of the dimensions and leader lines.

Query and measure

These tools occupy the 10th row, right side of the drafting tool bar and are used for querying characteristics of elements, their attributes, and for measuring length and distances.
**Geometric transformations**

These tools occupy the 11th row, left side of the drafting tool bar. They execute translations (moves), rotations, scalings, and reflections. They either operate on the original element or on copies of the element, which is determined by the active Self/Copy modifier (10th row, left side).

**Relative transformations**

These tools occupy the 11th row, right side of the drafting tool bar and are used for aligning and extending elements, as well as for placing elements on other elements.

**Attribute assignment**

These tools occupy the 12th row, left side of the drafting tool bar and are used to assign attributes, such as line type, line weight, and color.

**Ghost and layers**

These tools occupy the 12th row, right side of the drafting tool bar and allow you to ghost or unghost drafting elements as well as to place them on different layers.

**Delete**

This is the only tool in the 13th row, left side of the drafting tool bar and is used to delete elements and/or their parts. The active topological level determines the level at which the Delete operator will be applied.

**Hatch**

This is the only tool in the 13th row, right side of the drafting tool bar and is used to apply a variety of hatch patterns to closed shapes.
5.1 Generating delineators

This section discusses two modifiers that determine the type of delineator that will be generated and a number of drawing tools, as follows:

- Shape
- Enclosure
- Eight tools for drawing closed shapes (rectangles, polygons, circles, and ellipses)
- Six tools for drawing arcs
- Draw Point and Draw Segment
- Six tools for drawing polylines, including splines
- Six tools for drawing polyarcs

There are two categories of delineators: polylines, and arcs. Both of these categories can generate a variety of shapes, which are virtually unlimited for the polylines. The arcs are restricted to the generation of circular shapes only.

The generation of the delineators is controlled by the modifiers in the first row, left side, of the drafting tool palettes. The specific shape of elements is controlled by the drawing tools in rows two and three. There are tools that generate polylines and tools that generate arcs. How the two different types are represented internally is discussed later in this section.

While the drawing tools of the drafting environment generally work in ways identical to those of their modeling counterparts, the drafting environment is more permissive in two significant ways. First, segments of the same polyline are allowed to intersect one another. Second, the direction in which a shape is drawn is of no significance in the drafting environment.
5.1.1 Types of drafting elements

There are two modifiers that control the type of drafting element generated from the drawn shape:

*Shape*

When this modifier is on, the shape you draw becomes the basis for the generation of a polyline or an arc. Whether it is the first or the second type is determined by which drawing tool you use.

The polylines and the arcs are the backbone of the drafting environment. They are the elements that are used for the delineation of the physical entities, which are described in a drawing. However, they may also be used for the creation of any type of a graphic representation or notation. The other types of drafting elements are frequently of a derivative character, hence, they play a support role and are used to complement a drawing with additional alphanumeric or graphic information. A **polyline** is a single point, single segment, or a connected sequence of segments, which may be open or closed. They correspond to the shapes you create with the drawing tools. Examples of polylines are shown in Figure 5.1.1.1.

The polylines loosely correspond to the modeling surface objects. However, they are strictly two-dimensional and, unlike the surface objects, they have no “hidden” third dimension. When views of modeling objects are transported to drafting, they are imported as polylines. This is done by eliminating the third (Z) dimension. When polylines are exported to the modeling environment and become surface objects, a zero value is added for their third dimension. Only polylines can be exported to the modeling environment. All the other types of drafting elements are automatically transformed to polylines (exploded) before being pasted into the modeling environment.

An **arc** is a circular element that is generated on the basis of a **center**, a **radius**, an **initial** and a **terminal angle**. Its internal structure consists of the parameters that are used for its generation, rather than a sequence of segments which approximate its shape. Such an internal representation guarantees that the circular shape of an arc is preserved when operations are applied to it. An arc may be open, which is equivalent to a partial circle, or it may be closed.

The examples of arcs in Figures 5.1.1.1 and 5.1.1.2 illustrate that similar circular shapes can be generated as polylines as well as arcs. To distinguish the two, the first are called **polyarcs** and the second simply **arcs**. While the displays of such shapes appear the same, their internal representations are distinctly different and they behave differently when operations are applied to them.
Enclosure

The double line or “wall” polyline available in drafting is similar to the 2D enclosure available in modeling and is also called enclosure. It can be generated directly or as a derivative element.

When this modifier is selected and one of the drawing tools is used to draw a delineator (polyline or arc), an enclosure is generated. As in modeling, the enclosure is created by generating parallel lines to the right or left or both sides of the shape. Which option applies is determined by the selection in the Enclosure Options dialog (Figure 5.1.1.3) which can be invoked directly from the Enclosure modifier.

**Justification:** This group of options determines where the parallel lines will be generated.

- **Left:** When this option is selected, the line drawn is kept as the left line, and a new parallel line is drawn to its right. For open shapes, right and left is relative to the direction in which they are drawn. For closed shapes, left is to the outside and right to the inside of the shape.

- **Center:** This option generates two parallel lines, to the left and right of the drawn line.

- **Right:** This option keeps the line drawn as the right line, and a new line is drawn to its left.

**Enclosure Wall Width:** This value determines the distance between the two parallel lines. When using Center Justification, each line is half this distance from the shape drawn.

When an enclosure is generated from a closed shape, it consists of two outlines and is generated as a compound element. When it is generated from an open polyline, it is always a single closed shape. When it is generated from a closed arc element, such as a circle, it consists of two closed and concentric arc elements, and is generated as a compound. When an enclosure is generated from an open arc, it consists of two arcs and may or may not include two segments that connect the ends of the arcs, depending on whether the Close Arc Ends option is selected in the Enclosure Options dialog. However, note that, when including end segments, they are independent elements, and the enclosure line is not continuous (Figure 5.1.1.4).

![Enclosure Options dialog](image)

**Figure 5.1.1.3:**
The Enclosure Options dialog.

**Figure 5.1.1.4:** Enclosures generated by drawing (a) a rectangle, (b) a hexagon, (c) an open polyline, (d) an open spline, (e) a closed arc (circle) and an open arc with Close Arc Ends (f) off and (g) on.
5.1.2 Drawing polygonal polylines

The polygonal polylines of the drafting environment correspond to the polygons of the modeling environment. They are connected and closed sequences of segments. All polygonal polylines are drawn as in modeling (see subsection 4.2.2). The rectangle is a special type of a polygon. It consists of two pairs of equal sides that can be drawn parallel to the Cartesian axes or in a rotated position. Contrary to the modeling rectangle, the drafting rectangle is constrained by its internal representation and retains its rectangular shape at all times.

**Rectangle**

This tool draws a rectangle from two input points, which represent a pair of opposite corners. This drawing process is as in the modeling environment, described in subsection 4.2.2. The rectangles drawn by this tool are always generated with their sides parallel to the Cartesian axes. However, they may be subsequently rotated or reflected to take any orientation in the drafting space.

**Rectangle, 3 Point**

This tool draws a rectangle in a rotated position from three input points. The first point determines the position of one corner point of the rectangle. The second point determines its direction and the third input point determines the final size of the rectangle.

While the rectangles are drawn exactly as in the modeling environment, in drafting they are internally structured differently from the polygons. This allows them to preserve their rectangular character and to behave accordingly when geometric transformations are applied to them (see subsection 5.13.4). That is, when points or segments are moved, they are constrained to take positions that preserve their orthogonal shape. This is true for both rectangles parallel to the orthogonal axes and rotated rectangles. Examples of rectangles are shown in Figure 5.1.2.1.

---

**Figure 5.1.2.1**: Rectangles: (a) parallel to the axes, (b) rotated, and (c) reflected.
Polygon

This tool is used to draw closed polygonal shapes exactly as its counterpart in the modeling environment does. It invokes a dialog where the number of sides of the polygon to be drawn may be selected. This tool generates symmetric and regular polygons of 3, 4, 5, 6, 8, 10 and n-sides from two input points.

This tool invokes the **Polygon Options** dialog shown in Figure 5.1.2.2. This dialog contains options on how the polygons will be drawn, what the number of polygon sides will be, and if a pattern for the polygon will be used. The polygons are stored internally as closed polylines. Unlike the rectangles, the polygons are not constrained when geometric transformations are applied to them. Consequently, their shape may be changed freely, after they have been generated. The set of polygonal polylines is shown in Figure 5.1.2.3. The options in the **Polygon Options** dialog are as with its modeling counterpart which is discussed in subsection 4.2.2.

![Polygon Options dialog](image)

*Figure 5.1.2.2:* The drafting **Polygon Options** dialog.

![Set of polygonal shapes](image)

*Figure 5.1.2.3:* The set of polygonal shapes that can be generated directly. The last shape to the right was generated using the **Polygon By # Of Segments** option set to 24.
5.1.3 Drawing arcs

Arcs are parametric circular shapes. Complete circles and ellipses can be generated using tools on the second left row, and open arcs can be drawn using tools on the second right row.

These tools are used to generate circular shapes, which are internally represented as arcs. Complete circles are generated in three different ways. Ellipses can be generated in two ways. The drawing methods are identical to those in the modeling environment (see subsection 4.2.3).

All the above tools invoke the Arc/Circle/Ellipse Options dialog (see Figure 5.1.3.1) whose options are as follows:

By # Of Segments: When this option is selected, the resolution of the edges and iso lines is determined by the number of segments entered in its numeric field. The default is 24.

By Max Size Of Segments: When this option is selected, the resolution of the arcs, circles, and ellipses is not a constant number, but depends on their length. The larger the radius, the higher the resolution.

The program generates as many segments of the size $s$ indicated in its numeric field as it can fit. When drawing such shapes the program always starts by rubber banding a triangle. This changes to a rectangle as soon as it can fit four sides that are of size equal or greater than $s$. Open arcs have a minimum of three segments always.

Recall that the By # Of Segments parameter controls the resolution of both the polylines and the parametric arcs. For the former it controls how many straight line segments are used for the internal representation of the circle. When a circle is a parametric arc, it still needs to be approximated with line segments when it is plotted. The number of segments used when it is displayed is also set by the above parameter. Note that, for the arcs, this is not an attribute attached to individual arc elements, but a display option. Consequently, when it is changed, all the arcs currently displayed are replotted according to the new number of segments.
These tools are used to generate circular parametric shapes that are partial circles. These are six distinct tools that offer six variations for drawing arcs. These work as the corresponding tools in modeling do.

Note that these tools look the same as another set of tools found on the third right row together with the polylines and splines. Both sets work and draw the same way. However, the tools here generate parametric arcs while the tools on the third row generate polyarcs.
5.1.4 Drawing points and segments

The two tools on the third row, left side of the drafting tool bar are used to directly create single point and single segment elements. Note that point and segment elements cannot be drawn with the area or pane modifiers active. If you attempt to create them with either of these modifiers active, an error message will be presented.

Point

This tool behaves as its modeling counterpart. Clicking in the graphics window with this tool active creates a single point elements as discussed in subsection 4.2.4. The type of single point element it will create is dependent on which Element Type modifier is selected.

Segment

This tool also behaves as its modeling counterpart. With this tool active, clicking twice in the graphics window creates an element from a single segment. This is as discussed in subsection 4.2.4. Recall that single segments can also be created using the Polyline tool, but you must double click on the second point of the segment with that tool to complete the operation.
5.1.5 Drawing polylines and splines

The tools in the third right row of the tool bar are used to generate polylines, which may be of any regular or irregular shape. They may be open or closed. They are drawn through the input of a sequence of points, which is terminated by a double or a triple click of the mouse, or by typing “e” or “c” in the Prompts palette, when numeric input is used. The double click (or “e”) produces an open polyline. The triple click (or “c”) closes it, by connecting the point entered with the triple click to the first point. In general, these drawing procedures work as their counterparts in the modeling environment (see subsection 4.2.4).

These tools generate polylines from a sequence of points, as discussed in subsection 4.2.4. Recall that each of these tools can be used by itself, or, by changing tools during the drawing process, connected sequences of polylines may be produced, as illustrated in Figure 5.1.5.1(b).

As already mentioned, the polyarc generating tools appear identical to the arc generating tools in the second right row. They both work exactly the same and produce partial circles that appear the same when displayed. However, they are internally different, which affects how they behave when operations are applied to them. In addition, the polyarcs can be included in drawing sequences derived by switching tools, while the arcs cannot.

**Figure 5.1.5.1:**
(a) Polylines drawn with single tool. (b) Shapes generated by switching drawing tools.
5.2 Drawing the area and panes

This section discusses the two modifiers on the right side of the first row of the tool bar, namely:

- Area
- Pane

These are both special types of drafting elements that are generated using the same drawing tools with which the regular elements of the drafting environment are drawn (see previous section). Both the area and a pane are closed shapes. There can only be one area element, but any number of panes. Panes can only be generated in layout space. Even though these entities are discussed together, they perform very different functions, as discussed in the remainder of this section.
5.2.1 Generating and using the area

The area is a closed shape drawn with any combination of drawing tools except Point and Segment, with the following modifier icon.

![Area](image)

When this modifier is active, the closed shape drawn by the user becomes the current area element. The area is a drafting element that is of a temporary character. It is not used for the delineation of entities, but rather for the identification of other drafting elements, to which some operation is subsequently applied. The area can be of any shape, but is required to be always closed. When the user draws an open polyline the system closes it automatically by connecting its two end points. When the user draws an open arc, it is rejected by the system, since it cannot be closed. Examples of areas are shown in Figure 5.2.1.1.

![Figure 5.2.1.1: Areas drawn as (a) a rectangle, (b) a circle, (c) an irregular polyline, and (d) a self-intersecting polyline.](image)

The area is displayed on the screen with a gray fill, which clearly distinguishes it from the other drafting elements. Gray is the default color for the fill and can be changed to another color through the Project Colors dialog, as can all the other system colors (see subsection 2.6.3).

There is only a single area element per project. Generating a new area element replaces the previous one, if one exists. The area is delineated by a boundary shape that splits the universal space to two portions: that inside and that outside the boundary. This allows the area to be used for the identification of groups of drafting elements that are contained within it, are outside it, or cross its boundary. This leads to the area's main functionality, which is to pick other drafting elements on the basis of their position, relative to its boundary shape.

The area can also be used as a clipping tool. By selecting the Clip To Area option in the Area Options dialog, elements that cross the boundaries of the area are broken at their points of intersection, when the area is picked. Then an operation may be applied to the portions that are either inside or outside the boundary of the area. The Area Options dialog is invoked directly from the Area Generation and the Area Pick tools. It is shown in Figure 5.2.1.2. The functionalities of the area are discussed in subsection 5.3.3.

![Figure 5.2.1.2: The Area Options dialog.](image)

---

*Drafting • Drawing the area and panes*
5.2.2 Generating and using panes

The drafting environment consists of two distinct work spaces, draft space and layout space. Draft space is where the actual drafting functions take place, whether it be importing from modeling or actual production. Layout space is the mode which allows you to composite drawings.

When in layout space you can generate any number of panes, which may be thought of as windows of the layout space that open into the draft space. They can only be created and edited within the layout space and can be any closed shape. They can display different views, different scales, different properties, and each can be edited independently.

The layout space is engaged and disengaged by selecting and deselecting the Draft Layout Mode from the Display menu. When you select this item for the first time in a draft project, the contents of the project in the draft space disappear. You can draw elements in the draft space and generate panes, as discussed below. Multiple panes can be generated, each displaying the content of the draft space. After their initial generation, you can manipulate the displays in the panes to show different details, views, and scales of your draft project. If you turn off Draft Layout Mode and you then turn it back on again, the panes and their content as they were at the time you left the layout space will reappear, allowing you to continue your layout work from where you left it.

Although the layout space is separate from the draft space, they do share attribute lists (colors, layers, line weights, and line styles). If an attribute is changed in the draft space, the layout space and its panes will be automatically updated, and vice versa. There is one important exception to this rule. Within the Layers palette, the visibility, selectability, and snapability attributes are independent for each pane. If you are working within a pane and you change the visibility of a layer, it will only happen within the active pane. However, if you delete, add, or rename a layer, the change will occur in the draft space and all of the panes in layout space.

Pane

A pane can be drawn as any other drafting element. When this modifier is active and the layout space is on, the shape drawn becomes a pane element. Any number of panes can be drawn using any of the drawing tools, provided that the operation results in a closed shape. Although rectangles are most useful, form•Z allows any closed shape for presentation flexibility. For example, you may want to use an unusual shape to crop an area of the project, or you may want to use fancier shapes to graphically enhance the presentation. Except for the options associated with the drawing tools, there are no other options and no dialog is invoked from this modifier. The content of the draft space is fit into the bounding rectangle of the pane and is displayed in the pane. If the pane is rectangular, the bounding rectangle coincides with the pane. If it is not, the bounding rectangle is a different shape.
A pane is not immediately active after it is created. It must be selected before you can make any adjustments to the scene currently being displayed within it. The Activate Pane tool (found next to the Pick tool) must be used to activate a pane (see subsection 5.3.4). Clicking on a pane with this tool makes it the **active** pane. There is only one active pane. The active pane is where all the input is directed, which includes editing its view, layer visibility, and display options. The pane also accepts input for changing the elements displayed in it. In other words, you can change data in draft space through the active layout space pane.

When a pane is active, you cannot make any changes within the layout space. All operations will affect the drawings in the draft space showing within the active pane. However, if there is no active pane, you can draw directly into the layout space. This allows you to complement your presentation with lines, shapes, and text which is not part of the project image. As already mentioned, the layout space is primarily intended for the preparation of drawings before they are sent to a printer or plotter. An example is shown in Figure 5.2.2.1.

*Figure 5.2.2.1:* A project in draft layout space showing different views, layers, and project information. Shown drawings are by Richard Reynolds, Los Angeles, California.
5.3 Picking

The following modifiers and operators that relate to picking entities are discussed in this section. The top six are modifiers and the last two are operators.

- Point
- Segment
- Element
- Compound
- Automatic
- Area
- Pick
- Pick Pane

As in the modeling environment, the drafting elements frequently need to be *selected* or *picked*, so that some operation may be applied to them. The basic pick operation works in a manner very similar to the modeling pick. It may be applied at different topological levels, in either a prepick or a postpick mode. However, the functionality of the drafting pick is less uniform than its modeling counterpart. A number of behavioral variations are dictated by the specific character of the different types of drafting elements. These variations are discussed in subsequent sections.

In the drafting environment, elements can also be picked through the use of the *area*, a feature that does not have a direct counterpart in the modeling environment. The area is delineated by any closed shape generated with one of the drawing tools. It can be used to identify elements that are inside, outside, or are crossing its boundary. It can also be used to apply a clipping operation to elements that cross it. These functions are discussed in subsection 5.3.3.

The last operator is a special pick function that activates a pane in layout space. Recall that panes are only meaningful and available in layout space and thus this tool can only be used in that space.
5.3.1 The topological levels

The Pick tool works in conjunction with the topological level modifiers, located on the fourth left row of the drafting tool bar.

Point

This modifier sets the current topological level to Point. A point is picked by clicking the mouse on it or sufficiently close to it. The picked point is highlighted with a cross shaped mark. As in the modeling environment, red is the default highlight color, which can be set through the Project Colors dialog (see subsection 2.6.3). Points can be selected from polylines, rectangles, leader lines, and the boundaries of areas, panes, and image elements. Some of these cases are illustrated in Figure 5.3.1.1. Points cannot be picked from arcs, dimensions, and symbols.

Segment

This modifier sets the current topological level to Segment. A segment is picked by clicking the mouse anywhere on it, or sufficiently close to it. The picked segment is highlighted. Segments can be selected from polylines, rectangles, leader lines, and the boundaries of areas, panes, and image elements. These cases are illustrated in Figure 5.3.1.2. Segments cannot be picked from arcs, dimensions, and symbols.

Figure 5.3.1.1: Picking a point from (a) a polyline, (b) a rectangle, (c) an area drawn as a rectangle, and (d) a leader line.

Figure 5.3.1.2: Picking a segment from (a) a polyline, (b) a rectangle, (c) an area drawn as a rectangle, and (d) a leader line.
Element
This modifier sets the current topological level to Element. In general, elements are picked by selecting one of their segments. There are two exceptions: image elements and symbols are picked by clicking inside their bounding box. Picked elements are highlighted with the current highlight color. All types of elements can be picked at the element level, as illustrated in Figure 5.3.1.3.

Compound
This modifier sets the current topological level to Compound, which is essentially a composite or linked element consisting of two or more drafting elements. The Compound topological level is similar to the Group of the modeling environment, in that it provides a method for organizing elements together, and it maintains a hierarchy.

The tool which allows you to create compound elements from existing drafting elements is discussed in subsection 5.5.2. Compound elements can also result from operations such as the Boolean operations and the Enclosures. They are discussed in subsections 5.5.1 and 5.4.1, respectively.

A compound element is picked by clicking on any of the elements which it contains, with topological level set to Compound. Compound elements may include a number of hierarchical levels. However, they are always picked at the highest level, regardless of the level that the selected element represents in the hierarchy.

The Compound topological level treats individual elements that are not part of a compound as a compound that consists of a single element. Consequently, such elements can be picked individually with topological level set to Compound. The components of a compound can be picked with topological level set to Element. Similarly, parts of a compound, such as points and segments, can be picked using the respective topological level.

When a compound is selected at the Compound topological level, and then one of its elements is picked with topological level set to Element, the element is deselected. The other elements of the compound remain selected, but the compound itself is no longer selected. If parts of a compound are picked with topological level set to Element, and then the compound is picked with topological level set to Compound, all parts of that compound are selected.

A selected compound can be deselected in the usual manner, that is, by picking it again with topological level set to Compound.
Automatic

This is a context sensitive modifier that does not set a specific topological level, but determines the intended topological levels on the basis of where the mouse cursor is located.

- If the mouse is close to a point, the cursor changes to and that point is selected.
- If the mouse is close to a segment, the cursor changes to and that segment is selected.
- If the mouse is inside the boundary of a closed element, the cursor changes to and the element is selected. Likewise, if the mouse cursor is on or close to a segment of either an open or a closed element and the option (Macintosh) or ctrl+shift (Windows) key is pressed, that element is selected. If the element is a member of a compound and the control (MacOS) or ctrl+alt (Windows) key is pressed, the whole compound element is selected.

As in modeling, the Automatic level can be used with both the Pick tool and with any other tool.

Area

This modifier sets the current topological level to the Area element. The area is picked by clicking the mouse anywhere in the graphics window, provided the area element exists. It should be noted that picking the area and picking the boundary shape of the area are two semantically distinct operations. The boundary shape defines the extent of the area and can be treated as any other shape. That is, its points, its segments, and the complete shape can be picked and operated on as any other polyline. Picking the area itself, picks and applies operations to the drafting elements that are inside or outside its boundary, rather than to the boundary shape itself. These distinctions are illustrated in Figure 5.3.1.4.

Strictly speaking, the area is not a topological level in the structure of a drafting element in the same manner that the point and the segment are. However, it represents a mode that affects the pick tool in a manner similar to that of the real topological levels. Consequently, it is treated as one of the topological levels and its icon has been included with the topological level modifiers.

The area and its picking and clipping functions represent features that are only available in the drafting environment. They are discussed in the next subsection.

Figure 5.3.1.4: Picking (a) a point, (b) a segment, and (c) the complete bounding shape of an area. (d) Picking the area itself and its content.
5.3.2 The Pick tool and Pick Options dialog

Clicking with this tool on an entity selects it. The type of entity it picks is determined by the active topological level. If it does not find an entity to pick, it beeps and cancels the pick operation. Note that, contrary to the modeling environment, all entities can be picked with a single point. In the drafting environment, there is no topological level (such as the outline and face in modeling) that requires two selection points.

In the drafting environment, entities can also be picked indirectly, through the area element. That is, elements inside, outside, or crossing the area's boundary shape may be picked by picking the area. This can be done with both the prepick and postpick methods. When the Clip To Area option is selected, the area will also clip the selected elements to the boundary shape of the area. The area and its functions are discussed in detail in the next subsection.

Note that the clipping operation is applied at the time the area is picked. The implication of this is that, when the prepick method is used, the elements will be clipped even if no other operation is subsequently applied to them. Thus, in such cases, picking the area constitutes the complete operation. As with other operations, the clipping can be cancelled by executing Undo.

The Pick Options dialog, shown in Figure 5.3.2.1 is invoked directly from the Pick tool. The drafting version of the dialog is part of the modeling version. The options that appear in both actually refer to the same option. That is, if you turn one on in drafting, it is also on in modeling and vice versa.

Area picking

The area picking method in the modeling environment is also available in drafting. Recall that, when you press the mouse button at a position away from any pickable entity and drag it while holding the button down, you can draw a rectangular frame or a lasso shape. When the mouse is released, the entities inside the frame or the lasso shape are selected, according to the currently selected topological level.
Whether a rectangular frame or a lasso shape is drawn is controlled by an option set in the **Pick Options** dialog. As in modeling, the area picking by frame or lasso can only be used with the Pick tool and, consequently, it can only be used for prepicking.

Picking by frame or lasso (referred to as area picking) resembles picking with the area element, except that you cannot clip with the frame or lasso. In addition, the area element, which is drawn as any element, can be more precise.

**Unpicking**

Picked entities can be individually **unpicked**, by picking them again. In addition, all the currently picked entities may be unpicked by clicking the mouse anywhere on the window, away from any selectable entity. When this happens and the **Beep When Deselecting** option is on in the **Pick Options** dialog, a beep sounds. Turning the **Beep When Deselecting** option off deactivates the sound.

**Pick parade**

In some instances, multiple entities may overlap on the screen, and a single mouse click may not be enough to distinguish the desired selection. One solution to this problem is to zoom in on the overlapping area to see the elements at a larger scale. Often, elements that appear to overlap at a small scale, may not actually overlap, and zooming in will allow the selection of the desired entity.

When entities in fact do overlap, **form-Z** provides the **pick parade** facility. The pick parade sequentially highlights the possible selections for a given input point until the desired selection is highlighted. To use the pick parade, the **shift** key is pressed before the initial selection point. The first entity found is highlighted (for the point and segment topological levels, the element that the entity belongs to is highlighted in black). Each additional click of the mouse causes the parade to highlight the next overlapping entity. When the intended entity becomes highlighted, the **shift** key is released, and one additional click of the mouse is needed to end the parade and accept the current selection. Note that, if the **Use Shift Key For Multiple Pick** option is selected within the **Pick Options** dialog, the pick parade option becomes disabled.

**Prepicking and postpicking**

The drafting environment also offers a choice between a prepick and a postpick selection method when executing an operation. These methods work as in the modeling environment (see subsection 4.3.4). With the prepick method, the elements are picked first, the tool is activated next, and the mouse is clicked anywhere in the graphics window. With the postpick method, the elements are picked after a tool has been activated. The prepick method relies on the use of the Pick tool for selecting entities. With the postpick method, the picking procedure is embedded within each tool.
5.3.3 The area and its picking and clipping functions

An area element is generated by drawing a closed shape, while the Area modifier is on (subsection 5.2.1). The closed shape is referred to as the area's **boundary shape**. It is created using one or more of the drawing tools in rows two, three, and four of the tool bar. It can take any regular or irregular shape. Its boundary can be a polyline or an arc, but is required to be closed. Consequently, the partial arcs in row three cannot be used to draw an area. When an open polyline is drawn, it is automatically closed by inserting a segment that connects its first and last points. An area's boundary may intersect itself. Examples of areas are in Figure 5.3.3.1.

An area may or may not exist during a drafting session. When it does, there can only be a single area at a given time. Creating a new area replaces the previous, if one exists. The new area can be generated while the previous one is still displayed on the screen. As soon as a new area is drawn, the previous is automatically deleted. The area can also be explicitly deleted by deleting its boundary shape, while the topological level is at Element.

Once created, the area can be freely moved elsewhere within the drafting space, by moving its boundary shape. When created as a polyline, its shape can also be changed by moving individual points or segments. In general, any of the geometric transformations can be applied at the Point, Segment, or Element level to manipulate the shape and/or the position of an area. Under these operations, the boundary of the area behaves as any other polyline or arc. Note that applying an operation with topological level set at Element, is semantically different from applying the same operation with level set at Area (Figure 5.3.3.2).

The area is used to identify other groups of drafting elements on the basis of their position relative to its boundary shape. Moving the area, after picking it as an area (while the topological level is at Area), moves all the elements it has selected. All the geometric transformations work the same way and affect both the shapes that are selected by the area and its boundary shape. Other tools affect the shapes identified by the area only. For example, the Delete tool, when executed with topological level at Area, deletes all the elements selected by the area but not the boundary shape. When executed with topological level at Element, it deletes the boundary shape of the area, which deletes the area itself.

![Figure 5.3.3.1: Area generated as (a) hexagonal polyline, (b) mixed polyline that crosses itself, and (c) an ellipse arc.](image)

![Figure 5.3.3.2: Area is moved after picking it as (a) an element and (b) an area.](image)
The elements picked by the area are determined by its boundary shape and options in the **Area Options** dialog (Figure 5.3.3.3), invoked from the Area Generation and Area Pick tools.

**Boundary Type:** The area can pick elements that are completely and clearly inside or outside its boundary, or it can include elements that cross its boundary. The latter can be picked as complete shapes or they can first be *clipped* at the points where they intersect the boundary.

**In/Out Only:** When this option is selected (default), the area picks only elements that are completely inside (Figure 5.3.3.4 (1a)) or completely outside (Figure 5.3.3.4 (2a)) its boundary.

**On/Crossing:** With this option, the area picks elements that are touching or crossing its boundary, and elements completely inside (Figure 5.3.3.4 (1b)) or outside (Figure 5.3.3.4 (2b)).

**Clip to Area:** With this option the area first applies a clipping operation that breaks the elements at the points where they cross it. It then selects those portions of the clipped elements that are inside and all the elements that are completely inside (Figure 5.3.3.4 (1c)); or the clipped portions that are outside, as well as the elements that are completely outside (Figure 5.3.3.4 (2c)).

**Area Type:** The area subdivides the universal drafting space to two portions: that inside and that outside the boundary shape. It can thus be used to select either entities that are **Inside** (default) or those **Outside** its boundary. Depending on the option selected in the dialog.

*Figure 5.3.3.4:* Picking with Area element: (1) **Inside** and (2) **Outside**.  
(a) **In/Out Only**, (b) **On/Crossing**, and (c) **Clip To Area**.
As illustrated in Figure 5.3.3.5, the area element can be used to pick any type of a drafting element. The area cannot be picked when it does not pick any element. For example, if the **In/Out Only** and the **Inside** options are selected and no element is contained within its boundary shape (Figure 5.3.3.6(a)) or there are only elements that cross its boundary shape (Figure 5.3.3.6(b)), the area cannot be picked.

The clipping operation can only be applied to polylines. Consequently, when the **Clip To Area** option is selected and the area is used to clip elements other than polylines, these elements are first exploded (are transformed to polylines) before they are clipped. For example, when a dimension is clipped, it is transformed to a collection of line segments and numeric fields and it loses its associative character. The same is true for the leader lines. When an arc is clipped it is transformed to a polyline. Text elements and image elements cannot be exploded and, consequently, they cannot be clipped. When text is encountered by the clipping function of the area it is ignored.

**Figure 5.3.3.5:** The area can pick all types of drafting elements.

**Figure 5.3.3.6:** With **In/Out Only** and **Inside** on, (a) area contains no element, and (b) elements cross its boundary. In both cases it cannot be picked.
5.3.4 The Pick Pane tool

Panes are only available when the drafting environment is in the layout space mode, which is active when the Draft Layout Space item in the Display menu is selected. A drafting project can have any number of panes, only one is active. The following tool is used to make a pane active.

![Pick Pane]

Clicking with this tool within or on the boundary of a pane makes it the active pane. When a pane is activated, its border becomes gray. To deactivate a pane, click anywhere in the window outside the active pane. If the click activates another pane, the previously active pane becomes inactive. The content of the Layers palette is updated to reflect the change and its attributes are available for editing. All input by the view control tools and the element generation and editing tools is directed through the active pane and into the draft space.

A pane needs to be active in order to be able to operate on the elements it displays or to generate new elements. All such operations are made to the master drawing in the drafting environment and are also displayed in all the other panes, even though they are not active. On the other hand, all view manipulation operations, such as zooming, panning, and fitting only affect the active pane and none of the inactive panes. If such operations are applied while no pane is active, they are applied to the layout window itself.

While the Activate Pane tool is used to make a pane and its content active, the Pick tool can still be used to pick the complete boundary shape of a pane, one or more of its segments, or points. That is, with respect to the Pick tool, the pane is affected by the topological level modifiers. Once picked, a pane or parts of it can be moved or otherwise transformed to either position it elsewhere in the layout space or to change its shape. In addition, a pane or parts of it can be picked to be transformed directly by one of the geometric transformation tools.
5.4 Derivative enclosures and parallel elements

This section discusses two tools:

- Derive Enclosure
- Parallel Offset

Both of these tools generate elements from other elements by drawing parallel lines. The former simulates double line “walls”, as they are typically used in architectural floor plans. It completes the derivation of these wall elements in one step, including cleaning the intersections of the parallel segments. The latter tool is more iterative in nature and derives a parallel line on only one side of an element. However, the operation can be repeated to also derive a parallel line on the other side of the original shape. The Parallel Offset tool in general is more manual and at the same time more flexible than the Enclosure.
5.4.1 Derivative enclosures

In addition to those drawn directly, enclosures can also be derived from other previously drawn elements.

Enclosure

This tool can be used to derive an enclosure from another shape. It can be applied using either the postpick or prepick method, in the usual manner. Derivative enclosures can be generated from segments, complete elements, or complete compounds. They can also be derived from elements that are members of compounds. The level at which they are applied is determined by the active topological level. If the element selected has a dimension, it is transferred to the enclosure after it is adjusted to points of the right parallel line which correspond to the points referenced by the original dimension.

The options for the Enclosure tool are the same as for the Enclosure modifier as shown in Figure 5.1.1.3. When derivative enclosures are generated from elements and compounds, these elements are deleted after the enclosure is derived. When an enclosure is derived from a segment, the element whose segment is used remains as is. Examples of derivative enclosures are shown in Figure 5.4.1.1.

Note that an enclosure derived from an open shape is a closed element. An enclosure derived from a closed shape consists of two closed shapes which form a compound element (see next section).

Figure 5.4.1.1: Enclosures generated from (1) an open polyline, (2) a rectangle, and (3) a compound; at Topological Level (a) Compound, (b) Element, and (c) Segment.
5.4.2 Parallel offset

This tool applies to polylines (including the rectangles), leader lines, and arcs. Given a selected element, it creates a new element that is parallel to the selected element. When applied to a polyline, the new polyline is constructed by generating a parallel segment for each segment of the operand, and by calculating the points at which the parallel segments intersect, taken in their original order. When applied to an arc, the new arc is generated using the same parameters as the original arc, except for the radius, which is adjusted to the size corresponding to the position of the parallel arc. The generation of a parallel polyline and arc are shown in Figure 5.4.2.1.

Polylines that are generated at a relatively small distance from the operand polyline, generally have a shape that is similar to that of the original polyline. When the distance is beyond a certain threshold, one parallel segment may jump to the other side of its neighboring segment, which leads to a distortion of the original shape. Concentric shapes such as polygons may be completely reflected, relative to their centers. Such cases are illustrated in Figure 5.4.2.2.

The parallel elements are generated at a distance from the original element, which is determined by the Relative or Absolute position of the mouse, or by a preset Numeric value representing the distance. These options are selected from the Parallel Options dialog, shown in Figure 5.4.2.3, that is invoked directly from the Parallel Offset tool.
When the Parallel Offset tool is executed in the **Absolute** mode, it constructs a parallel element at the position of the mouse. When executed in the **Relative** mode, it constructs a parallel element at a distance determined by the distance traveled by the mouse. The first mouse click is used as a **base** point that controls two factors. First, it determines the side of the element where the parallel element will be constructed. Second, it is used as the reference point to determine whether the distance traveled by the mouse is a positive or a negative value. If the mouse travels to the right of the base point, the distance is **positive**; if it travels to the left the distance is **negative**. Note that when the distance is negative, the parallel line will be created on the side opposite from the one selected by the base point.

By definition, the Parallel Offset tool works at the Element level only. It can be applied by using the prepick or the postpick method. With the postpick method, the Parallel Offset tool is activated first, then the entity is picked. A parallel element is rubber banded, as soon as the operand is picked, and follows the motion of the mouse. That motion is interpreted differently for the **Absolute** and the **Relative** modes. One more click of the mouse completes the execution of the operation and generates a parallel element. With the postpick method, the topological level is ignored and the operation is always executed at the Element level. Examples of parallel elements generated at the position of the mouse (**Absolute** mode) are shown in Figure 5.4.2.4(a).

With the prepick method, any number of elements may be picked first, followed by the selection of the Parallel Offset tool. Then the mouse is clicked twice. The first click generates a set of parallel elements, corresponding to the prepicked operands, and initiates the rubber banding process. The second mouse click completes the execution of the operation. Note that, in the **Absolute** mode, all the parallel elements are generated at the position of the mouse (see Figure 5.4.2.4(b)). In the **Relative** mode, the distance traveled by the mouse is applied in deriving the parallel elements. When the prepicked elements include entities that were picked at a level other than Element, they are ignored.

![Figure 5.4.2.4](image-url)

*Figure 5.4.2.4:* Parallel elements are generated at the position of the mouse (**Absolute** mode).
- (a) The postpick method was used and the parallel elements were generated one at a time.
- (b) After prepicking, parallel elements were generated simultaneously at the same mouse position.
When the Parallel Offset tool is used in the **Numeric** mode, it works similarly to the **Relative** mode, but uses a preset distance to construct the new parallel element. That distance is controlled by the user and is set through the **Linear Distance** field of the **Parallel Options** dialog. The default value for the distance is 2'-0" (or 0.50 m in metric).

When the postpick method is used, the selection of the Parallel Offset tool is followed by a single click of the mouse, which picks the element and executes the operation immediately. No rubber banding occurs. The position where the mouse is clicked is significant and determines the side of the element where the parallel element will be generated. That is, to pick the element, the mouse is clicked sufficiently close to it, but on either one of its two sides. Examples are shown in Figure 5.4.2.5(a). Where the mouse is clicked determines the side where the parallel element is generated. If a negative value has been entered in the **Linear Distance** field, then the parallel element is created on the side opposite from the one selected. With the postpick method, the topological level is ignored and the Parallel Offset tool is always executed at the Element level.

When the prepick method is used, any number of elements may be picked, followed by the selection of the Parallel Offset tool, followed by a single mouse click, which determines the sides on which the parallel elements will be generated and triggers the execution of the operation. The position of the mouse click is compared with each of the operands independently, which causes the parallel elements to be generated on different sides of the different elements (see Figure 5.4.2.5(b)). With the prepick method, all the entities should be picked at the Element level. Those picked at other levels, if any, are ignored by the Parallel Offset tool.

![Figure 5.4.2.5: Parallel elements are generated at a preset distance (Numeric mode). (a) The postpick method was used and the parallel elements were generated one at a time. (b) The prepick method was used and a single click determined where the parallel elements were generated.](image-url)
5.5 Boolean and compound operations

This section covers the following tools:

- Union
- Intersection
- Difference
- B-Split
- Compound

The first four of these tools are Boolean operations, which work as in modeling for 2D shapes. The last tool is similar to the Join or Group tool in modeling. It links two elements into a composite element, which is called compound.

The Compound tool is discussed in the same section with the Booleans for a number of reasons. It can be viewed as a weaker Boolean. Actually, under certain conditions results derived with the Union and Difference operations can also be derived with the Compound tool. The result of the Booleans is frequently a compound element, which is always the case with the result of the Compound tool. Finally, the Compound tool is frequently useful for preparing elements before executing a Boolean.
5.5.1 Boolean operations

The Boolean operations can only be applied to closed shapes, at either the Element or the Compound level. Unlike the modeling surface objects which may contain holes as parts of their overall structure, in drafting, shapes that are inside other shapes must be parts of the same compound element to be processed as holes. Such elements should first be composed into a compound (using the Compound tool) before applying a Boolean operation to them. Which shapes are processed as holes depends on the overall spatial relationship of the shapes contained in the compound. This is illustrated in Figure 5.5.1.1.

These are the “plain” Boolean operations which are by definition binary. While they are usually executed in postpick mode by clicking on two elements or compounds with a Boolean tool, they can also be executed in prepick mode as in modeling.

When applying a Boolean operation using the postpick method, the topological level is ignored and the Boolean is applied at the Compound level. The Boolean operations can also be applied at the Element level, but only with elements that are not members of compounds. When the result of a Boolean consists of only one part, a simple element is generated. When it consists of more than one part, then a compound element is generated. Examples are shown in Figure 5.5.1.2.

Boolean Split

This is a multi-Boolean operation, which uses one element to split another. It can be executed One Way or Two Way, which is selected in the Boolean Split Options dialog (Figure 5.5.1.3) that is invoked from the tool. The Two Way option causes both elements to split each other. The B-Split operation is a combination of a Difference and an Intersection.
5.5.2 Compound operation

A compound allows you to group or link other previously created elements, including other compounds. Thus a compound may consist of compositions at a number of hierarchical levels. Compound elements can be transformed and manipulated as single elements by setting the topological level to Compound (see subsection 5.3.1). At the same time, the elements it consists of and their parts can also be manipulated using lower Topological Levels. Compounds are restricted from an operation when one or more of their component parts are restricted from that operation. Otherwise, operations can be applied to compounds as with any other element.

While, for all practical purposes, any individual element may be considered a compound that consists of a single element, at least two elements are required to be picked in order to construct a compound. Compound elements can be created by using the following tool.

![Compound]

This tool may be applied using either the prepick or the postpick method. To use the postpick method, activate the Compound tool and, with topological level set to Compound, click on two elements. The operation is executed immediately and the two picked elements are composed into a compound element. To use the prepick method, activate the Pick tool and select two or more elements or compounds, with the topological level set to Element or Compound, respectively. Then activate the Compound tool and click in the graphics window. The preselected elements are composed at the same hierarchical level and become a compound element. When only one element is preselected, a second should be selected after the Compound tool is activated. A compound cannot be created from a single element. Note that when using the postpick method, at most two elements can be composed. The prepick method allows you to compose any number of elements.

Compounds can be composed of any type of drafting element, namely delineators (polylines, rectangles, circles, ellipses, and arcs), leader lines, dimensions, text elements, and symbols. Elements that are already parts of a compound cannot be composed into another compound. When encountered, such elements are skipped and an error message is posted. Hatch patterns, dimensions, and leader lines which are associated with elements that are composed into a compound are retained.

The Explode tool can be used with compounds and each time it is applied it decomposes the highest level. A compound with compositions at multiple hierarchical levels will require more than one application of the Explode operation to be completely dismantled into individual elements. When using the Explode tool with compound elements, the topological level should be set to Compound. The Explode tool is discussed in section 5.9.
It is also possible to explode an element that is part of a compound, by using the Explode tool with topological level set to Element. The parts to which the element is exploded become parts of the compound. For example, if a polyline contained in a compound is exploded into its segments, these segments remain in the compound.

The compound elements are affected by the Self/Copy modifiers, like any other element, when a geometric transformation is applied to them. In the Self mode, the compound is transformed by transforming all its parts. In any of the copy modes, new copies of the entire compound are created by copying all its parts.

If the Join or Connect tools are applied to elements contained in compounds, they are executed, but the resulting elements are removed from the compounds and become individual elements.

Compounds, or parts of compounds, can be hatched, dimensioned, or have a leader line associated with them. When a compound is hatched at the Compound level, all its parts are hatched as if they were selected individually at the Element level. When the compound is exploded, the hatch remains as is.

The compound elements have no attributes associated with them, but their member elements do. They retain the attributes they had at the time they were composed. These attributes can be subsequently changed by setting the topological level to either Element or Compound. In the former case the attributes of only one element are changed. In the latter case the attributes of all the elements in a compound are changed. Examples are shown in Figure 5.5.2.1.

![Figure 5.5.2.1](image)

*Figure 5.5.2.1:* Compounds consisting of (a) rectangles, a dimension, and a leader line; (b) concentric polygons; (c) rectangles, circles, and open lines. Applying hatching at the Compound level to (d) intersecting and (e) non-intersecting rectangles. (f) Applying hatching at the Element level to the four rectangles contained in a compound.
5.6 Line editing

This section covers the following tools:

- Break Line
- Break with Line
- Trim Open
- Trim Join
- Trim Fit Fillet
- Trim Bevel
- Trim Lines
- Trim with Line

These tools are used to edit drafting elements. They work at the Point, Segment, and Element level, but not at the Area level. The editing tools can be used to restructure the topology of previously created drafting elements. A number of polylines can be joined into one, or one polyline can be broken into a number of smaller polylines. Open polylines can be closed, or closed polylines may be opened. In addition, the shapes of polylines can be changed by trimming them or through fillet fitting.

These tools often work better with the postpick rather than the prepick selection method. This is because they typically rely on the relationship between a specific selection point and the entity selected. When the postpick method is used, entities are picked until the required number of operands is selected, then the operation is applied or dynamic rubber banding action begins. Some of these tools require two entities to be picked; others require only one selection.

When the prepick method is used, any number of entities are selected, then the desired editing tool is activated, followed by a mouse click on the project window, which initiates the execution of the operation. When a line edit tool requires a pair of entities for the operation, an even number of entities should be prepicked. If an odd number of entities is prepicked, the last one is ignored.
5.6.1 Breaking lines

Two tools for breaking lines are available. They share the same dialog and options.

**Break Line**

This tool can be used to break an element at a segment or point. Options for the execution of this tool can be selected in the **Break Line Options** dialog (Figure 5.6.1.1). This dialog is the same with its modeling counterpart (subsection 4.21.1) except for the **Break With Line** variation, which in drafting is not an option in the dialog but a separate tool (see below).

In postpick mode, the Break Line tool works as in modeling (see Figure 5.6.1.2). With the tool active you click on a line, which is split at the click point. If the click is close to a point, the line is split at the point. If the **Break Distance** option is on, the break points are also offset.

The Break Line tool can also be used in prepick mode. Any number of elements can be preselected with the Pick tool. Then, with the Break tool active you click close to the preselected elements. The break points are determined by drawing perpendicular lines to each of the preselected lines. Also, if the click point is close enough to a point, the respective element is broken at that point. Breaking by prepicking is illustrated in Figure 5.6.1.3.
Break with Line

This tool is used to break one or more lines at the point of intersection with another line, rather than at a click point, which the Break tool does. It can be executed in either prepick or postpick mode and it is affected by the same options the Break tool is. That is, the **Break Line Options** dialog (Figure 5.6.1.1) can also be invoked from this tool.

Using the postpick method, with the Break with Line tool active click on the line to be broken, and then on the line with which it will be broken. If the two lines intersect, the first line is broken at their point of intersection. If the **Break Distance** option is on, the break is clearly visible. This is illustrated in Figure 5.6.1.4.

When using the prepick method, many lines can be broken with a single line, provided they are intersected with that line. With the Pick tool, prepick all the lines you wish to be broken. Then activate the Break with Line tool and click on the line the previous lines will be broken with. This is illustrated in Figure 5.6.1.5. Note that, when using the Break with Line tool, the click is only significant for selecting the lines. The exact position of the click is insignificant.
5.6.2 Trimming segments

The four tools discussed in this subsection all start by trimming two segments to their point of intersection. They vary in that one leaves the trim points open, another joins them, and two others apply a fillet or bevel at the trim point.

**Trim Open**

This tool trims two segments at their point of intersection. The operation is actually executed by shrinking or extending each of the two segments to their common point of intersection. Whenever the points of the segments that are affected by the operations are shared by other segments in the polyline, the adjacent segments are also affected.

Where the mouse is clicked, when the segments are picked for trimming, is significant. It determines which portions of the segments will be trimmed. For example, as illustrated in Figure 5.6.2.1(a), when the two line segments intersect, there are four different ways in which they can be picked, each producing a different result. If only one of the segments crosses over the other segment, there are two distinct ways in which they can be picked (Figure 5.6.2.1(b)).

When the two segments do not intersect, where the click is has no significance. The two segments can only extend in the direction of their point of intersection (Figure 5.6.2.1(c)). If attempt is made to apply the Trim Open tool to two parallel segments, an error message is posted (Figure 5.6.2.1(d)). When the trimming is successful, the two segments appear to be continuous at their point of intersection. However, they remain open or disconnected and can be pulled apart by applying a geometric transformation.

Applying Trim Open to a segment of a polyline consisting of more segments also affects the shape of the segment to which the selected segment is connected. When a segment is trimmed, it does not lose its connection to the segment next to it. Thus, as a segment is extended or shrunk, it also pulls the neighboring segment to that position, as shown in Figure 5.6.2.2.

**Figure 5.6.2.1:** Trimming (a) crossing segments, (b) segments only one of which crosses the other, (c) non crossing, and (d) parallel segments.

**Figure 5.6.2.2:** Trimming segments of polylines that consist of more than one segment: (a) open polylines and (b) closed polylines.
Both the postpick and the prepick methods can be used with the Trim Open tool. When the postpick method is used, the Trim Open tool is always executed at the Segment level. If the topological level is set to a level other than Segment, it is ignored. To execute the Trim Open tool, activate the tool and click on two segments. The operation is executed immediately.

When the prepick method is used, any number of segments are selected with the Pick tool, then the Trim Open tool is selected and the mouse is clicked anywhere in the graphics window, which initiates the execution of the operation. The Trim Open tool is, by definition, binary. When more than two entities are prepicked, they are taken two at a time, in the order they were picked. If the number of picked segments is odd, the last segment is ignored. With the prepick method, the Trim Open tool only accepts entities picked at the Segment level. Entities picked at another level are ignored.

Trim Join

This tool works exactly as Trim Open does with the addition that the trimmed segments are also joined, when they are free (endpoints). When they are connected to other segments the result is exactly the same with that produced by Trim Open, except that a warning message is also issued. The examples in 5.6.2.1 and 5.6.2.2 apply equally well to the Trim Join tool.
Trim Fit Fillet

This tool is also executed exactly as the Trim Open tool, but with an additional function. It executes a trim, but also fits a circular fillet at the corner of intersection of two segments provided that the segments meet at an angle less than 180°. The fillet is an arc created tangent to the segments. How the fillet is calculated and other parameters are set in the Trim Fit Fillet Options dialog (Figure 5.6.2.3) invoked from this tool.

Fillet Method: Two methods are available for calculating the fillet.

Use Radius: With this option the fillet is derived from the radius value set in its field.

Use Distance: With this option the fillet is derived by placing its point of tangency at the specified distance from the trim point.

Parametric Fillet: When this option is selected, the fillet is generated as a parametric arc, which is a separate element.

Polygonal Fillet: When this option is selected, the fillet is generated as a polyline. The number of segments used to generate the fillet is set in the # Of Edges field. The polygonal fillet may or may not be joined to the trim segments, depending on whether Join is on or off. The fillet can only be joined when the trim segments have open ends.

The examples in Figures 5.6.2.4 and 5.6.2.5 repeat the examples in Figure 5.6.2.1 and 5.6.2.2 that were shown for the Trim Open tool. As shown, where you click when selecting the segments to which a fillet will be fitted is significant. It determines the parts of the segments where the fillet will be applied.

When the trimmed segments are not open, they retain the connections they had before the application of the Trim Fit Fillet tool, but their shapes change (see Figure 5.6.2.5).
Trim Bevel

This tool is similar and is executed as the Trim Fit Fillet tool, but applies a bevel instead of a fillet. The length of the bevel or its offset are set in the **Bevel Options** dialog (Figure 5.6.2.6), invoked from this tool.

![Figure 5.6.2.6: The Bevel Options dialog.](image)

**Offset**: When this option is selected, the bevel, is generated by applying the offsets specified in the **1st** and **2nd** numeric fields. The segments at the point of the bevel are shortened by the lengths entered in the fields, and then connected by a bevel segment. If the lengths entered are not equal, an asymmetrical bevel is created.

**Length**: When this option is selected, a symmetrical bevel of the specified length is created. By how much the segments are shortened depends on their angle and the length of the bevel.

For both options, if the trim segments are too short to fit the bevel, an error message is issued and the operation is not executed.
5.6.3 Trimming lines

The tools in this subsection resemble the previous in that they also apply trimming. However, while the previous were applied at the Segment level, these are applied at the Element level.

Trim Lines

When this tool is applied to two or more polylines, it trims the last segment of the first polyline to the first segment of the next, and so on until the list of picked polylines is exhausted. When using the postpick selection method, this tool can only be applied to at most two polylines. When using the prepick method any number of open polylines can be selected. Then clicking anywhere in the project window with the Trim Lines tool executes the operation.

How the trim points are treated depends on options set in the Trim Lines Options dialog (Figure 5.6.3.1) invoked from this tool.

Open: Selecting this option leaves the trimmed end points of the lines open or disconnected.

Join: Selecting this option causes the pairs of trimmed endpoints to also be joined, resulting in a single polyline produced from the trimmed polylines.

Close Sequence: This option causes formZ to also trim the last segment of the last line and the first segment of the first line.

Trim with Line

Clicking with this tool on two polylines trims the end segments of the first to their point of intersection with the second. This is illustrated in Figure 5.6.3.2. When using the prepick method, any number of open polylines is selected with the Pick tool and then, with the Trim with Line tool active you click on the trim line. All preselected lines are trimmed to their point of intersection with the last line picked. Examples are shown in Figure 5.6.3.2, 5.6.3.3, and 5.6.3.4. The examples in 5.6.3.2 and 5.6.3.3 also illustrate the significance of the position of the click.
5.6.4 Joining

![Join Lines](image)

This tool is used to join the ends of two or more open lines together, provided these ends are within a tolerance value. There are two variations of this operation, which are selected from the **Join Lines Options** dialog (Figure 5.6.4.1) invoked from this tool.

**Join**: This operation joins the ends of specifically selected polylines, in the order they are selected. It may be used in postpick or prepick mode.

When using the postpick selection method, with the Join Lines tool active you click on two open polylines. They are joined immediately if they have a pair of endpoints that are close enough (within the **Tolerance** distance). That is, the program compares the distance of each of the endpoints of the first polyline with each of the endpoints of the second polyline and joins the pair of points that are the closest. When the points are not exactly on top of each other, one has to be moved to be able to join them. The program moves the point of the polyline selected first. Consequently, the order in which the polylines are selected affects the final shape as illustrated in Figure 5.6.4.2.

---

**Figure 5.6.4.1**: The **Join Lines Options** dialog.

**Figure 5.6.4.2**: Join Lines with **Join** on: Where we click is significant:

- **Close Line Sequence** (a,b) off and (c) on.
**Join All:** This variation of the Join Lines tool joins lines as **Join** does, however, its spectrum is broader and it is executed differently. While this option is on and the Join Lines tool is active, click anywhere in the project window. The program acts as if all the elements in the project have been preselected, but not in any particular order. It checks the distances of each endpoint of each open polyline with all the other endpoints and from those that are within the tolerance progressively selects the pair with the smallest distance. These endpoints are joined and the program searches for the next pair, until all the possibilities have been exhausted. This operation is useful when data comes in from other programs segmented and you need to have it as continuous shapes. An example is shown in Figure 5.6.4.3.

Note that in the example the joined lines appear to have moved. This is because one of the matching points has to move in order to become coincident with the other point before the two can be joined. Our example does exaggerate the condition. Real life data is expected to be more accurate than this and its points already coincident, which will not result in motions.

**Visible Layers Only:** This option is only available when the **Join All** variation is on. When on, the program will only process the open polylines on the visible layers. This allows you significant control over which lines to be joined by placing them on different layers.

**Close Line Sequence:** This option affects both the **Join** and **Join All** variations. When on, it also closes either the sequence picked for the former or that detected for the latter operation. The sequence is closed by joining the two open ends, provided they are within the **Tolerance** value. The examples in Figure 5.6.4.2(c) and 5.6.4.3 were executed with this option on.
5.6.5 Closing open lines

When clicking with this tool on an open polyline, it is closed using one of three methods that are selected in the Close Line Options dialog (Figure 5.6.5.1), invoked from the tool.

Trim: Selecting this option will close the element by trimming the ends to their common point of intersection, providing such a point exists. When the end segments of a polyline are parallel, they cannot be trimmed. Figure 5.6.5.2(a) shows an open shape that is closed using the Trim option. Note that, with this tool, it is not important which segment you click on to select the shape.

Join: When this option is on, one of the endpoints of an open polyline is moved to become coincident with the other endpoint and the two are joined into one. The point of the end segment on which you click to execute the operation is the one that is moved. If you do not click on an end segment, the program will pick one. Consequently, where you click when executing Close Line with Join on is significant, as illustrated in Figure 5.6.5.2 (b) and (c). Note that this option does not depend on a tolerance value and the ends of an open line will always be joined no matter how far apart they are.

Connect: When this option is on the ends of an open line are closed by generating a new segment between them. This is illustrated in Figure 5.6.5.2(d).

There are certain restrictions relative to the minimum number of segments required in a polyline to be able to close it. At least three segments are needed to be able to close it with the Trim or Join method. At least two segments are required to be able to close it with the Connect method.
5.6.6 Connecting segments and lines

“Connect” is the term used when two free endpoints are linked through the generation of a new segment. Two connect tools are discussed below: one is applied at the Segment level and the other at the Element level.

Connect Segments

Clicking with this tool on two segments with open ends generates a new segment between the two open ends and connects them. If one of the segments you click on does not have an open end, an error message is posted and the operation is not executed.

The selected open segments can be on different polylines or on the same polyline. In the former case, the two polylines become one. In the latter case an open polyline becomes closed. When two single segment polylines are selected, that is entities with both ends free, then where you click to select them determines which ends will be connected. Examples are shown in Figure 5.6.6.1.

This tool can be applied using either the postpick or prepick method and is always applied at the Segment topological level. When using the postpick method, with the Connect Segments tool active you click on two segments. The operation ignores the current topological level and is always executed at the Segment level. When using the prepick method with topological level set to Segment you use the Pick tool to select an even number of entities. Then with the Connect Segment tool active you click in the project window. The prepicked segments are connected in pairs, in the order they were selected. If an odd number of segments were preselected, the last selection is ignored.

![Figure 5.6.6.1: Connecting (a) crossing segments, and (b) partially crossing segments. Connecting a pair of (c) arbitrarily oriented and (d) parallel segments to derive closed shapes.](image-url)
Connect Lines

This tool connects the ends of two or more open lines by generating a new segment between them. When using the postpick method two lines are connected at a time. With the prepick method any number of lines can be preselected and connected in the order they were picked. A single option affects this operation and is set in the Connect Lines Options dialog (Figure 5.6.6.2) that is invoked from this tool.

**Close Line Sequence**: Selecting this option causes form•Z to take the first and last element of a sequence of elements and connects them with a segment. If this option is not selected, form•Z will leave the last endpoints unconnected.

Examples are shown in Figures 5.6.6.3 and 5.6.6.4. Note that in 5.6.6.3(b) the ends that are closer are connected. In 5.6.6.4, the prepick method is used, and, again, the smallest distance between end points determines which ones are connected at each step.
5.6.7 Filleting and beveling

Tools are available that allow you to round or bevel corners. They work in very similar fashions.

**Fit Fillet**

When you click with this tool on a point, segment, or element, it inserts a fillet to the points contained in the entity. The radius and type of fillet inserted is determined by the options in the **Fit Fillet Options** dialog (Figure 5.6.7.1) invoked from this tool. The options are the same as those in the **Trim Fit Fillet Options** dialog except for **Make Compound**.

When **Polygonal Fillet** is used, all the fillets generated are inserted and connected to the rest of the polyline. When **Parametric Fillet** is used, the fillets are elements separate from the segments between which they are inserted. However, the following option can be used to structure them as a single compound element.

**Make Compound**: This option is only available when the **Parametric Fillet** option is on. When it is selected, the element and the arc fillet are combined into a compound. Note that with the parametric fillet, the element is broken at the point of the fillet insertion. Depending on which topological level is used during the operation, a closed element becomes an open element, an open element is split, or the element is effectively exploded into single segment lines.

Figure 5.6.7.2 illustrates how fillets can be generated at different topological levels. In (a), the Point topological level is selected and the tool is clicked on the shown points. In (b), the Segment topological level is selected and the tool is clicked on the shown points. Notice that if a segment shares vertices with two other segments, fillets will be placed at both vertices. In (c), the Element topological level is selected. Note that when using the Element topological level with this tool, it is not significant where you click on the element.

**Figure 5.6.7.1**: The **Fit Fillet Options** dialog.

**Figure 5.6.7.2**: Using Fit Fillet on element with Topological Level at (a) Point, (b) Segment, (c) Element.
Bevel

This tool works as the Fit Fillet tool does, except that it applies a bevel instead of a fillet. The **Bevel Options** dialog (Figure 5.6.7.3) invoked by the tool contains the same options as the Trim Bevel tool. Bevels are inserted and connected to a polyline, which remains a single entity. Rectangles, however, are turned into closed polylines.

Figure 5.6.7.4 shows the same example as Figure 5.6.7.2, but executed with the Bevel tool.

**Figure 5.6.7.3:** The **Bevel Options** dialog.

**Figure 5.6.7.4:** Using the Bevel tool on an element with the Topological Level set to (a) Point, (b) Segment, and (c) Element.
5.6.8 Inserting points

Insert Point

Clicking on an entity with this tool inserts a new point at the position of the click. This tool can be executed with the topological level set to Point, Segment, or Element. It results in the insertion of a point on a point, the insertion of a point on a segment, or the insertion of a point on an element.

The insertion of a point on a point essentially generates a copy of the existing point and connects it to the original point as well as to its next point. The insertion of a point on a segment generates a new point between the two endpoints of a segment. This effectively subdivides the original segment into two segments. Inserting a point on an element inserts it on one of its segments.

Note that the insertion of a point does not produce any immediately visible result. The inserted point becomes visible when it is moved. In the examples in the Figures of this subsection the inserted points have been moved to show how the insertion was executed.

While postpick is the most direct way of executing the Insert Point tool, the prepick method is also available. With postpick, the tool is selected first, followed by a single mouse click. The mouse click identifies (picks) the entity where the point will be inserted. When the topological level is at Segment or Element, the position where the mouse is clicked also indicates where the point will be inserted. If the system is unable to identify an entity at the currently active topological level, it cancels the operation. Point insertion on points, segments, and elements are shown in Figure 5.6.8.1.

With the prepick method, any number of entities can be picked first, then the Insert Point tool is selected, followed by a single mouse click. The location of the cursor where the mouse is clicked determines the positions of the insertions. A line is drawn from the mouse location perpendicular to the segment, and their point of intersection is the position where the new point is inserted. If the point of intersection is beyond either end of the segment, then no insertion is made. Multiple insertions of the same point, using the prepick method, are illustrated in Figure 5.6.8.2.

![Figure 5.6.8.1: Inserting a point on (a) a point and (b) a segment. (Inserted points have been moved to show result.)](image1)

![Figure 5.6.8.2: Simultaneous insertion of points to prepicked entities: (a) crossing segments, (b) parallel segments, (c) parallel sides of two polygons, (d) point of one and segment of another polyline, and (e) on segments of three polylines.](image2)
5.7 Text

This section discusses three tools:

- Text Place
- Text Edit
- Text Search And Replace

It also discusses the options available for displaying text in drafting.

Text in drafting is similar to modeling’s plain text and can be placed through two methods: at point and between points. In addition to the TrueType and PostScript text, which are available in both modeling and drafting, there is one more type of text in drafting: bitmap text.

There are more display options for text in drafting than there are in modeling. This is necessary in order to control refresh speeds, as drawings grow in complexity and size. In addition to the bounding box approximation of text, a stick font is also available. This is a resident font that does not require time to be loaded and is simple enough to be drawn quickly. At the same time it is a good approximation of the real text it temporarily substitutes.
5.7.1 Generating and editing text

Text can be placed anywhere in a drafting window using the Text Place tool. After it has been placed it can be changed using the Text Edit tool.

Text Place

The Text Place tool can be used to place text using one of two methods: Text At Point or Text Between Points. The desired method is selected from the Text Place Options dialog, shown in Figure 5.7.1.1, that can be invoked directly from the Text Place tool.

Text is placed by selecting the Text Place tool, then clicking the mouse in the drafting window once (Text At Point) or twice (Text Between Points). After the one or two clicks, the Drafting Text Editor dialog (Figure 5.7.1.2) is invoked. As in modeling, it is used to enter the text and to set a variety of parameters that will determine the form of the text. It is similar to its modeling counterpart and its parametric options work in a similar fashion. However, there are a few differences.

The font menu contains TrueType, PostScript, and Bitmap (Windows only) fonts. The font type, TrueType, PostScript, of Bitmap, is displayed to the right of the font menu for informational purposes. Please note that OpenType fonts are also supported, however the font type is displayed as a TrueType font. The String Angle parameter only applies to Text At Point placement and its value determines the angle at which the text string will be generated. It is dimmed when Text Between Points is used. In the latter case the angle of the text is determined graphically by the orientation of the line that connects the two input points.

Text Edit

Previously created text can be changed. With the Text Edit tool active, click on a text string to invoke the Drafting Text Editor dialog where any of the parameters of the text can be changed. The text is regenerated and redrawn as soon as you click on OK to close the dialog.
5.7.2 Searching and replacing text

Text Search and Replace

After a text string has been generated as an object, it can be edited and changed. One of the ways text strings can be changed is by searching the string and replacing matching text with new text.

The Text Search And Replace tool searches the text strings of all selected objects (or all objects) for text matching the Search String. All instances of this text can then be replaced with the Replacement String. This tool can also be used to search and change the formatting of text elements for font, size, and style.

To search and replace text strings in previously created text elements, select the Text Search And Replace tool and click on any of the characters of the text element. Or prepick any number of text elements, select the Text Search And Replace tool, then click anywhere in the modeling window.

If Search All Objects is on, then just click in the drafting window and all text elements will be searched. No elements need to be picked in this case. The Text Search And Replace dialog is shown in Figure 5.7.2.1. In it you can select which of the text string or formatting parameters you wish to replace and the string and formatting you wish to replace it with. The options in the dialog are as follows:
Search: This box of options contains the specifications of the string to be searched.
  Font: The font to search for. The content of the Font menu depends on the fonts you have loaded on your machine.
  Size: The font size to search for.
  Style: The font style to search for. The content of this menu depends on the selected font. This option is only available if a font is selected.
  Search String: The string to search for.
  Ignore Case: When this option is selected, the case of the strings does not need match the case of the search string in order to be replaced. If this option is not selected, the case must match.
  Match Whole Word Only: When this option is selected, the search string must not be part of a larger word in the element’s text to be considered a match.

Search All Elements: When this option is selected, all elements are searched. If it is not selected, only picked elements are searched. This item is only available if objects are picked. Otherwise, it is disabled and selected.

Replace: This box of options contains the specifications of the replacement string.
  Font: The font to assign to the matching text.
  Size: The font size to assign to the matching text.
  Style: The font style to assign to the matching text. The content of this menu depends on the selected font. This option is only available if a font is selected.

  Replacement String: The text to replace the matching text with.
  Found Text For: Displays the element’s text with the current selection highlighted. Pressing the replace button will replace the highlighted text with the replacement string.

After the proper selections have been made in the Search and Replace boxes, the following buttons execute the desired operation.

Find: Clicking on this button finds the next occurrence of text that matches the selected search parameters.

Replace: Clicking on this button replaces the currently found occurrence of text that matches the selected search parameters with the selected replace parameters.

Restart: Clicking on this button restarts the search from the beginning.

Replace All: Clicking on this button silently replaces all occurrences of text that match the selected search parameters with the selected replace parameters.

Following are the operations that can be executed with the Text Search And Replace tool:
Replacing the entire text of all selected elements with a new string:

- In the **Replace** box, select **Replacement String** and enter the replacement string.
- In the **Search** box, make sure that **Font**, **Style**, **Size**, and **Search String** are off.
- Click the **Replace All** button.
The entire text of the selected elements will be replaced with the new string.

Replacing all instances of a search string in the text of all selected elements with a new string:

- In the **Search** box, select **Search String** and enter the search string.
- In the **Replace** box, select **Replacement String** and enter the new string.
- Click on the **Find** button.
Each instance of the **Search String** will be displayed in the **Found Text** text area.
- Click on the **Replace** button to perform the replacement.
- Click on the **Find** button to find the next instance of the **Search String**.
This method provides a review and selective replacement of the text. If desired, all instances of the **Search String** can be replaced without review, as follows:
- Select the **Replace All** button.
Instances of the search string will be replaced with the new string.

Changing the font, size, and/or style within the text of all selected elements:

- In the **Replace** box, select **Font**, **Size**, and/or **Style** and select the new font, size or style.
- In the **Search** box, make sure that **Font**, **Style**, **Size**, and **Search String** are off.
- Click the **Replace All** button.
The entire text of the selected elements will be set to the new font, size, and/or style.

Replacing All instances of a font, size, and/or style within the formatting of the text of all selected elements:

- In the **Search** box, select **Font**, **Size**, and/or **Style** and set the font, size, and/or style to search for.
- In the **Replace** box, select **Font**, **Size**, and/or **Style** and set the new font, size, style.
- Click the **Find** and **Replace** Buttons or the **Replace All** button.
Instances of the selected combination of the search font, size and/or style will be replaced with the new font, size, and/or style.

Changing the formatting of all instances of a search string within the text of all selected elements:

- In the **Search** box, select **Search String** and enter the search string.
- In the **Replace** box, select **Font**, **Size**, and/or **Style** and set the new font, size, and/or style.
- Click the **Find** and **Replace** Buttons or the **Replace All** button.
Instances of the search string, will have the new font, size, and/or style
If more than one search parameter is selected, all parameters in the object must match in order for the replacement to occur.

If more than one replace parameter is selected, the text matching the search parameters will be replaced by all selected replace parameters.

Note that, if only checkboxes in the **Search** section of the **Text Search And Replace** dialog are selected and no selections are made in the **Replace** box, no changes will be made to the picked text objects.
5.7.3 Displaying text

There are three ways to display text in the drafting environment, as outline, stick font, or bounding box. The different options affect screen refresh times.

The options for displaying text are set in the Drafting Display Options dialog, which is invoked through the Display Options... menu item (Display menu). This dialog is shown in Figure 5.7.3.1. The options specific to text are found in the Text Display section of the tab.

Outline: The three items selected from this popup menu control the color in which the outline of the text will be displayed. The options are Off, Black/White, and Color (default).

Fill: The three items in this popup menu, which are the same with Outline, control the color with which outline fonts will be filled. Default is Off.

Therefore, four mutually exclusive ways in which text can be displayed. They are illustrated in Figure 5.7.3.2.

Use Text Element’s Smoothness: When this option is on, which is the default, text is displayed with its actual font and in its actual resolution.

Figure 5.7.3.1: The Drafting Display Options dialog.

Figure 5.7.3.2: Options for displaying drafting text. (a) Outline set to Color and Fill to Off. (b) Outline set to Black/White and Fill to Off. (a, b) Use Text Element’s Smoothness on. (c) Use Stick Font on. (d) Bounding Box on.
Use Stick Font: Selecting this option causes form•Z to substitute the font(s) currently selected for display with form•Z’s stick font.

The stick font is a midway alternative to the slower refresh times of the outline text generated from TrueType, Postscript, and Bitmap fonts, and the fast refresh but illegible bounding boxes (see below). The stick font preserves text element legibility and speeds up screen refreshes, while sacrificing smoothness, double outlines, and fill. The stick font will retain the proportions of the selected outline font to ensure space continuity. The form•Z stick font is described in a formatted ASCII file that is loaded into the program. The font format file is included in the form•Z application folder, and form•Z automatically searches this folder for the stick font. Note that the file needs to remain in the application folder and should not be moved to the system folder. There is currently only one stick font available. When additional stick fonts become available, they will be listed in the pop-out menu. The stick font will be drawn using the stick font file selected from the menu.

Bounding Box: Selecting this option causes form•Z to convert the font(s) currently selected for display to bounding boxes. The bounding boxes provide the fastest screen refresh, but the text elements they represent are illegible. As with the stick fonts, the bounding boxes retain the proportion of the text. They are also drawn with diagonals to easily distinguish them from any rectangular elements that may exist in the project space.

Override Text Smoothness: When this option is active, all the text elements are displayed with a smoothness that can be controlled through the slider bar below the option. The smoothness slider bar functions as its counterpart in the Text Editor dialog.

Bounding Box Below n Pixels: Selecting this option causes all text elements with a display size smaller than n pixels to be displayed as bounding boxes. Using this option avoids the longer screen refresh time of text elements that are illegible because of their small size. However, in contrast to the Bounding Box option, this option fully displays text elements that are greater than the size specified in the numeric field.
5.8 2D drafting symbols

This section discusses tools that create, place, and edit symbols, namely:

- Symbol Create
- Symbol Place
- Symbol Edit

Similar to the modeling symbols, the drafting symbols offer a method by which groups of elements that are to be repeated in one or more drawings can be defined as symbol definitions, and stored in symbol libraries, from which they can be selected and placed in a drawing as symbol instances. In drafting, these operations are executed using the same set of tools that is available in modeling.

Recall that the form•Z symbol libraries consist of a modeling and drafting part, either one of which or both may contain symbol definitions. For users that use both modeling and drafting, it is recommended that in developing symbol libraries, parallel symbol definitions be stored in both the modeling and the drafting parts. That is, an orthographic view that corresponds to a 3D symbol definition stored in the modeling part, should be stored in the drafting part of the same library. While in modeling the three available levels are best used for storing modeling representations at three different levels of detail, in drafting the three levels can be used to store three distinct orthographic projections, such as the top, front, and side views of the symbol. This is the practice that has been followed in the symbol libraries that are shipped with the form•Z package.

Given that the drafting symbols work in a manner very similar to the modeling symbols, the discussion of the drafting symbols in this section relies on the more complete discussion of the modeling symbols contained in section 4.20. Only the differences are discussed in more detail.

Explode is a tool that is very relevant to symbols, but also applies to other types of drafting elements. Because of this it is discussed by itself in the next section. When it is applied to symbols it returns them to their component parts. Note that the Explode tool has a more limited functionality in modeling, where it is only used with symbols.
5.8.1 Symbol libraries, their palette, and dialog

Drafting symbol definitions are stored in the drafting portion of symbol libraries, any number of which can be loaded in a *form-Z* project. One of the libraries is the active library, and its name appears on the top of the Symbols palette. The symbol definitions contained in the active library are listed or displayed in the Symbols palette in one of three formats. The three formats for displaying drafting symbols are shown in Figure 5.8.1.1. The display formats and the details of the Symbols palette are discussed in subsection 4.20.1.

![Symbols palette](image)

*Figure 5.8.1.1:* The Symbols palette displaying drafting symbol definitions in the three available formats.

Symbol libraries and symbol definitions in the drafting environment are also manipulated in the **Symbol Libraries** dialog, which is identical to its modeling counterpart except that it displays the drafting symbol definitions, as shown in Figure 5.8.1.2. It is invoked by clicking in the title bar of the Symbols palette, or by selecting **Symbol Libraries**... from the **Options** menu.

![Symbol Libraries dialog](image)

*Figure 5.8.1.2:* The **Symbol Libraries** dialog displaying drafting symbols.

Recall that the Symbol Libraries dialog contains **Display** options which allow you to switch between displaying the modeling and the drafting symbols. When invoking the dialog, the **Display** selection is initially determined by the type of window that is active. All the operations in the **Symbol Libraries** dialog work as in modeling (see subsection 4.20.2).
5.8.2 Creating and editing symbol definitions

Symbol Create

This tool is used to create a symbol definition from one or more drafting elements and/or other drafting symbol instances. As in modeling, both the prepick and postpick methods can be used for the creation of symbol definitions. However, the postpick method allows you to pick only one element. In order to create a new symbol definition, there must be an active library.

When you click in the drafting window with the Symbol Create tool, the Name New Definition dialog appears in which you enter a name of at most 31 characters and click OK. The symbol is constructed, stored in the active library, and its image or name appears in the Symbols palette.

The options that apply to the creation of symbols are selected from the Symbol Create Options dialog (Figure 5.8.2.1), invoked from the Symbol Create tool. These options are as in the modeling environment with two differences:

- The Status Of Objects tab is not available. Drafting symbols are always created with the Replace With Symbol Instance option. The elements selected for the creation of a symbol are deleted and replaced by a symbol instance.
- The first option reads World Origin (instead of Reference Plane Origin), since the drafting module only uses 2D world coordinates.

The geometry and attributes of symbol definitions can also be edited through a temporary project window by opening a symbol using Open... (File menu) as for modeling (see subsection 4.20.4).

The placement parameters (origin and handle) of a symbol definition can be edited in the Symbol Definition Edit dialog invoked by double clicking on the symbol icon or on the name in the Symbols palette. The drafting version of this dialog is shown in Figure 5.8.2.2. This is essentially identical to its modeling counterpart except for the following differences:

- Under the viewing window, only Fit, Zoom, and Pan icons are available, since there is no 3D.
- There are only two X and Y fields for moving the Handle and the Origin.
5.8.3 Placing symbol instances

Symbol Place

In drafting, symbol instances can be placed using the Symbol Place tool, in the same five ways available in modeling, except that there is one less dimension to deal with. The method used is determined by the option selected in the Symbol Place Options dialog (Figure 5.8.3.1) that is invoked from the Symbol Place tool. In the drafting version of this dialog, there are only two Scale Factor fields (X and Y) and only one Rotation field.

The Pick Origin option, which requires one mouse click, and the Pick Origin And Uniform Scale option, which requires two mouse clicks, work as in modeling. The Pick Origin And X, Y Scale requires three mouse clicks, one less than its counterpart in modeling.

Pick Origin And X, Y Scale: With this option, the symbol instance is placed with three input points. The first determines the location of the instance origin. The second point determines the rotation and, when the option key is not pressed, the X scale factor. If the option key is pressed, the value in the X Scale Factor field is applied. After the first click, a line that corresponds to the symbol definition's extent in the X direction is rubber banded and follows the motion of the mouse until the second click. After the second click, the complete symbol or a bounding box is rubber banded and, when the option key is not pressed, it follows the motion of the mouse, applying a scaling factor in the Y direction. If the option key is pressed, the value in the Y Scale Factor field is applied and no rubber banding occurs. The placement of the symbol instance is completed as soon as the mouse is clicked for the third time. For examples see Figure 5.8.3.2.

![Figure 5.8.3.2: Using the Pick Origin And X And Y Scale method: (a) The original symbol. (b, c, d) Three placements by clicking as shown (on 1, 2, and 3).](image-url)
5.8.4 Editing symbol instances

Symbol Edit

The parameters of symbol instances are edited using the Symbol Edit tool. Both the prepick and postpick methods can be used. When the prepick method is used to preselect several symbol instances, these instances can be edited individually or simultaneously.

The method to be used is selected in the Symbol Edit Options dialog (Figure 5.8.4.1), which is identical to that used in modeling and is invoked from the Symbol Edit icon. Each method invokes a different variation of a Symbol Instance Edit dialog, the drafting versions of which are shown in Figure 5.8.4.2.

The drafting versions of the Symbol Instance Edit dialogs differ from the modeling in that there are only two fields (X and Y) for Origin and Scale, one for Rotation, and only three icons (Fit, Zoom, and Pan) for manipulating the preview image. They also contain the Symbol Instance Text field which is not available in modeling.

Figure 5.8.4.2: The drafting Symbol Instance Edit dialogs invoked when (a) Edit Individually and (b) Edit Simultaneously is on.

When a drafting symbol definition contains text elements, their content can be changed each time a symbol instance that references it is placed. In other words, each symbol instance can store and display a text string which overrides the text string in the symbol definition it references. This mechanism makes it possible to create, for example, drafting symbols which contain text elements with notes or sheet numbers which may vary from instance to instance.
Symbol Instance Text: This text field, shown in Figure 5.8.4.3, displays the text currently in the instance. It can be changed by typing another string. If a blank text string (i.e. no characters) is entered, then the text string in the symbol definition will be used. The dialog also contains a previous and a next arrow which allow you to scroll through all the text elements of the definition.

All the other options in the Symbol Instance Edit dialog work as in the modeling environment. However, certain drafting element types, namely arcs, circles, ellipses, rectangles, and text, behave differently when scaling transformations are applied to them, because the elements themselves have their own local coordinate system. These types of elements behave like symbol instances nested in a symbol definition. Examples are shown in Figure 5.8.4.4.

In drafting, when placing an arc, the first two points determine the direction of the X axis of the arc’s coordinate system, even when that direction coincides with the Y axis of the world coordinate system, as shown in Figure 5.8.4.4(1a). When such an arc is included in a symbol definition which is placed as an instance and a uniform scale is applied, the result appears normal because scaling is applied in all directions. However, if the instance is placed by applying independent scale factors, they will be applied to the coordinate system which is local to the arc element.

Figure 5.8.4.4(1) shows a door symbol created from a rectangle and an arc. The arc was created by clicking the second point vertically above the first point. Therefore, the local X axis of the arc aligns with the Y direction of the world coordinate system (a). When the door symbol is placed with a uniform scale, it is displayed as expected (b). When a non uniform scale is applied, such as X = 1.0 and Y = 2.0, the arc appears stretched in the X direction (c). Figure 5.8.4.4(2a) illustrates the same symbol with the arc defined by points matching the world coordinate system. The placed symbol displays properly with (b) uniform, and (c) non uniform scale factors.
5.9 Exploding drafting elements

This section discusses the Explode tool which is useful for reducing a symbol to its component parts as well as for reducing other higher level drafting elements to their next lower level.

Some types of the drafting elements are either completely screened or simply constrained when certain operations are applied to them. This allows them to retain their specific character and to behave accordingly, when they are operated on. On the other hand, the polylines are the most permissive and are receptive to all the drafting operations.

The drafting environment provides the Explode tool for transforming any type of element to a set of polylines. The only exceptions are the text image and point elements. When transformed to polylines, all the drafting elements retain their shape and they produce the same image when they are displayed. However, they lose their special character as well as their ability to constrain themselves when operations are applied to them. When transformed to polylines, they can be freely operated on, as any polyline.

For example, after being exploded, an arc becomes a polyarc and any of its segments can be freely moved, which is not possible while it is internally structured as an arc. After it is transformed to a polyline, the arc loses its ability to preserve its circular shape when it is operated on. After exploding a dimension, it becomes a collection of polylines and text, any of which can be individually moved. However, the dimension loses its associativity and can no longer be automatically adjusted when the points it references are moved.
Explode

This tool reduces the structure of a drafting element by one level. The Explode tool ignores the topological level and is always applied at the Element level. It can be executed using either the prepick or the postpick method. The exact result of the Explode tool depends on the type of the drafting element to which it is applied. To reduce an element to a collection of single segment polylines typically requires that the Explode tool be applied more than once. The structural levels for each type of drafting element are summarized in the table of Figure 5.9.0.1.

The most elementary elements are the single point and single segment polyline (single line), text, and image element. They cannot be exploded any further. An open or a closed polyline, when exploded, is reduced to a group of single lines. Exploding a rectangle produces a common polyline, which can be further exploded to four single segment polylines. Similarly, an open or a closed arc is exploded to an open or closed polyline, which can be further exploded to individual lines.

A leader line consists of its main line and possibly one or two line terminators. Each becomes a polyline when it is exploded and each can be further reduced to individual lines. A dimension consists of a dimension line (which may be in two parts), two extension lines, two terminators and a text block. When exploded, it transforms into five or six polylines and a text element. The polylines can be exploded further to become single segment elements.

Compounds are exploded to the elements from which they were constructed and so are the symbols. How many times the Explode tool needs to be applied to reduce these elements to single segment polylines and text, depends on the types of elements that were used for their construction.

Figure 5.9.0.1: The structural levels of the drafting elements
5.10 Dimensions and leader lines

This section discusses the tools that generate and manipulate dimensions and leader lines, namely:

- Dimension Horizontal/Vertical
- Dimension Parallel
- Dimension Angular
- Dimension Radius
- Dimension Diameter
- Dimension Arc
- Leader Line
- Get Dimension Attributes
- Set Dimension Attributes
- Reposition Dimension

Dimensions are used to indicate physical sizes of elements, distances between elements, or parts of elements, in a drawing. Leader lines are used to indicate an association between two or more elements, or to attach additional textual or graphic information to other elements.

Dimensions are constructed automatically by the system on the basis of the points of elements you select. They are generated according to the type of dimension that is applied and the current settings of the parametric options that affect them. The numeric information the dimensions display is generated automatically. You determine the exact position where they are drawn.

There are four basic types of dimensions: linear, angular, radius/diameter, and arc dimensions. They are illustrated in Figure 5.10.0.1. The linear dimensions may be horizontal, vertical, or parallel to an orientation that is determined through a variety of available methods. Individual or multiple dimensions can be created. Multiple dimensions can be drawn as chains, stacks, or in an ordinate arrangement. Multiple dimensions may also be linked and, when they are, they behave as compound entities. There is a single type of leader line, but a number of variations are available.

Figure 5.10.0.1: The four dimension types: (a) linear, (b) angular, (c) diameter, and (d) arc.
The generation of dimensions and leader lines is affected by parametric options at three different levels. It is affected by type specific options that are selected from dialogs which are invoked directly from the tools that generate each type of dimension. It is also affected by the general dimension and leader line attributes, which are selected from a number of dialogs invoked from the type specific dialogs. Finally, it is affected by the general parametric conditions that affect the generation of all the drafting elements, namely the active line weight and type, the active color, the active layer, and the current arc/circle/ellipse resolution.

A dimension always has a dimension line and a text block. A dimension can also have up to two extension or witness lines, and up to two line terminators. The dimension line may be straight or circular, depending on the type of the dimension. It may be a continuous line or it may be broken into two portions, depending on the position of the text. Several types of line terminators can be placed at the intersections of the dimension line with the witness lines, and are used to indicate the exact positions of the points of measurement. The parts of the dimensions are illustrated in Figure 5.10.0.2.

A significant feature of the dimensions and the leader lines, as implemented in formZ, is their associativity. However, creating associated dimensions and leader lines is an option; non-associated dimensions and leader lines are also available. Associativity causes an annotation to be automatically adjusted when one or more of the points it references (it is associated with) are moved. This includes the numeric information it displays. The feature is illustrated in Figure 5.10.0.3.
5.10.1 Linear dimensions

This type of dimensions includes two tools that are discussed in this subsection: the Dimension Horizontal/Vertical and Dimension Parallel tools.

The Dimension Horizontal/Vertical tool generates dimension lines in a horizontal or vertical orientation, depending on the options selected in its dialog. The Dimension Parallel tool places dimensions either parallel to the two points they reference when they are generated, or at any orientation for multiple dimensions depending on the options selected in its dialog.

The linear dimension operators place dimension lines which indicate a linear distance between pairs of points. The distance can be measured in the horizontal or vertical direction, parallel to the two referenced points with individual dimension lines, or along any other orientation with a uniform dimension line. The selected points may be from the same or different drafting elements. A minimum of two points must be selected to define a dimension. However, multiple points may also be selected to create any of the three graphics types that are discussed later in this subsection.

With horizontal dimensions, vertical dimensions, and parallel dimensions with uniform dimension lines, the orientation of the dimension line is chosen independently from the actual orientation of the line that can be constructed to connect the dimensioned points. These three types of dimensions define the distance that is the orthogonal projection of the actual distance between the dimensioned points onto the dimension line. The projected distance and the actual distance are identical only if the line that connects the dimensioned points and the dimension line are parallel. This parallel orientation is guaranteed by the parallel dimension with the individual dimension line option. Therefore, the latter cannot be constructed as a multiple dimension with uniform dimension line orientation, regardless of whether the orientation is horizontal, vertical, or at any other angle.
Dimension Horizontal/Vertical

The Dimension Horizontal/Vertical tool is used to generate horizontal and/or vertical dimensions using either the postpick or the prepick method.

With the postpick method, the Dimension tool is activated first, then any number of points are selected in the order in which they are to be dimensioned. The Topological Level currently selected is ignored, as this tool always picks at the Point level. The selection is completed by a double click of the mouse. The position where the mouse is double clicked determines the position where the dimension line initially appears. It is rubber banded and follows the motion of the cursor. A final click of the mouse determines the location of the dimension line(s), and the dimension element is constructed.

With the prepick method, use the Pick tool to first select the points in the order in which they are to be dimensioned, and then select the Dimension Horizontal/Vertical tool. A single click in the graphics window displays the dimension and initiates the rubber banding process. From this point on the prepick method works identically to the postpick method. Note that no advantage is gained when the prepick method is used, since more than two points can also be selected when using the postpick method. Prepick actually has the disadvantage that only existing points can be picked for dimensioning, while the postpick method can also dimension to positions where there are no points.

The Dimension Horizontal/Vertical tool may be restricted to generate horizontal dimensions only, or vertical dimensions only, or to automatically decide whether to generate a horizontal or a vertical dimension. This is determined by the selection of the respective option in the Dimensions: Horizontal/Vertical Options dialog shown in Figure 5.10.1.1. This dialog can be invoked directly from the Dimension Horizontal/Vertical tool. It contains options that control the Orientation, the Graphics, the Association, and the Link Multiple Dimensions settings for the Dimension Horizontal/Vertical tool.
**Orientation:** The three items contained in this pop up menu determine the orientation in which the dimension will be generated.

**Horizontal:** When this item is selected, horizontal dimensions are generated exclusively. When the two points picked have zero horizontal distance, a horizontal dimension cannot be generated and an error message is posted.

**Vertical:** When this item is selected, vertical dimensions are generated exclusively. When such a dimension cannot be generated due to zero distance, an error message is posted.

**Dynamic Horizontal/Vertical:** When this item is selected, the system will automatically determine whether to generate a horizontal or a vertical dimension, depending on the position of the mouse cursor relative to the last point selected for dimensioning. That is, if the horizontal distance is greater than the vertical distance, a horizontal dimension will be drawn, and vice versa. During the rubber banding process, the system switches from one orientation to the other, depending on the current position of the cursor. When points are picked that have zero horizontal distance, the dimension orientation defaults to vertical. If subsequently points are picked with zero vertical distance, the dimension cannot be generated. An error message is posted and the operation is terminated. The same error condition results when a zero vertical distance is encountered first, followed by a zero horizontal distance. These options are illustrated in Figure 5.10.1.2.

![Figure 5.10.1.2](image)

**Figure 5.10.1.2:** (a) **Dynamic Horizontal/Vertical:** Selecting a1 and a2, the dimension may be placed either horizontally or vertically, depending on whether the mouse is clicked on a3 or a4.

(b) **Horizontal:** Selecting b1 and b2, and clicking on b3 and b4: both generate a horizontal dimension.

(c) **Vertical:** Selecting c1 and c2, and clicking on c3 and c4: both generate a vertical dimension.

(d) **Dynamic Horizontal/Vertical:** Selecting d1 and d2, and clicking on d3 and d4: both generate a horizontal dimension, because vertical distance is zero.

(e) **Dynamic Horizontal/Vertical:** Selecting e1 and e2 and clicking on e3 and e4: both generate a vertical dimension, because horizontal distance is zero.

(f) **Horizontal:** Selecting f1 and f2, the operation terminates because horizontal distance is zero.

(g) **Vertical:** Selecting g1 and g2, the operation terminates because vertical distance is zero.
Graphics: The three items in this pop up menu only affect multiple dimensions, that is dimensions generated when more than two points are selected.

Chain: This item produces a colinear sequence of dimensions, as shown in Figure 5.10.1.3(a). The chain dimensions measure the distances between pairs of consecutive points.

Stacked: This item produces dimensions that are stacked in the order of selection, as shown in Figure 5.10.1.3(b). These dimensions measure the distance between the first point and each of the other points selected. The stacked dimension lines are drawn at equal distances to each other, starting with the outermost dimension that is placed at the location of the last click.

Ordinate: This item draws, for each selected point, a line perpendicular to the dimension orientation. The dimension value is displayed at the end of this line (Figure 5.10.1.3(c)). These dimensions measure distances from the first point selected, as do the stacked dimensions.

Association: This pop up menu contains items that set the association of the placed dimension.

Automatic: With this option, the placed dimension is associated, non-associated, or mixed depending on whether a click point is on a point of an element (associated) or away from a point (non-associated). If all the points picked are associated, the entire dimension is associated. If they are all non-associated, so is the whole dimension. If some points are associated and some are not, the dimension has mixed associativity. The Automatic option is on by default.

Always: When this option is on, only associated dimensions are created and if a click is made that is not close to a point of an element, an error message is posted.

Never: When this option is on, only non-associated dimensions are generated, regardless of the proximity of the clicks to the points of elements.

Link Multiple Dimensions: With this option on, multiple dimensions become compound elements; that is, they are linked to each other. While this has no visible effect, linked dimensions behave differently when geometric transformations are applied to them (see subsection 5.10.8). When linked dimensions are picked with topological level set to Compound, they behave as a single unit. They can also be moved individually with topological level set to Element.

Dimension Lines, Witness Lines, Terminators, Dimension Text: The content of these tabs is discussed in subsection 5.10.6. The dialog invoked by the Text Style... button is also discussed in subsection 5.10.6.
Dimension Parallel

The Dimension Parallel tool is used to generate dimensions parallel to a slope that may be defined through a variety of available methods selected from the Dimensions: Parallel Options dialog (Figure 5.10.1.4), invoked directly from the Dimension Parallel tool. The Dimension Parallel tool can be applied using either the postpick or prepick method, as with the Dimension Horizontal/Vertical, except when the Orientation As Picked option is selected, which requires additional mouse clicks (see below).

Individual Dimension Line: When this placement method is selected, all other options in the dialog are dimmed and inactive. This option produces an individual dimension line for each pair of points selected for dimensioning. Each dimension line is parallel to the orientation defined by the two points selected. These dimension lines are constructed on the same side of the object for all pairs of points, and at the same distance from it, which is determined by the distance of the final mouse click from the last pair of points picked. These dimensions always maintain an orientation parallel to the dimensioned points during all subsequent manipulations. An example is shown in Figure 5.10.1.5.

Uniform Dimension Line: When this option is selected, a single multiple dimension is generated whenever more than two points are selected. That is, all the dimensions have the same orientation. This type of parallel dimension generally resembles the horizontal and vertical dimensions. There are three methods for determining the orientation.

Orientation By Angle: When this option is selected, the orientation of the dimension is determined by the value entered in the text field next to it. The value is in degrees measured from the positive X direction (3 o’clock).
**Orientation As Picked:** When this method is selected, after the double click the system expects the input of two additional points to determine the orientation of the dimension line. After the first point is entered, a line is rubber banded and follows the motion of the cursor until the next click, which establishes the orientation of the dimension as the angle between the two points. From this point on, the dimension is rubber banded, and as it moves it remains parallel to the established orientation until the final mouse click which determines its exact location. An example is shown in Figure 5.10.1.6.

![Figure 5.10.1.6: Parallel dimension with Uniform Dimension Line and Orientation As Picked on: Select points 1 through 6, then 4 and 5 again, then 7.](image)

**Dynamic Orientation:** When this option is selected, the orientation of the dimension line is determined during the rubber banding process. After the double click, a dimension line that is perpendicular to the line which connects the last point selected with the position of the cursor is rubber banded and follows the motion of the cursor. The final click establishes the orientation as well as the position of the dimension line.

**Graphics:** These options (Chain, Stacked, Ordinate) are as for the horizontal and vertical dimensions, discussed in the previous subsection.

**Association:** These options (Automatic, Always, Never) are as for the horizontal and vertical dimensions, discussed earlier in this subsection.

**Link Multiple Dimensions:** Similar to the horizontal and vertical dimensions, selection of this option links the uniform parallel dimensions when they are generated as multiple dimensions. Linked dimensions preserve their uniform orientation when transformations are subsequently applied to them.

**Dimension Lines, Witness Lines, Terminators, Dimension Text:** The content of these tabs is discussed in subsection 5.10.6. The dialog invoked by the Text Style... button is also discussed in subsection 5.10.6.
5.10.2 Angular dimensions

The angular dimensions are used to measure and post the size of an angle, in degrees. They indicate the angle that is defined by three points. One of these three points is the apex of the angle; the other two points establish the two sides between which the angle is measured. Angular dimensions can be generated as multiple dimensions when more than three points are selected.

Dimension Angular

The Dimension Angular tool is used to generate dimensions that measure angles. It can be used with either the postpick or the prepick method. With the postpick method, first activate the Dimension Angular tool, then click on three or more points to select them, then double click to terminate the selection of the points and to start the rubber banding of the dimension line. One final click generates the dimension line at the position where the mouse was clicked. If fewer than three points are selected, an error message is posted. If more than three points are selected, a multiple dimension is generated.

With the prepick method, use the Pick tool to preselect three or more points with the Topological Level set to Point. Then activate the Dimension Angular tool, and click in the graphics window. From this point on the prepick method works the same as the postpick method.

The order in which the points are selected determines whether the dimension measures the interior or the exterior angle. The apex may be the first or the second point picked, depending on the option selected in the Dimensions: Angular Options dialog (Figure 5.10.2.1). This dialog can be invoked directly from the Angular Dimension tool.

**Dimension Apex**: These options determine which of the first two picked points is designated as the apex point of the angle to be measured.

**First Point**: When this option is on, the apex is picked with the first click. The next two points establish the two sides of the angular dimension.

**Second Point**: With this option on, the apex is determined by the second click. The first point and the third point establish the two sides of the angle. When more than three points are picked, multiple angular dimensions are generated.

*Figure 5.10.2.1*: The Dimensions: Angular Options dialog.

*Figure 5.10.2.2*: Angular dimensions: (1) Apex at first and (2) second point. Measuring (a) inside and (b) outside angle.
Angle: This pop up menu determines if the angular dimension is placed **Counter-Clockwise** (default) or **Clockwise** (Figure 5.10.2.3).

Dimension Graphics: This is the same pop up menu available for the linear dimensions, except that its options are applied in a circular manner.

Chain: This item generates dimension arcs around the apex, with the same radius for all the dimensions in the chain. An example is shown in Figure 5.10.2.4(a).

Stacked: This option generates concentric dimension arcs around the apex, with a constant distance between them. The angles are measured from the first point selected (that is not the apex). An example is shown in Figure 5.10.2.4(b).

Ordinate: This option generates lines which pass through the points which delineate the angles and converge to the apex point. The dimension values are posted at the end of these lines and measure angles from the first point selected, as with the stacked dimensions. An example is shown in Figure 5.10.2.4(c).

Association: These items (Automatic, Always, Never) are as for the horizontal and vertical dimensions, discussed in subsection 5.10.1.

Link Multiple Dimensions: When this option is selected, multiple angular dimensions are linked and maintain their graphic layout during subsequent transformations.

Dimension Lines, Witness Lines, Terminators, Dimension Text: The content of these tabs is discussed in subsection 5.10.6. The dialog invoked by the Text Style... button is also discussed in subsection 5.10.6.
5.10.3 Radius and diameter dimensions

Recall that the drafting module of form-Z, in addition to the polyarcs (circular shapes approximated by polylines) contains a parametric element called arc. The radius and diameter dimensions discussed in this subsection apply to the arc elements only, which include complete circles, ellipses, and open arcs. These dimensions can be generated with a **horizontal**, **vertical**, or an **arbitrary** orientation, either through the center point of the dimensioned element or at any location. With ellipses and elliptical arcs, the radius and diameter dimensions can only be drawn on their major or minor axes.

**Dimension Radius**

**Dimension Diameter**

Radius and diameter dimensions are generated with the Dimension Radius and Dimension Diameter tools, using either the postpick or the prepick method. With the postpick method, activate either tool and click on the arc element to be dimensioned. A radius or a diameter dimension may be generated immediately or a dimension may be rubber banded until the mouse is clicked again. This depends on the option used for the generation of the radius or diameter dimension, as discussed below. Also, when rubber banding occurs, its behavior varies depending on the options used for the generation of the dimension. With the postpick method, the active Topological Level is ignored, and the Dimension Radius or Dimension Diameter tools always pick at the Element level.

With the prepick method, use the Pick tool to preselect any number of arc elements, with Topological Level set to Element. Then activate the Dimension Radius or Diameter tool and click in the graphics window. From this point on the prepick method behaves as the postpick.

The type specific parametric options that affect the generation of the radius and diameter dimensions are selected from the **Dimensions: Radius Options** or **Diameter Options** dialogs, the former of which is shown in Figure 5.10.3.1. The dialogs are identical, except in name, and are invoked from the respective tools.

![Dimensions: Radius Options dialog](image-url)

**Figure 5.10.3.1:** The **Dimensions: Radius Options** dialog.
**Orientation:** This group of options controls the orientation at which the radius or diameter dimension is constructed.

**Horizontal/Major Axis:** Selection of this option produces a horizontal dimension when the dimensioned element is circular (has constant radius). With ellipses and elliptical arcs the dimension is constructed parallel to the major (long) axis.

**Vertical/Minor Axis:** Selection of this option produces a vertical dimension when the dimensioned element is circular (has constant radius). With ellipses and elliptical arcs the dimension is constructed parallel to the minor (short) axis.

**Parallel Dimension Line:** When this option is selected, a dimension line with an orientation at an angle is constructed. The orientation may be defined through one of three available options.

**Orientation By Angle:** With this option, the angle of the orientation is specified numerically by entering a value in degrees in its field.

**Orientation As Picked:** When this option is selected, the orientation of the dimension is determined by the line that joins the center of the dimensioned element and the point where the mouse was clicked when the element was selected. If the element was picked by area, or with the Select All Unghosted item (Edit menu), and consequently no pick point is defined, this option reverts to the Orientation By Angle option.

**Dynamic Orientation:** When this option is selected, the orientation of the dimension is determined through dynamic action. That is, a rubber banded dimension line is first generated through the point where the mouse is clicked and follows the motion of the mouse until it is clicked again. The position of this click determines the final orientation of the dimension line. With elliptical elements, the rubber banding action is not continuous, but the line snaps to the closest axis of the elliptical element and switches between the two axes as the mouse moves.
Method: Two placement methods are available.

Through Center: This option forces the dimension line to pass through the center of the dimensioned element. If the Dynamic Orientation option is selected, the dimension line is rubber banded and rotates about the center of the arc element as the mouse moves. When this option is not selected, the dimension is generated immediately when the element is picked.

Dynamic: When this placement method is selected, the position of the dimension line is determined through dynamic action. A line parallel to the specified orientation is rubber banded, and follows the motion of the mouse until it is clicked again. The final position of the dimension line is determined by the last click. When this placement method is combined with the Dynamic Orientation option, then both the orientation and the position of the dimension line are determined through rubber banding. As the mouse moves, the dimension line both rotates about the center and moves farther or closer to it, until the mouse is clicked again.

Association: The items in this pop up menu (Automatic, Always, Never) are as for the horizontal and vertical dimensions, discussed in subsection 5.10.1.

Leader Dimension: When this option is selected, Through Center is the only available placement Method and the Dynamic option is dimmed. This option causes the Dimension Radius/Diameter tool to generate a leader dimension.

A leader dimension is similar to the regular radius/diameter dimension constructed with the Through Center option selected, except for the placement of the text (Figure 5.10.3.8). The leader dimension includes a two segment leader line extension which is placed outside of the dimensioned element. One end of the leader line touches the dimensioned element, and text is attached to the other end. The leader dimension is affected by the same options that apply to the regular radius/diameter dimension.
Another option that affects the graphics of the leader dimensions is the **Always Draw Dimension Line Inside** option, which is selected from the **Dimension Lines** tab (Figure 5.10.6.1). It controls whether a dimension line is generated inside of the dimensioned arc, circle, or ellipse, as shown in Figure 5.10.3.9. When it is on, a dimension line with two terminators is generated which indicates the radius or the diameter of the dimensioned element. The **start terminator** is placed on the center of the arc, for radius dimensions, or at the end which is opposite to the leader extension, for diameter dimensions. The **end terminator** is placed where the leader line extension is generated. When the **Always Draw Dimension Line Inside** option is off, no dimension line is generated inside the dimensioned element, and a terminator corresponding to the end terminator is drawn at the end of the leader line extension that touches the perimeter of the dimensioned element.

![Figure 5.10.3.9: (1) Radius and (2) diameter leader dimensions generated with the Always Draw Dimension Line Inside option (a) on and (b) off.](image)

**Dimension Lines, Witness Lines, Terminators, Dimension Text**: The content of these tabs is discussed in subsection 5.10.6. The dialog invoked by the **Text Style**... button is also discussed in subsection 5.10.6.
5.10.4 Arc dimensions

The arc dimensions measure the lengths of arcs, and can only be applied to arc type of elements that are not elliptical and are not closed. Although similar in appearance to the angular dimensions, the arc dimensions denote the distance along an arc, and not its angle.

**Dimension Arc**

Arc dimensions are generated with the Dimension Arc tool, which works with both the postpick and the prepick method. With the postpick method, activate the Dimension Arc tool and click on an arc. A circular dimension concentric to the selected arc is rubber banded and follows the motion of the mouse until the mouse is clicked again, which determines the final position of the dimension line. With the prepick method, use the Pick tool to preselect any number of circular arcs with Topological Level set to Element. Then activate the Dimension Arc tool and click in the graphics windows. This rubber bands an arc dimension for each of the preselected elements. These dimensions are initially placed at a distance which is determined by the distance of the click point from the arc that was selected last.

The Dimension Arc tool invokes the **Dimensions: Arc Options** dialog (Figure 5.10.4.1).

**Association:** This pop up menu is as for the other dimensions.

**Non Associative Placement:** When placing a non associative arc dimension, since there is no arc for the dimension to reference, one must be drawn. This group of six icons represent the standard arc drawing methods available in form•Z. The selected method will be used when placing non associative arc dimensions.

The witness lines of the arc dimensions can be generated parallel to each other or at an angle. In the latter case, they pass through the endpoints of the arc and, if extended, through the center of the arc. These options are selected from the **Witness Lines** tab and are discussed in more detail in subsection 5.10.6. Examples of arc dimensions are illustrated in Figure 5.10.4.2.

**Dimension Lines, Witness Lines, Terminators, Dimension Text:** The content of these tabs is discussed in subsection 5.10.6. The dialog invoked by the **Text Style...** button is also discussed in subsection 5.10.6.

![Figure 5.10.4.1: The Dimensions: Arc Options dialog.](image1)

![Figure 5.10.4.2: Arc dimensions with (a) parallel and (b) angular witness lines.](image2)
5.10.5 Leader lines

A leader line is a special type of polyline that may be used to point from one element to another, or to indicate that some block of textual or graphic information refers to another element. Either or both of its end points may have a line terminator, and may be associated with points of other elements. An associated end of a leader line follows the point with which it is associated when that point is moved. Leader lines can only be associated with points of polylines, rectangles, and point elements.

![Leader Line](image)

This is essentially a line drawing tool that allows you to generate a leader line, which is drawn as an open polyline. Using the mouse, draw a connected sequence of straight line segments. Each click creates a new segment and the drawing process is completed by a double or a triple click. A leader line is always open, even when it is terminated with a triple click. The first and last points of a leader line are referred to as start and end points, respectively. Whether an endpoint of a leader line is associated with a point of another element is controlled by options selected in the Dimensions: Leader Line Options dialog, shown in Figure 5.10.5.1, which can be invoked directly from the Leader Line tool.

**Association:** These options (Automatic, Always, Never) are as for the horizontal and vertical dimensions, discussed in subsection 5.10.1.

**Terminators:** This tab contains the options for the terminators for the start and end points can be set, as discussed in the next subsection.

![Figure 5.10.5.1: The Dimensions: Leader Line Options dialog.](image)

When the endpoint of a leader line is associated with a point of a polyline, it can only be associated with a single point. Associated points of leader lines are updated when the elements they are associated with are transformed, as illustrated in Figure 5.10.5.2. Associated points also affect the behavior of leader lines during transformations, as discussed in subsection 5.13.8.

![Figure 5.10.5.2: The behavior of associated leader lines when the points they reference move: (a) non associated, (b) one and (c) two endpoints associated. (Associated points are shown with terminators.)](image)
5.10.6 The dimension and leader line attributes

New dimensions and leader lines are affected by parameters set at three different levels:

(i) Global parameters that affect all the drafting elements, namely the active line type and line weight, the active layer on which they are placed, and the active color that is assigned to them.

(ii) The type specific parametric options that are selected from the dialogs that are invoked from each dimension tool, as discussed in the previous subsections.

(iii) General parametric options that affect all dimensions and leader lines, which are selected from the Options (Figure 5.10.6.1), Witness Lines (Figure 5.10.6.4), Terminators (Figure 5.10.6.14), and Dimension Text (Figures 5.10.6.16 and 5.10.6.24) tabs.

The Options tab

The options in this tab, shown in Figure 5.10.6.1, are as follows:

Outside Dimension Line Length: This value controls the length of the dimension lines drawn outside the witness lines (rather than between them), whenever the dimension text does not fit between the witness lines.

Minimum Dimension Arc Radius: This field sets how close to the apex angular/arc dimensions can be. With stacked dimensions, this affects the inner most dimension.

Stacked Dimension Line Distance: This value sets the distance between stacked dimension lines (Figure 5.10.6.2).

Always Draw Dimension Line Inside: With this option, a dimension line is always generated between the witness lines, even when the text is outside due to lack of space (Figure 5.10.6.3). This does not affect the text placement, nor does it suppress the generation of outside dimension lines. This option also causes ordinate dimensions to be drawn with dimension lines.
The Witness Lines tab

The options in this tab, shown in Figure 5.10.6.4, are as follows:

**Show Witness Line**: The two options *At Start Point* and *At End Point*, when selected, cause a witness line to be drawn at the start and/or end point of a dimension, respectively.

**Offset From Element**: This value specifies the space that is automatically generated between the dimensioned point and the start of the witness line (Figure 5.10.6.5). For leader dimensions, this value specifies the length of the first segment of the leader line extension of the leader dimension, as shown in Figure 5.10.6.7. First is the segment that touches the elements being dimensioned.

**Witness Line Extension**: This value determines the length of the part of the witness line that extends beyond the dimension line or arc (Figure 5.10.6.6). This is the extension on the side of the dimension line opposite to the side of the dimensioned point. For leader dimensions, this value specifies the length of the second segment of the leader line extension, as shown in Figure 5.10.6.7. The second segment is aligned with the text, which is oriented according to the Drawing Read Direction option, selected in the Dimension Text dialog.

![Figure 5.10.6.4: The Witness Lines tab.](image)

![Figure 5.10.6.5: Using different values for the Offset From Element parameter.](image)

![Figure 5.10.6.6: Using different values for the Witness Line Extension parameter.](image)

![Figure 5.10.6.7: Leader dimensions with Witness Line Extension = 4", and Offset From Element = (a) 12", (b) 16", (c) 20".](image)
**Uniform Line Length**: With this option, a uniform length is applied to all witness lines, which is determined by the value entered in the field next to it. The uniform length is measured from the dimension line toward the dimensioned point or element (Figure 5.10.6.8).

With stacked dimensions the uniform length is applied differently. All witness lines are not of the same length but rather their ends are aligned. The position of the end points is determined by the witness line closest to the first point, which is set to a length equal to the value in the numeric field (Figure 5.10.6.9).

The **Offset From Element** and **Uniform Line Length** parameters affect each other. When both are applied, they may produce different lengths for the same witness line. In such a case, the witness line is constructed with the shortest of the two lengths (Figure 5.10.6.10). The result produced by the **Witness Line Extension** parameter is only affected when the dimension line is placed very close to the dimensioned entity, when no witness lines are generated (Figure 5.10.6.11).

**Arc Dimension Witness Lines**: These options affect the arc dimensions only.

**Parallel**: When this option is selected, the witness lines of arc dimensions are generated parallel to each other (Figure 5.10.6.12).

**Angular**: When this option is selected the witness lines converge to the apex point of the angle being measured and pass through each of its side points (Figure 5.10.6.13).

**Parallel When Angle Less Than**: When this option is selected and the measured angle is less than the angle specified in its numeric field, parallel witness lines are generated. Otherwise, angular witness lines are generated.
The Terminators tab

This tab, shown in Figure 5.10.6.14, contains options that specify the terminators that may be used to delimit the dimension lines or arcs, and may be placed at the ends of leader lines.

Terminators can be set separately for the Start Point and the End Point of a dimension or leader line. The available options are to place None (no terminator), or any of the five types: Arrow Head, Slash Mark, Dot, Circle, or Cross. The dot and the circle are similar except that the circle is drawn at a higher resolution than the dot. The arrow head, the dot, and the circle terminators can also be filled by selecting the Fill box.

Terminator Size: The values entered in these fields control the size of the terminators, independently for the Start and End Point and independently for their Length and Height. Length is the direction along the dimension line. Height is the direction perpendicular to the dimension line, and parallel to the witness line.

Figure 5.10.6.14: The Terminators tab.

Figure 5.10.6.15: Line terminators: (1) none, (2) arrow head, (3) slash, (4) dot, (5) circle, (6) cross, (7) filled arrow head, (8) filled dot, (9) filled circle, (10) arrow heads, and (11) filled circle and arrowhead.
The Dimension Text tab

The options in this tab, shown in Figure 5.10.6.16, control how text is laid out and placed in the dimensions.

**Drawing Read Direction:** These mutually exclusive options affect how dimension text is oriented relative to the horizontal axis.

- **Bottom:** When this option is selected, all the text is oriented horizontally, and is read from the bottom of the drawing. This is true even for vertical dimension lines.

- **Bottom & Right:** When this option is selected, the dimension text is aligned to either the horizontal or the vertical direction, depending on whether the angle of the dimension line with the horizontal axis is less or greater than 45°, respectively. When equal to 45°, the text is directed horizontally. When it is directed vertically, it is read from the right.

- **Normal:** This option is the default. When selected, the text appears at the orientation of the dimension line. These options are illustrated in Figure 5.10.6.17.

**Distance To Terminators:** This value controls the minimum spacing between the dimension text and the dimension's terminators, if any. Depending on the length of the dimension text, the sizes of the terminators, and the space specified by this parameter, the text may not fit between the terminators. In such a case both the text and the dimension lines are generated outside the witness lines. Examples are shown in Figure 5.10.6.18.

---

*Drafting • Dimensions and leader lines*
**Offset From Dimension Line:** This value controls the distance between the dimension text block and the dimension line. How this offset distance is applied also depends on the **Justification** option selected from the **Dimension Text Edit** dialog (see below). For **Bottom** justification the offset is measured from the lower edge, for **Center** from the vertical midpoint, and for **Top** from the upper edge of the text block. In the first two cases a positive offset moves the text up, while in the third case a positive number shifts the text down relative to the dimension line. These variations are illustrated in Figure 5.10.6.19.

**Dimension Line Split Distance:** When the text block intersects the dimension line, then the latter is broken into two portions, one on each side of the text. This parametric value specifies the space between the ends of a split dimension line and the text block. Examples are shown in Figure 5.10.6.20.

**Always Place Text Between Witness Lines:** This option, when selected, forces the dimension text to be placed between the witness lines, regardless of whether or not it fits. This option is useful with dimension chains, where text placed outside of the witness lines may overlap with other text. This option can be used in conjunction with the **Width** parameter that is set in the **Dimension Text Edit** dialog to squeeze text blocks to fit in the available space. Examples are shown in Figure 5.10.6.21.

**Show Unit Indicators:** When this option is selected, the unit indicators are also displayed with the dimension numerals. This option is on by default. It can be set independently for each dimension and can be read and altered as all the other dimension attributes, using the Get and Set Dimension Attributes tool, or with the Query tool. Examples of dimensions with and without unit indicators are shown in Figure 5.10.6.22.
Prefix and Suffix: These options, when selected with a string of at most 16 characters is entered in their field, append that string before or after the numeric text of a dimension, respectively. Examples are shown in Figure 5.10.6.23.

The Dimension Text Edit dialog

As already mentioned, clicking on the Text Styles... button in the dimension type specific dialog invokes the second Dimension Text Edit dialog shown in Figure 5.10.6.24. The parametric options in this dialog are identical with those in the Drafting Text Editor dialog, discussed in section 5.7.1, except that the text itself cannot be edited, since it is automatically generated by the dimension tools. An independent dialog is used so that the dimension text parameters may be set independently from the drafting text parameters. For examples see Figure 5.10.6.25.

Figure 5.10.6.23: Dimensions generated (a) with Prefix on and the prefix string "PREFIX," and (b) with Suffix on and the suffix string "SUFFIX."

Figure 5.10.6.24: The Dimension Text Edit dialog.

Figure 5.10.6.25: Dimensions placed with varying (a) Font, (b) Height, (c) Width, and (d) Justification parameters.
5.10.7 Changing the attributes of dimensions and leader lines

Once a dimension or leader line has been generated according to the currently active attributes, those attributes may be changed. The line types, line weights, layers, and colors are changed as with the common polylines. All other attributes may be changed by using the Set Dimension Attributes tool. The Get Dimension Attributes tool is also useful when the attributes of one dimension or leader line need to be transferred to another. Note that, even though “Leader Line” is not included in the names of these tools, they apply to both dimensions and leader lines.

Get Dimension Attributes

This tool retrieves the dimension attributes of a dimension or leader line and places them in the **Dimension Lines**, **Witness Lines**, **Terminators**, and **Dimension Text** tabs. It can be applied using either the prepick or the postpick method. With the postpick method, activate the Get Dimension Attributes tool and click on a dimension or leader line. With the postpick method, the current Topological Level is ignored and the Get Dimension Attributes tool always picks at the Element level. With the prepick method, use the Pick tool to preselect one dimension or leader line with Topological Level set to Element, then activate the Get Dimension Attributes tool and click in the graphics window to execute the operation. Note that, when applying the Get Dimension Attributes tool there is no visible effect. To see its results you need to open the dialogs and tabs where its dimension’s attributes were recorded. This tool is useful for transferring the characteristics of one dimension to another. That is, the Get Dimension Attributes tool is used to record a dimension’s attributes and then the Set Dimension Attributes tool is used to assign them to another dimension.

Set Dimension Attributes

This tool adjusts the attributes of a dimension or leader line according to the options currently selected in the dialogs that are invoked from the **Dimension Attributes** dialog. This dialog can be invoked directly from this tool. The options in the **Dimension Lines**, **Witness Lines**, **Terminators**, and **Dimension Text** tabs are as discussed in section 5.10.6.

The Set Dimension Attributes tool works at the Element and the Compound level. Either the prepick or the postpick method may be used. With the postpick method, the operator is activated first, and then the dimension or leader line whose attributes are to be changed is picked. The operation is executed immediately and the entity is redrawn with its new attributes. When the prepick method is used, any number of dimensions or leader lines are selected first, then the operator is activated, followed by a single click in the graphics window to initiate the attribute changes. Again, the affected elements are redrawn in the graphics window to reflect the new attributes. If elements other than dimensions or leader lines are selected, they are ignored.
5.10.8 Moving dimensions

While non-associated dimensions can be transformed using any of the geometric transformation tools, associated dimensions require the following special tool.

**Reposition Dimension**

This tool is used to reposition dimensions graphically. It moves a dimension along its witness lines and can be applied on associative, non-associative, and mixed dimensions.

The Move tool can also be used to move dimensions or parts of them under certain conditions, as follows:

- It can be used to move complete non-associated dimensions.
- It can be used to move the non-associated parts of mixed dimensions.
- It can be used to move the text of any dimension.
- It can be used to also move the elements a dimension is associated with, when moving the associated dimension. This is done after an affirmative answer to a warning issued by the program.

Figure 5.10.8.1 illustrates the effect of the Move and Reposition Dimension tools on dimensions.

![Figure 5.10.8.1](a) A non-associative dimension that will be moved.  
(b) With topological level at Element and the Move tool the whole dimension is moved.  
(c) With the Reposition Dimension tool the dimension moves along the witness lines.  
An associative dimension could only be moved as in (c).
5.11 Image elements

Image elements are bit-mapped pictures that can be included in drafting drawings. They are the only elements that are not created by a tool in the drafting tool palette but are imported into the drafting environment. They are always rectangular and they remain rectangular when they are moved or scaled.

Image elements can be created through the Import... item in the File menu or with the Paste Image item in the Edit menu. An image element can be created from any supported image file format.

When the Add to Project option in the File Open dialog is off, the image is imported and the element is created in a new window. When the Add to Project option is on, the image element is created in the active window. The Paste Image command can be used to create an image element from any graphics which are stored in the operating system's clipboard in both the modeling and drafting environments.

Image elements can be either linked or unlinked. Linked image elements contain information about the original image that was used to create the image element. This link is used to locate the original file for high resolution display and printing. Linked images also have the advantage that, if the linked image file is changed or updated, the image element can be updated without needing to import it again. If the original file can not be found, form•Z will look for a file with the same name in the directory where the form•Z project file resides. If a file is not found there, it will look in the directory that contains the form•Z application for a file with the same name. Unlinked elements do not have a connection to an external file and rely on a copy of the data stored in the project file for high resolution display and printing.

Image elements created by using the Import... method are by default linked to the original file that they are created from. Images added to a project using the Paste Image method have no associated image file and are by default unlinked.

The entire image data may be stored directly in the project file in which it is placed. When the image is stored in the file, the image element will use much more disk space and memory, but high resolution displays and printing will be faster. When the image is not stored in the file, it uses significantly less memory and disk space, however, it must be linked to an external file for proper high resolution display and printing. Images that are not stored in the project file and are not linked, will not display or print properly in high resolution.
When images are added to a project file, whether they are stored in the project file or not is controlled by the **Store New Image Elements In Project** option in the **Preferences** dialog. When this option is on, which is the default, images added to the project are stored directly in the project. If the image requires more than 256 K of space to keep it in the file, a warning is issued offering the option to store the image as a linked image or store it directly. It is recommended that images larger than 256 K not be stored directly in the file unless you have a machine with a significant amount of memory and disk space.

Image elements are displayed on the screen and printed in either **bounding box**, **preview**, or **high resolution** display methods. These three display types represent increasing quality of the image display at the expense of speed and memory. Bounding box is the simplest and fastest display method and only shows the image placement and size. With this option, each image element is displayed as a rectangle which identifies the bounding box of the image with diagonal lines connecting the corners of the rectangle. The preview display shows the image in a low resolution bit-map that is stored in the project file at the time the image element is created. The preview image for each image element uses approximately the same amount of memory, therefore, the preview of a small image will likely look more accurate than the preview of a large image (small and large referring to the memory or disk space needed to represent the image, not the geometric size at which it is placed in the project). The high resolution display is the most accurate, but is the slowest. Images that are stored in the project will display in high resolution quickly because all of the image data is in memory and it is simply drawn to the screen. Images that are not stored in the project file, use the linked image file and are displayed by reading the image from the disk. This is slower because the disk is slower than memory, however, it conserves significant amounts of memory for large image files. If an image is neither stored in the project nor linked to an image file, it can not be shown in high resolution. Such images are automatically drawn in preview mode when a high resolution display is selected.

The appearance of the image elements in the drafting windows is preview by default. It can be changed using the **Detail** popup menu in the **Image Element Options** section of the **Drafting Display Options** dialog. The quality of printed output of image elements is high resolution by default and can be changed using the **Image Detail** popup menu in the **Plot/Print Setup** dialog.
5.12 Querying and measuring

The tools in the 10th row, right side of the drafting tool bar are used for accessing information about drafting elements and their parts as well as the ability to measure lengths and distances. They are:

- Query
- Query Attributes
- Measure
5.12.1 Querying drafting elements

Query

This tool gives you access to information about elements and their parts. You can query an element’s type, typology, and geometry; all of which can also be edited at the time of the query.

As in modeling, the Query tool can be used in postpick or prepick mode. When the prepick method is used to preselect a number of drafting entities, the system parades through these entities, one at a time, until the list of preselected entities is exhausted, or the Cancel button is selected.

In general, the Query tool can be applied at the Point, Segment, Element, or Compound level. However, whether an element can be queried at all these levels depends on its type.

- Compound elements can be queried at all four levels.
- Polyline elements and leader lines can be queried at the Element, Segment, and Point levels only. Querying them at the Compound level is equivalent to querying them at the Element level.
- Arc elements (including circles, ellipses, and open arcs), dimensions, text, image element, and symbols can only be queried at the Element level.
- Area elements can only be queried at the Segment and Point levels.

When querying a compound element at the Compound level, the information is presented through the Query Compound Element dialog, shown in Figure 5.12.1.1. It contains information about the # Of Elements, the # Of Points, and the # Of Segments In Compound. It contains no editable text fields, thus reported characteristics of a compound cannot be changed through this dialog.

When querying an element (including a member of a compound) at the Element level, the information is presented through the Query Element dialog, as shown in Figure 5.12.1.2. The text fields within the dialog cannot be edited. The popup menu next to the Surface Area option allows you to see this measurement in either square inches or square feet.

Query Attributes...: Clicking on this button invokes the Query Element Attributes dialog discussed in subsection 5.12.3.
When querying certain types of drafting elements, namely points, rectangles, arcs, dimensions, leader lines, text, image elements, and symbols, the **Query Element** dialog also contains a button labeled **Edit...** Clicking on the **Edit...** button invokes a supplementary dialog that contains editable information about the element being queried. The **Rectangle Parameters** dialog invoked for rectangles, and the **Circle/Ellipse Parameters** dialog invoked for arc elements, are shown as examples in Figures 5.12.1.3, and 5.12.1.4. Querying image elements is discussed in the next subsection. For the dimensions, leader lines, text, and symbols the same dialogs that are used for generating or editing them are invoked and can be used to revise their parameters.

When querying at the Segment level, the **Query Segment** dialog shown in Figure 5.12.1.5 is invoked. It contains the coordinate positions of the two end points of the segment in editable text fields whose content can be changed to revise the geometric definition of a segment. It also contains information about the length of the segment and a button labeled **Query Element...** which allows you to access the **Query Element** dialog to obtain more information about the element to which the queried segment belongs.

When querying at the Point level, the **Query Point** dialog shown in Figure 5.12.1.6 is invoked. It contains the coordinates of the point in editable fields and a **Query Element...** button.
5.12.2 Querying image elements

When you click on an image element with the Query tool and then click on Edit... when the Query El-

tement dialog appears, the Image Parameters dialog (Figure 5.12.2.1) is invoked. This dialog is used to view and change the geometric parameters of an image element or its link.

**Image Position And Size:** This section contains the location of the lower right corner of the image, its width, and height.

**Image Info:** This box contains information about the image, as follows:

- **File Name:** This is the name of the original image file. If the image element was created using Paste Image... (Edit menu), this field shows “<Not Linked>”. If the image is linked, the size of the image file on the disk is shown in parenthesis after the name. If the image is not linked, “<Not Linked>” is displayed.
- **Type:** Type of the original image file (TIFF, Targa, JPEG, etc.).
- **Size:** Amount of memory/disk space that this image element uses.
- **Resolution:** Pixel dimensions of the original image.
- **Modified:** For linked images, this shows the time and date when the original image file was last modified. “<Not Linked>” is displayed for images that are not linked.
- **Location:** This shows the full path of the location of the original image. “<Not Linked>” is displayed for images that are not linked.

The items at the bottom of the dialog are used for managing the image element.

- **View...:** This button invokes the form-Z image viewer allowing inspection of the image at its full resolution. If the image is not linked and not stored directly in the file, this option is dimmed.
- **Load...:** This button loads a new image into the image element. Clicking on it invokes the Open File dialog where an image file can be selected. The old image is discarded and the new image is imported. This does not change any of the geometric characteristics of the image element. The Image Info section is updated to show the updated information for the image file.
- **Update:** This button is used to reload the image data from the linked image file. It is only available when the image is linked and the modification date is later than the date the image element was created or last updated. Otherwise this button is dimmed.
- **Unlink:** This button breaks the link between the image element and the image file. If the image is not stored in the file, you are prompted to load the image into the file before breaking the link.
- **Store Image In Project:** When on, this option indicates that the image is stored directly in the project file. Deselecting it unloads the image and releases the memory/disk space it occupies. Unloading an unlinked image element leaves it without a detailed representation and thus it cannot be displayed or printed in high resolution. A warning is presented before unloading an unlinked image. Selecting this item causes the image to be loaded from the original file. If the image is not linked, it cannot be loaded into the project file and this item is unavailable.

*Drafting • Querying and measuring*
5.12.3 Querying element attributes

This tool offers information about an element and its attributes. The Query Attributes tool always functions on the element level, regardless of which topological level is selected when clicking on a drafting element with it.

When you click on an element with this tool, the **Query Element Attributes** dialog (Figure 5.12.3.1) is invoked. The dialog consists of four popup menus which allow you to change the layer, color, line weight, and line style of the element being queried. It also includes information on whether or not the element is dimensioned or hatched. Clicking on its **Query Element** button invokes the **Query Element** dialog discussed in subsection 5.12.1.

![Query Element Attributes dialog](image)

*Figure 5.12.3.1:*
The **Query Element Attributes** dialog.
5.12.4 Measuring lengths and distances

This tool is used to measure lengths of elements, distances between points, and angles between segments.

Double clicking on this tool invokes the **Measure Options** dialog (Figure 5.12.4.1). One of three types of measurement operations are available within this dialog.

**Length Of Vector/Boundary Line**: When this option is selected, the Measure tool can be used to measure the length of an open or closed shape or the length of a segment. With topological level set to Element and the Measure tool active, click on a shape. The message dialog shown in Figure 5.12.4.2 is invoked and shows you the length of the boundary of a shape, which can be open or closed. This shape can be a polyline, a rectangle, or an arc.

Set the topological level at Segment and repeat the operation by clicking on the segment of a shape. The message dialog shown in Figure 5.12.4.3 is invoked and reports the length of the segment you selected. Recall that segments can be selected from rectangles and polylines, including polyarcs, but not arcs. Of the latter you can only measure their complete length with topological level set to Element.

Note that the Query tool with the Length option can only be used at the Segment and Element topological levels. All the other levels do not produce any results. Lengths of parts of compound elements can be measured by setting the topological level to Element.

**Interactive Point To Point**: When this option is on, the Measure tool can be used to measure distances between two points, which can be anywhere on the reference plane or on an element. Any of the element snaps can be used to accurately place the cursor at the point you measure.

After selecting the **Interactive Point To Point** option, with the Query tool active click on the first point. As soon as you do, a line is rubber banded, anchored at your first point, and follows the motion of the mouse. Simultaneously, a value is displayed in the Prompts palette, after the prompt “Distance:” (see Figure 5.12.4.4).

As you move the mouse, this value is continuously updated and gives you the length between your first point and the current position of the mouse. As soon as you click again, the distance is permanently recorded in the Prompts palette and no other message is issued. Note that this option ignores the topological level and always measures distances between points.
**Point/Segment To Point/Segment:** When this option is on, the Measure tool can be used for measuring the distance between two points, two segments, or a point and a segment. Note that the distance between two segments is only defined when the segments are parallel. If two non-parallel segments are selected, an error message will be issued.

After selecting the above option, with the Measure tool active, click on a point or a segment and then on another point or segment. Depending on what combination of points or segments you clicked on one of the message dialogs shown in Figure 5.12.4.5 will be posted informing you about the distance. Note that this operation requires no specific topological level to be set. It is able to recognize whether you click on a point or a segment and to calculate the distance. However, note that, contrary to the **Interactive Point To Point** operation, the point you select for this option has to be on an element. It cannot simply be on the reference plane and away from an element.

**Angle Between Segments:** When this option is selected, the Measure tool is used to measure the angle between two segments. The angle is measured from the first segment that is selected to the second. When more than two segments are selected, the angles are measured in pairs (first to second, third to fourth etc.). The message dialog shown in Figure 5.12.4.6 is posted with the angle.

The most common way to measure angles is by selecting two segments, however, other topological levels will work as well as follows:

- Picking a point that is shared by exactly two adjacent segments of the same element measures the angle between the segments. The angular direction is determined by the direction of the first segment.

- Picking an element that contains exactly two adjacent segments measures the angle between the segments. The first segment is determined by the direction of the segments.

- Picking an element that contains exactly one segment is treated as selecting the segment at the segment level.
5.13 Geometric transformations

This section discusses the modifiers that determine whether a transformation is applied to an element itself or one or more copies. They are:

- Self
- One Copy
- Continuous Copy
- Repeat Copy
- Multi-Copy

It also discusses the geometric transformations namely:

- Translate (Move)
- Rotate
- Independent Scale
- Uniform Scale
- Reflect (Mirror)

The geometric transformations (move, rotate, scale, and mirror) affect the geometry of a point, a segment, or an entire element. They work in conjunction with both the Topological Level and the Self/Copy modifiers. The Self/Copy modifiers only affect operations that are applied at the Element level. They have no effect when they are applied at the Point or the Segment level.

The geometric transformations of the drafting environment are very similar to their modeling counterparts. However, in drafting, they are less uniform. This section covers the generally applicable processes first, then the distinct behavior of each type of element is discussed in its own subsection. Given the similarities with the geometric transformations of the modeling environment, frequent references to the corresponding modeling sections are included.
5.13.1 The Self/Copy modifiers

The Self/Copy modifiers control whether a geometric transformation is applied to the original element, or to one or more copies.

These tools function identically to their counterparts in the modeling environment (subsection 4.23.1). When the Self mode is active, the original element is transformed. When the One Copy mode is selected, the transformation operation is applied to a single copy of the original element. With Continuous Copy on a new copy is made for each click of the mouse, and the copying process is terminated by a double click of the mouse. Repeat Copy also creates a new copy for each click of the mouse, but the transformation parameters are determined by the first mouse click, and are repeatedly applied to all the subsequent copies, until the mouse is double clicked. Multi-Copy generates a number of repetitively transformed copies. The number of copies is determined by a parameter set in the Copy Options dialog (see subsection 4.23.1). The Self/Copy modifiers have no effect on the area element, which is always transformed in Self mode.
5.13.2 The geometric transformation tools

The drafting geometric transformations parallel the first five tools of the modeling geometric transformations and they behave in a very similar fashion. However, there are also variations that are dictated by the specific character of each type of drafting element. This subsection covers the general procedures by which the drafting geometric transformations are applied. The specific behavior of each type of drafting element, for each geometric transformation, is discussed in subsequent subsections.

**Translate (Move)**

This tool moves a point, a segment, or an element to a new position, in a manner similar to that of the modeling environment (see subsection 4.23.2). All drafting elements can be moved at the Element level. At the Point and Segment levels, the Move operator can only be applied to polylines, including the rectangles, leader lines, and image elements.

**Rotate**

**Independent Scale**

**Uniform Scale**

**Reflect (Mirror)**

These tools apply the respective geometric transformation to a point, a segment, or a complete element, in a manner similar to that of the modeling environment (see subsection 4.23.2). All drafting elements, except for the dimensions and the image elements, can be rotated, scaled, and reflected at the Element level. At the Point and Segment levels, the Rotate, Scale, and Reflect tools can only be applied to polylines, excluding the rectangles, and leader lines.

The drafting geometric transformations can be executed using either the prepick or the postpick method. The prepick method offers the ability to select a group of entities and transform them all through the execution of a single tool. Due to the diverse character of the different types of drafting elements, some restrictions apply relative to what types of elements can be mixed when they are prepicked. These restrictions are discussed in the remaining subsections.

All the geometric transformations are executed dynamically and the shapes that are affected are rubber banded as they are transformed. For all the transformation tools, the rubber banding process is activated by the next to the last mouse click.
5.13.3 Transforming polylines

When geometric transformations are applied to polylines they behave very similarly to the surface objects of the modeling environment. Any of their points or segments can be freely moved, rotated, scaled, or reflected, or they can be transformed as complete elements. At the Element level, the polyline itself can be transformed, or one or more copies can be made and transformed, by setting the Self/Copy mode where appropriate. Points and segments can only be transformed in Self mode. Portions of polylines can also be freely transformed, after they have been clipped, through the use of the Area element and the Clip To Area option selected, in the Area Options dialog.

The behavior of the polyline when a Move tool is applied, is illustrated in Figure 5.13.3.1. The other geometric transformations work in a similar fashion. Note that the postpick method is used in the description of the example. An open polyline is drawn, as shown in Figure 5.13.3.1(a). A point is moved by setting the topological level to Point and, with the Move tool active, clicking the mouse on b1, then b2 (Figure 5.13.3.1(b)). A segment is moved by setting the topological level to Segment and clicking on points c1 and c2, while the Move tool is active (Figure 5.13.3.1(c)). In Figure 5.13.3.1(d), two copies of the polyline are made and are repetitively moved by setting the topological level to Element, and the Self/Copy mode at Repeat Copy. With the Move tool active, the mouse is clicked on d1, then on d2, then anywhere on the screen, followed by a double click which completes the copying process.

The last part of the example (Figure 5.13.3.1(e)) illustrates the clipping operation, as applied to a group of polylines. A rectangular area element is generated where the dashed line shape is shown. With the topological level at Area, the Self/Copy mode at Self, the Clip To Area and the Inside options selected (Area Options dialog), and the Move tool active, the mouse is clicked on point e1, then on e2. The first mouse click picks the elements which cross the boundary of the area and breaks them at the points where they intersect that boundary. The second mouse click moves the portions of the polylines that are inside the area to their new position.

![Figure 5.13.3.1](image)

*Figure 5.13.3.1:* (a) A polyline as initially drawn. (b) A point is moved. (c) A segment is moved. (d) Two more repetitive copies of the polyline are made. (e) Portions of the polylines are clipped and moved, through the use of the Area element.
5.13.4 Transforming rectangles

The rectangle is a special type of a polyline and, when it is geometrically transformed, its behavior is constrained in a way such that its orthogonal shape is preserved at all times. Segments can only be moved in a direction perpendicular to their original orientation. When points are moved, they carry with them the two segments that contain them, in a manner that retains their orthogonality. When a rectangle is moved as a complete element, it behaves as any polyline. Such a move can be applied at the original element or to one or more copies of the element. When clipped, the rectangle is transformed to a common polyline. The response of the rectangle to the Move tool is illustrated in Figure 5.13.4.1.

A 3 Point rectangle is generated in a rotated position as shown in Figure 5.13.4.1(a). With the topological level at Point and the Move tool active, a click on point b1 followed by a click on point b2 moves one of its points and transforms the rectangle, as shown in Figure 5.13.4.1(b). The move of a segment is shown in Figure 5.13.4.1(c). With the topological level at Segment and the Move tool active, the mouse is clicked on point c1, then c2. Figure 5.13.4.1(d) illustrates the generation of six copies, positioned at equal distances. This operation is executed by setting the topological level to Element, the Self/Copy mode to Multi-Copy, and clicking on points d1 and d2 while the Move tool is active. The example was executed with the default settings (Even Increment and # Of Copies = 6) of the Copy Options dialog. Figure 5.13.4.1(e) illustrates a move applied to a clipped portion of the seven rectangles produced earlier. An elliptical area element is generated as shown by the dashed line. With the topological level at Area, the Self/Copy mode at Copy, and the Move operator active, the mouse is clicked on point e1, followed by a click on point e2. The result is a copy of the portions of the shapes that are inside the boundary of the area. The copy is moved to the position corresponding to the second click (e2). After the execution of this operation, the original shapes appear to have remained as they were before the operation, which is not true. Before the application of the clipping operation, the rectangles were exploded and were transformed to common polylines. Then they were clipped at the points where they intersect the elliptical area element.

Figure 5.13.4.1: (a) A rotated rectangle is generated. (b) One of its points is moved. (c) One of its segments is moved. (d) Six more copies are made using the Multi-Copy mode. (e) Portions of the rectangles are clipped, copied and moved, using an elliptic area element.

Drafting • Geometric transformations
With the exception of the Move tool, none of the other geometric transformations can be applied to individual points or segments of rectangles. They can all be applied at the Element level, as shown in Figure 5.13.4.2.

![Geometric transformations applied to a rectangle: (a) Rotation, (b) Independent Scale, (c) Uniform Scale, and (d) Reflection.](image)

The Rotation, Uniform Scale, and Reflection tools apply the same transformation parameters to all the points of a shape in a uniform manner, which preserves the original shape. This is also true for the rectangular shapes. In contrast, the Independent Scale tool, which applies distinct scaling factors to the x and y directions, does not normally preserve the original shape. However, as implemented in form•Z, Independent Scale is constrained when it is applied to a rectangle, whose orthogonal shape is preserved at all times. This distinction is illustrated in Figure 5.13.4.3.

A rectangle is generated and then rotated as shown on the left of Figure 5.13.4.3(a). A copy of the rectangle is made to its right and is then exploded. Recall that the Explode tool transforms the rectangle to a common polyline representation. The two shapes are prepicked. With the topological level at Element and the Self/Copy mode at Self, the Independent Scale operator is activated and the mouse is clicked on points 1, 2, and 3, in that order. This results in the scaled shapes shown in Figure 5.13.4.3(b). Observe how differently the scale has affected the two shapes. The orthogonal shape of the rectangle to the left has been preserved. The orthogonal shape of the polyline to the right has been disturbed.

![Applying the Independent Scale to a rectangle (R) and a rectangular polyline (P), by clicking on points 1, 2, and 3: (a) The original shapes and (b) the shapes after the scaling operation.](image)
5.13.5 Transforming arcs

Elements that are internally represented as arcs are also constrained when geometric transformations are applied to them. Recall that the arc elements are distinct from the polyarcs and are parametric entities. They can only be moved, rotated, scaled, or reflected at the Element level. They contain no points or segments that may be transformed individually. When they are clipped by an area element, they are exploded to common polylines and they subsequently behave as any other polyline. The response of the arc element to the Move tool is illustrated in Figure 5.13.5.1. The other geometric transformations behave in a very similar fashion and they all preserve the circular shapes of the arc elements. The Independent Scale tool may transform a circle to an ellipse, and vice versa.

A complete circle is first created, as shown in Figure 5.13.5.1(a). An attempt to move a point, by setting the topological level to Point and clicking the mouse on point b1 fails right from the beginning, since the system finds no point to pick (Figure 5.13.5.1(b)). The generation of three more, symmetrically arranged copies is illustrated in Figure 5.13.5.1(c). With topological level at Element, Self/Copy mode at Continuous Copy, and the Move tool active, the mouse is clicked on points c1, c2, c3, and c4, and is then double clicked to terminate the copying process. Note that the same result could have been derived using the Rotate tool.

A clipping operation is illustrated in Figure 5.13.5.1(d). A rectangular area is generated as shown by the dashed line. With topological level at Area, the Self/Copy mode at Self, and the Move tool operator active, the mouse is clicked on point d1, then d2. Now that the original arcs have been transformed to polylines, their points and segments can be freely moved. This is illustrated in Figure 5.13.5.1(e) where two points and two segments have been moved in the usual manner.

![Figure 5.13.5.1](image)

**Figure 5.13.5.1:** (a) A circle is generated as an arc element. (b) An attempt to move a point fails. (c) Three more copies are made, resulting in a symmetric arrangement of four arc elements. (d) A square area is clipped from the center of the composition and is moved to a new location. (e) After clipping, which transformed the arcs to polylines, two points and two segments are moved.
5.13.6 Transforming text

Text elements can only be geometrically transformed at the element level. The Independent Scale tool cannot be applied to a text element. When it is, an error message is posted. A text element cannot be clipped through the use of the area element, since text elements cannot be exploded. The area element can be used to select a text element.

Moving text at the Element level simply repositions the text elsewhere in the drafting space (Figure 5.13.6.1(a)). It does not affect its orientation or its size. Text can also be freely rotated (Figure 5.13.6.1(b)) and reflected (Figure 5.13.6.1(c)). These operations change the orientation of the text. They do not affect the size of the text. Also note that the reflection maintains the left to right direction of the characters. The Uniform Scale, the only scaling operation that can be applied to a text element, affects the size of the characters in the text and possibly the location of the text (Figure 5.13.6.1(d)).

![Figure 5.13.6.1: (a) Moving, (b) rotating, (c) reflecting, and (d) scaling text.](image_url)

While text is the most restrictive type of drafting element, because it cannot be exploded to become a regular polyline, recall that the option to import text from modeling as polylines is always available.
5.13.7 Transforming dimensions

Dimensions, but not parts of dimensions, may be transformed using one of the general geometric transformation tools and also their own special transformation tool called Reposition Dimension. How exactly they behave depends on whether they are associated or non-associated. Also recall that, when dimensions of either type are exploded, except for their text, they reduce to common polylines and can be transformed as any other polyline at any topological level. The following discussion does not apply to exploded dimensions.

Transforming associated dimensions

Only the Reposition Dimension tool can be used to move the dimension line and text along its witness lines. The tool can also be used to reposition vertical/horizontal dimensions from vertical to horizontal and vice versa.

When multiple dimensions have been linked to become compound entities, they can be repositioned as a single unit by setting the topological level to Element. Examples are shown in Figure 5.13.7.2. Example of repositioning nonlinear dimensions are shown in Figure 5.13.7.1. The common geometric transformations can only be used to move the text part of an associated dimension.

The complete associated dimension can be transformed with the Move, Rotate, Scale, and Mirror tools by transforming the elements with which it is associated. If you try to transform the dimension directly, a warning message will ask you if you wish to also transform the associated elements. Depending on you answer, it will either proceed with the transformation or cancel it.

Figure 5.13.7.1: Repositioning nonlinear dimensions.
Transforming non-associated dimensions

Non-associated dimensions can be transformed with the special Reposition Dimension tool the same way the associated dimensions can. In addition, all the common transformations can be applied to them, but only at the Element or Compound topological level.

Transforming dimension text

As mentioned above, the text part of a dimension can be transformed independently if the translate tool is applied in postpick mode and the mouse is clicked on the text. This will move the text away from its default position. Once this has been done, subsequent transformations of the dimension will NOT include the dimension text anymore.

Figure 5.13.7.2: Repositioning linked dimensions at the Compound level.
(a) A horizontal chain is moved to the other side of the dimensioned elements.
(b) A dynamic horizontal/vertical stack is repositioned from the top to the right of the dimensioned element.
(c) A uniform parallel chain is moved to the inside of the dimensioned element.
5.13.8 Transforming leader lines

The leader line is the drafting element whose behavior is the closest to that of the polyline. When none of its endpoints are associated with points of other drafting elements, it can be freely moved, rotated, scaled, or reflected, at any topological level. At the Element level, any of the Self/Copy modifiers can be used.

When either one or both of its endpoints are attached to other elements, they are restricted from moving. That is, when a geometric transformation is applied at the Element level, it affects all the points of a leader line, except those of its endpoints that are associated with other elements. At the Segment level, if the segment contains an associated point, that point is not affected by the geometric transformation. At the Point level, associated points cannot be transformed.

![Figure 5.13.8.1](image)

*Figure 5.13.8.1:* Transforming leader lines (1) without and (2) with associated endpoints. (a) The leader line in its original position. (b) Moving, (c) rotating, (d) scaling, and (e) reflecting it.

The associated points of a leader line can be indirectly moved by moving the points to which they are attached. This is true for all the geometric transformations. The area element can be used to move, rotate, scale, or reflect a leader line. The area element can also be used to clip it. When clipped, its parts are transformed to individual polylines and it loses its associativity.
5.13.9 Transforming hatch patterns

Hatch patterns can only be transformed indirectly, by transforming the bounding shapes to which they have been attached. Hatch patterns can be attached to polylines, including rectangles, and to closed arcs. The geometric transformations work with these types of drafting elements as discussed in sections 5.13.3, 5.13.4, and 5.13.5, respectively. When a hatch pattern is attached to an element that is subsequently transformed, the hatch pattern is regenerated and adjusted to the new position or shape of the element. This is true for both types of hatch patterns (bit-maps and vector lines). Examples of geometric transformations applied to the bounding shapes of hatch patterns are shown in Figure 5.13.9.1.

![Figure 5.13.9.1: Transforming the bounding shapes of hatch patterns.](image)

(a) Moving a point of the bounding shape of a vector pattern.
(b) Moving the complete bounding shape of a bit-mapped pattern.

Hatch patterns can also be indirectly affected by the area element. The area cannot be used to clip a hatch pattern. However, it can be used to clip the bounding shape of a hatch pattern. When such a clipping operation breaks the closure of a bounding shape, the hatch pattern is deleted, as illustrated in Figure 5.13.9.2.

![Figure 5.13.9.2: (a) Two hatched shapes, which (b) lose their hatch patterns when a clipping operation transforms them to open shapes.](image)
5.13.10 Transforming symbols and image elements

All the geometric transformations (Move, Rotate, Scale, and Mirror) can be freely applied to a symbol, but only at the Element level. The points and the segments of a symbol cannot be independently picked or transformed.

A geometric transformation can be applied to the instance of a symbol when it is first placed in the drafting space. This is done through the numeric parameters in the Symbol Place Options dialog when the symbol is placed (section 5.8). The interactively executed transformations applied by using one of the Transformation tools are applied in addition to whatever transformations were applied when the symbol was placed. Any number of transformations can be freely applied in this manner.

The area element can also be used to transform complete symbols, as well as to clip them. When clipped, they are exploded and transformed to polylines, which can subsequently be transformed as common polyline elements.

Image elements can only be moved and scaled. They cannot be rotated or mirrored. When they are scaled and moved they behave like rectangles.
5.14 Relative geometric transformations

This section discusses the following three tools:

- Align/Distribute
- Extend
- Place

These tools transform the geometry of one or more elements relative to another element. These are three of the five relative transformations available in modeling. Even though they are restricted to only two dimensions, as is normal for drafting, they offer some significant functions for rearranging and composing shapes.
5.14.1 Alignment and distribution

Elements scattered in drafting’s 2D space can be aligned along the X and Y directions and can be distributed at equal distances using the Align/Distribute tool.

**Align/Distribute**

This tool is used to align and/or distribute a number of preselected elements. The exact type of alignment and/or distribution is set in the *Align/Distribute Options* dialog (Figure 5.14.1.1) which is invoked directly from the tool.

This dialog contains a preview area, which, as in modeling, can be used in two different ways.

- If the dialog is invoked from the tool, three non-existing elements are used to preview the effect of the option selected in the dialog.

- If a number of elements are selected at the time the dialog is invoked, these elements are displayed and the effect of the selected operation previewed.

The Align/Distribute tool makes more sense when applied in prepick mode. That is, after selecting a number of elements with the Pick tool, you activate the Align/Distribute tool and click somewhere in the project window. The *Align/Distribute Options* dialog is invoked and, after you set the options you desire and you click on *OK*, the operations are applied to the elements in the project.

*Figure 5.14.1.1:* The drafting *Align/Distribute Options* dialog.
5.14.2 Extending lines and arcs

Extend Line/Arc

This tool works with polylines (excluding the rectangles) and arcs only.

When applied to a polyline, this tool extends or shrinks the selected segment. The segment that is extended or shrunk, retains its original orientation (slope). However, the shape of the segment to which it may be connected, is affected, in a fashion similar to that of the Trim tool. When applied to polylines, the Extend tool is always executed at the Segment level. With the postpick method, the topological level is ignored and defaults to Segment. With the prepick method, polylines that are picked at a level other than Segment are ignored.

When applied to an open arc, one end of the arc is extended or shrunk, following a circular path, which is consistent with the parameters of the arc. The Extend tool cannot be applied to closed arcs. When applied to arcs, it is always executed at the Element level. With the postpick method, the topological level is ignored and defaults to Element. With the prepick method, attempts to pick arcs at a level other than Element fail, since arcs can only be picked at the Element level.

Examples of the Extend Line tool applied to both segments of polylines and arcs are illustrated in Figure 5.14.2.1.

![Figure 5.14.2.1](image)

**Figure 5.14.2.1:** (a) Extending one end of a single segment polyline. 
(b) Shrinking one of the open ends of a polyline does not affect any other segment. 
(c) Extending a connected segment of a polyline affects the shape of the adjacent segment. 
(d) Shrinking a segment of a closed polyline affects the shape of its adjacent segment. 
(e) Extending the end of an arc.
The Extend Line/Arc tool can be executed in any one of three optional modes: **Relative**, **Absolute**, and **Numeric**. These options are selected from the **Extend Options** dialog shown in Figure 5.14.2.2. It is directly invoked from the Extend Line/Arc tool.

In the **Relative** mode, which is the default, the length of the extension is given by the distance that the mouse travels. In the **Absolute** mode the elements are extended to the point defined by the position of the mouse cursor. In both of these modes, the extension is dynamic and the affected elements are rubber banded, allowing you to inspect the results before the final click of the mouse, which completes the execution of the operation. The different results derived by the **Relative** and **Absolute** modes are more apparent when the prepick method is used and the Extend tool is applied to a number of elements simultaneously.

In the **Absolute** mode, the point to which a segment is extended is determined by drawing a line from the position of the mouse perpendicular to the segment and calculating the point of intersection. It is not necessary to click the mouse on the segment, neither is it possible to do so when extending two or more segments simultaneously. For example, given three prepicked disjoint segments, as shown in Figure 5.14.2.3(a), there is no way to position the mouse on all three segments at the same time. After clicking the mouse on point \( p \), it is projected to points \( p_1 \), \( p_2 \), and \( p_3 \). These are the points to which the segments are extended or shrunk.

The **Numeric** mode extends the elements to a predefined distance. For extending the segments of polylines, the predefined distance is expressed in linear units of measurement. For extending arcs, the distance is expressed in degrees. The values for these distances are entered in the **Linear Distance** and **Arc Angle** fields, respectively. Preset default values are 2'-0" (or 0.50 m in metric) for linear and 45° for angular distances. The **Numeric** mode does not rubber band the elements that are being extended, but rather executes the operation immediately.
5.14.3 Placing elements on elements

This tool allows you to place 2D shapes on other 2D shapes. It can be used in both prepick and postpick modes and it is always executed at the element level, since it only places elements on elements. The shape that is placed is called the source shape and can be open or closed. The polyline where the source shape is placed is called the placement line and can also be open or closed.

The options for placement are selected in the Place Options dialog shown in Figure 5.14.3.1. Note that these options are similar to those found in the modeling Place Options dialog (see section 4.24.4), however, options that are inherently 3D in nature are not available. At the same time, there are a few options that are specific to drafting, as follows:

Orientation: Perpendicular To Line, Parallel To Line: These options are as in modeling (see subsection 4.24.4).

Aligned To World Plane: When this option is on, the source shape is placed relative to the axes of the drafting space. That is, its length and height are positioned parallel to the X and Y axes, respectively. Examples of the Orientation options are shown in Figures 5.14.3.2.
Alignment: Centroid, Middle Of Open Ends, World Origin: These options are as in modeling and are illustrated in Figure 5.14.3.3.

Selection Point: This option in the Alignment section is unique to drafting. When selected, the point on the source shape where you click to select it becomes the alignment point and is placed on the placement line. As it is placed, the source shape preserves its original orientation. This is illustrated in Figure 5.14.3.4.

The Multi-Placement options are similar to those in modeling, but have their own flavor, as they are executed exclusively in a 2D space. They are illustrated in Figures 5.14.3.5.

Figure 5.14.3.4: Placing a shape on polyline using the Selection Point option.

Figure 5.14.3.5: Placing End-To-End:
(a) A triangle to be palced on shown shapes.
(b) Use Increment, (c) Use # Of (=6),
(d) At Point Bisectors, (e) On Midpoints.
5.15 Element attributes

When new elements are created they are assigned the currently active drafting attributes. These attributes are the line weight, the line type, the layer, and the color. After an element has been created, these attributes can be changed through the use of the attribute setting tools, discussed in this section. They are:

- Line Style
- Line Weight
- Color
- Set All Attributes
- Ghost
- Unghost
- Layer

The Hatch tool may also be considered an attribute setting tool. However, given its special character, the generation of hatch patterns is discussed in a separate section.

All attributes cannot be assigned to all types of drafting elements. Which attributes apply to which elements is summarized in the table of Figure 5.15.0.1. Note that no attribute can be assigned to the area element, which is consistent with its function.

<table>
<thead>
<tr>
<th></th>
<th>Line Type</th>
<th>Line Weight</th>
<th>Color</th>
<th>Layer</th>
<th>Ghost</th>
<th>Hatch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polylines</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Rectangles</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Arcs</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Text</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Dimensions</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Leader Lines</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Symbols</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Area</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Image border</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
</tbody>
</table>

*Figure 5.15.0.1:* Which attributes apply to which elements.

The attributes assigning tools work with both the prepick and postpick selection methods. With postpick, the tool is selected first, then the elements are selected, one at a time. The attribute assignment is executed right away, the element is redrawn, and the effect of the new attribute is visible immediately. When prepick is used, any number of elements are selected first with the Pick tool, then the desired operator is activated, followed by a click in the project window.

In the drafting environment, all attributes are assigned at the Element level. The topological level has no effect when postpick is used. With prepick, all entities should be picked at the Element level. Entities picked at another level are ignored when an attribute assignment tool is applied.
5.15.1 Setting and assigning line styles

This tool changes the line style of the picked element to the active style. This is a technique by which you define, in addition to continuous lines, a variety of dashed lines consisting of pieces of lines with blanks between them. form-Z offers eight predefined styles and also allows you to replace them or add up to 127 styles.

A line style is an element level attribute. Line styles have default names that may be changed. Line styles within one project must have unique names. One line style is always active. It is automatically assigned to newly generated elements.

The line styles are displayed in the Line Styles palette shown in Figure 5.15.1.1. When the default preferences are used, this palette appears on the screen when launching the program and a drafting window is opened. If it does not appear, it can be opened by selecting the Line Styles item in the Palettes menu.

The Line Styles palette works as all the other palettes, and can be resized, scrolled, hidden, and closed. It consists of a horizontal title bar, a header bar, two viewing icons, and two columns. The first displays the status of a style: a check mark (✓) indicates that a style is active. The second column displays the line itself, either graphically or by its name (Figure 5.15.1.1(a) and (b)). The format used is set by the viewing icon selected from the palette’s lower left corner.

Clicking in the header bar of the Line Styles palette, or clicking in the Line Styles column while pressing the option key on the Macintosh, or ctrl+shift on Windows, invokes the Line Styles dialog, which can also be invoked from the Options menu.

The Line Styles dialog, shown in Figure 5.15.1.2, consists of two areas: the line style list on the left, which is essentially a copy of the Line Styles palette, and a number of button commands that allow you to execute a variety of operations. These are the same operations that are available in the dialogs that support other palettes.
Clicking on the Edit... button in the Line Styles dialog invokes the Line Style Editor dialog (Figure 5.15.1.3). It can also be invoked from the Line Styles palette by double clicking on a line style. It offers the tools for defining the parameters of a new line style, or for editing those of a previously defined style, through either graphic or numeric input. The dialog also contains a preview where the currently defined style may be inspected.

There are basically two types of lines: **solid**, which are continuous lines, and **dashed lines**. Either type can be selected from the Line Style Editor dialog.

The dashed lines can be designed in a variety of ways. They consist of two types of components: solid pieces of lines, called **dashes**, and **gaps**, which are blank spaces between the dashes. There is always one (and only one) gap between the dashes, and vice versa. The dashes and the gaps can be of the same or different lengths, and can be arranged in any order. Dashed line styles are defined through a **pattern** or **module**, which is repeated to produce the complete line when the line is displayed on the screen or printed. Defining a line style essentially consists of defining the pattern of the style. A style pattern may be of any length appropriate for the description of its composition. A style pattern must start with a dash and end with a gap. If the design entered by a user does not satisfy this condition, an adjustment is made.

In form-Z, a dashed line pattern may consists of at most 32 components, which are made up of equal numbers of dashes and gaps. In the EPS and Illustrator formats there is a limit of 11, and in the DXF format there is a limit of 12 components. form-Z line styles exceeding these limits default to a solid line when exported to an EPS, Illustrator, or DXF file. Consequently, it is advisable to limit the number of components used in a dashed style definition to 10, when it is to be exported to EPS, Illustrator, or DXF.

**Name**: A name of at most 31 characters for the line style is entered in this field. When a new style is created, a name is initially defaulted by the system and can be changed through this field.

**Solid Line**: When this option is selected, the line style is defined as a continuous solid line. The other options in the dialog are dimmed and cannot be used.

**Dashed Line**: When this option is selected, the other options in the dialog are used to define the pattern of a dashed line.

**Pattern Length**: The number entered in this field specifies the length of the pattern or module that will be used to define the line style. The value is complemented by the unit selected from the pop up menu next to it.
When this value is changed, a message is posted warning the user that this change will affect the line style. Clicking the **OK** button of the warning dialog will scale the components (dashes and gaps) of the pattern proportionally to the new pattern length. Clicking the **Cancel** button keeps the sizes of the components as they are. If the new pattern length is shorter than the previous, the right end of the pattern is trimmed according to the new length. If the new length is longer, the additional space is filled with a blank (gap).

**Pattern Unit:** The item selected from this pop up menu determines the units of measurement to be used to interpret the value entered in the **Pattern Length** field. Available units are **Point**, **Inch**, **MIL**, **mm**, and **cm**. Each time the unit is changed, the texture length displayed on the pattern ruler (see below) is automatically recalculated, but the layout of the style does not change.

Line style patterns are defined graphically using the **pattern ruler**, or numerically by entering values in the **Start**, **Length**, or **End** fields, located under the ruler, or by a combination of these two methods.

The total length of the pattern ruler is determined by the value entered in the **Pattern Length** field, and the unit selected from the **Unit** menu. The ruler is also subdivided according to the selected unit. A pattern in the ruler is defined by using sliding **dash bars**, which are representations of the physical characteristics of dashes.

A dash bar consists of the following parts:

- The **dash marker** is a horizontal line corresponding to the size and position of a dash. Clicking on it selects it and also selects the **Length** field. Selected entities are shown in the highlight color.
- The **start marker** is a small vertical line with a diamond at its top. Clicking on it selects it and also selects the **Start** field.
- The **end marker** is as the start marker except it is longer. Clicking on it selects it and also selects the **End** field.

All the dash bars and their parts can be moved to the right or left, except for the first (left most) dash, only the end marker of which can be moved. Move the complete bar by clicking the mouse on the dash marker and dragging it. The operation is completed when the mouse is released.

The start and end markers are moved by clicking inside their diamonds and dragging them. Pressing the **shift** key while moving the markers moves the entire dash bar. When you click on a part of a dash bar to move it, the mouse cursor changes to a single (←, →) or a double (↔) arrow pointing in the one or two directions towards which the move can occur. The arrows usually point left and right. It points in one direction when a dash bar touches another bar or the right end of the ruler, which constrain its movement.
Each time you select a dash bar or one of its parts, the number in the **Start**, **Length**, and **End** fields are updated to reflect the values corresponding to the selected dash bar. Any of these values can be changed, resulting in a move which is executed through numeric input. The graphics in the pattern ruler are updated as soon as you click in another text field.

When you edit a previously defined line style, the pattern of that style is displayed in the pattern ruler when the **Line Style Editor** dialog is invoked. When the dialog is invoked from the **New...** button in the **Line Styles** dialog, a **Solid** line is selected and displayed by default. To define a dash line, first select the **Dash Line** option. The pattern ruler continues to display a solid line. Move its end marker to transform it into a dash. Additional dashes can be added by picking and dragging the half diamonds that appear at the right end of the pattern ruler. The new dash can be positioned and sized as discussed above.

When you move dash bars or their parts graphically, you are constrained by the neighboring dashes and cannot make a dash overlap another. When you move them through numeric input, it is possible to cause a dash to overlap another, which is not allowed. When this happens, the system posts a warning message. While they cannot overlap, dashes can touch each other. However, when they do, they will be automatically connected by the system into a single dash when the pattern of the line style is stored.

**Snap To Ruler:** When this check box is selected (default), and dash bars or their parts are moved graphically in the pattern ruler, they will snap to the ruler subdivisions. This option has no effect on the numeric input entered in the **Start**, **Length**, and **End** text fields.

The value in the **Pattern Length** parameter and the selection from the **Pattern Unit** menu affect how line styles are printed or plotted on plotters directly. These measurements will be used when generating dashed line patterns. To display them on a computer screen, some additional measurement in pixels needs to be superimposed, which is controlled by the next option.

**Screen Display:** The value entered in this field determines the number of pixels that will be used to construct and display on the screen the pattern defined in the pattern ruler. That is, if pattern length is 1 inch, and 100 pixels are entered in this field, the line style will be displayed using 100 pixels per module. If 200 is entered then each module will be 200 pixels, resulting in a larger dash line pattern. That is, the second time the dash pattern will appear scaled by 200%. Note again that this option affects only the screen display and has absolutely no effect on how line styles are printed.
5.15.2 Setting and assigning line weights

![Line Weight](image1)

This tool changes the line weight of the picked element to the active line weight. A line weight is an element level attribute. It complements the element’s line style, and follows the same rules. This operation has no effect on text or symbol elements.

The line weights are displayed in the Line Weights palette shown in Figure 5.15.2.1. When the default preferences are used, this palette appears on the screen when launching the program and a drafting window is opened. If it does not appear, it can be opened by selecting the Line Weights item in the Palettes menu.

Clicking in the header bar of the Line Weights palette, or clicking in the Line Weights column while pressing the option key (Macintosh) or ctrl+shift (Windows), invokes the Line Weights dialog, which can also be invoked from the Options menu. The Line Weights dialog (Figure 5.15.2.2) works as the Line Styles dialog.

![Line Weights](image2)

**Figure 5.15.2.1:** Line Weights palette with weights displayed (a) graphically, and (b) by name.

![Line Weights](image3)

**Figure 5.15.2.2:** The Line Weights dialog.
The **Line Weight Editor** dialog (Figure 5.15.2.3) is invoked from the Line Weights palette by double clicking on a line weight, or by selecting the **New...** or **Edit...** button in the **Line Weights** dialog.

![Figure 5.15.2.3: The Line Weight Editor dialog.](image)

**Name**: This field displays the name of the line weight that is being edited. For new line weights, a name is initially defaulted by the system and can be changed by the user.

**Print Width**: The value entered in this field, interpreted according to the selection from the Unit menu next to it, determines the width of a line when it is printed or plotted on paper. This popup parameter has no effect on how lines are displayed on the screen.

**Unit**: The item selected from this menu is used to interpret the value entered in the **Print Width** field. The available units of measurement are as in the **Line Style Editor** dialog.

**Screen Width**: The value entered in this field is in pixels and determines the width of a line when displayed on the screen. It has no effect on how lines are printed or plotted.

**Color Code**: Line weights have a color associated with them. This color is displayed in this box, and can be changed by clicking on it to invoke the *form-Z* Color palette. The color associated with a line weight is used to display a line when the **Color Of Line Weight** option is selected in the **Drafting Display Options** dialog.
5.15.3 Setting the color and all attributes

Color

This tool changes the color of the picked element to the current color, which is selected from the Colors palette. This operation is identical to that of the modeling environment, and is discussed in sections 2.6 and 4.25.

Set All Attributes

This tool simultaneously changes the line style, the line weight, the color, and the layer of the picked element to the current settings of those attributes. That is, the Set All Attributes tool is equivalent to the execution of four distinct attribute setting tools. It is restricted by the same conditions that apply to the individual operators. It skips the line style and line weight for symbol elements and the line style for text elements.
5.15.4 Ghosting/unghosting and setting layers

![Image of Ghost and Unghost buttons]

These tools ghost and unghost elements, respectively. They can only be applied at the Element level.

The “ghost” attribute causes an element to become inactive. A ghosted element retains its existence, but it cannot be picked for operations while it remains ghosted. Ghosted elements are plotted in the graphics window in a light gray color. Gray is the default color and you can change it in the Project Colors dialog (see subsection 2.6.3). The only operator that can select ghosted elements individually is the Unghost operator. All the ghosted elements can be selected by executing the Select All Ghosted command, under the Edit menu (see subsection 3.2.4).

These tools work with both the prepick and postpick selection methods. When the postpick method is used, the elements are selected and ghosted or unghosted one at a time, after the operator has been selected. When the prepick method is used, the elements are selected first, then the desired operator is activated and a single click in the graphics window initiates the execution of the ghosting/unghosting operation. Prepicking of the ghosted elements is only possible using the Select All Ghosted command.

![Image of Set Layer button]

Set Layer

This tool moves the selected elements to the active layer or a layer specified in a pop up menu, depending on th selection option in the Set Layer Options dialog. Layers can be created, labeled and selected through the Layers palette and the Layers dialog, as in modeling (see section 2.7).

![Set Layer Options dialog]

Figure 5.15.4.1: The Set Layer Options dialog.
5.16 Hatching

Hatching or patterning is used to texture or fill an area of a drawing that is delineated by one or more shapes. *form-Z* provides two types of hatching: *vector* and *bit-mapped* hatching. The bit-mapped patterns are pre-stored and are based on the standard conventions. The vector line patterns are generated by the system each time they are displayed.

The vector hatchings are sensitive to scale, while the bit-mapped hatchings are not. That is, whenever the scale of a drawing changes, the density of the vector pattern also changes. The bit-mapped hatching is always displayed in the same scale, regardless of the scale of the drawing. This is illustrated in Figure 5.16.0.1. In addition, the bit-mapped hatches are opaque, while the vector hatches are transparent. If a bit-mapped hatch is placed on top of another hatch, it covers it. If a vector hatch is placed on top of another hatch both are visible.

*Figure 5.16.0.1:* (a) Vector hatching and (b) bit-mapped hatching at (1) large and (2) small scales.
5.16.1 Selecting the current hatching

Whether a vector or a bit-mapped hatch pattern is applied depends on the **active hatch pattern**, which may be either one of the two types. The active hatch pattern is selected from the Hatch Patterns palette, that is invoked from the Palettes menu. This palette is shown in Figure 5.16.1.1.

![Figure 5.16.1.1: The Hatch Patterns palette: (a) the active hatch pattern, (b) the vector, and (c) the bit-mapped patterns.]

The Hatch Patterns palette is divided into three areas. The large square with the dark border (a) shows an image of the **active hatch pattern**. The two boxes to its left (b) represent the two vector patterns available: the **parallel line** pattern (top) and the **stipple** pattern (bottom). The 38 boxes to the right of the active hatch pattern box (c) are images of the bit-mapped patterns available. Either type of pattern is selected by clicking on a pattern box. The selected pattern is displayed in the active hatch pattern box and becomes the active hatch pattern.

The bit-mapped patterns are laid out as shown in the Hatch Patterns palette. The density and orientation of the vector patterns can be changed through parameters available in the **Hatching Options** dialog (Figure 5.16.2.1) that is invoked directly from the Hatch tool. Note that changing these parameters does not affect the display in the active hatch pattern box, but the current parameters are used when a vector hatch pattern is placed on a drawing.
5.16.2 Changing the parameters of the vector patterns

The vector textures are controlled by the parameters in the **Hatching Options** dialog, shown in Figure 5.16.2.1. The **Hatch Scale** controls the distance between the lines in the parallel line pattern, and the density of the stipple pattern. The default is 12" for English and 25 cm for metric. The **Hatch Angle** parameter controls the slope of the lines in the parallel line pattern. The default is 45°, where the horizontal line is at 0° and the positive angles are counterclockwise. This parameter has no effect on the stipple pattern.

The vector patterns are also affected by the current line style and weight, which are selected from the Line Styles and Line Weights palettes. The parameters in the **Hatching Options** dialog, especially when they are combined with different line styles and weights, lend themselves to the generation of a large variety of patterns. Some examples are shown in Figure 5.16.2.2. The stipple patterns in 5.16.2.2(a) were generated with the **Hatch Scale** ranging from 3" to 1'-6", in increments of 3". Figure 5.16.2.2(b) illustrates vector patterns at angles of 45°, 135°, 45°+135°, 0°, 90°, and 0°+90°, in the default scale. Figure 5.16.2.2(c) illustrates patterns generated by a variety of dashed lines and weights. The four patterns on the right are composite patterns.

---

**Figure 5.16.2.1:** The **Hatching Options** dialog.

**Figure 5.16.2.2:** Samples of vector hatching patterns.
5.16.3 Applying hatching

This tool fills the selected shape with the current hatch pattern, which may be a vector or a bit-mapped pattern. Hatching can be applied to all element types, except text, dimensions, and symbols. By definition, hatch patterns require bounded areas and are normally applied to closed shapes. However, the system also allows patterns to be applied to open shapes. Such shapes are first closed, by connecting their open ends, and then the hatch pattern is applied. Note, these shapes are only closed for the purpose of generating the hatch pattern and are not closed permanently. Examples of hatch patterns applied to both closed and open shapes are shown in Figure 5.16.3.1.

Hatch patterns can be applied using either the postpick or the prepick method. When the postpick method is used, the Hatch tool is selected first, then the mouse is clicked on the element to which the pattern is to be attached. The hatch pattern is displayed immediately. Hatching requires the entities to be picked at the Element level.

The prepick method can be used to apply the same hatch pattern to any number of shapes. With this method, the elements are selected first, then the Hatch tool is activated, followed by a single click on the graphics window, which causes the hatch patterns to be displayed. The prepicked shapes may be disjoint or they may intersect. The system depicts the closed areas and applies the hatch pattern to every other closed area, starting with the outer most areas (Figure 5.16.3.2). Preselecting shapes also provides the ability to treat shapes inside other shapes as holes.

Hatching is associative. That is, when the shape to which it has been attached changes, the hatch pattern is regenerated to fill the new shape. When the element to which hatching has been attached is moved or otherwise geometrically transformed, the hatching follows the shape. If the element is deleted, so is the hatching. The hatching can only be deleted from a shape by clicking the mouse inside the hatched area, after activating the Delete tool.
5.17 Delete

The delete tool works in conjunction with the Topological Level modifiers. Deletions can occur at the **Point, Segment, Element, Compound,** or **Area** topological level.

Unlike the modeling environment, the drafting environment has a single Delete tool. However, depending on which topological level is active, it may execute a topological editing operation, or it may simply be used to erase an element from memory.

![Delete](image)

This tool deletes the selected entity or entities. When the topological level is at Point or Segment, it deletes a point or a segment, respectively, which are topological editing operations. When the topological level is at Element or Compound, it deletes the selected element or compound completely. When at Area, it deletes the elements that are selected by the area. Complete elements or clipped elements may be deleted when using the area.

The Delete tool works with both the prepick and postpick methods. With the postpick method, the Delete tool is selected first, then the entities are selected and deleted one at a time. When the prepick method is used, any number of entities are selected first, then the Delete tool is activated, followed by a single click of the mouse anywhere in the graphics window.

A prepick method for deleting multiple points and/or segments of the same element is not available.
5.17.1 Deleting points

A point is deleted by connecting the point before to the point after. If the point is the first or the last point of an open element, that point is simply removed and the corresponding segment of the open curve disappears. If the element contains only one segment then the deletion of a point results in a degenerate element, which causes the complete element to be deleted. Note that the effects of the deletions of points in the drafting environment are identical to the point deletions of the modeling environment when applied to surface objects.

Points can be deleted from polylines (excluding the rectangles and point elements), leader lines, and the boundary shape of an area when the area has been generated from a polyline (see Figure 5.17.1.1). To delete points from other types of drafting elements, they must first be exploded.

Figure 5.17.1.1: Deleting points from (a) a closed polyline, (b) an open polyline, (c) first point of an open polyline, (d) a leader line, and (e) the boundary of an area.
5.17.2 Deleting segments

A segment is deleted by removing the connection between its two end points. Note that, this is similar to the topological deletion in modeling. The connections at the end points of a segment are not rerouted, but they are simply broken. An implication of this is that, while the deletion of a point affects the shape of the polyline from which it is deleted, the deletion of a segment does not. However, the deletion of a segment breaks a polyline to two portions whenever that segment is not the first or the last segment of an open polyline element.

The deletion of the first (or last) point and the deletion of the first (or last) segment of an open polyline produce identical results. A segment deletion similar to the geometric deletion of the modeling environment (see subsection 4.26.3) can be simulated by deleting the endpoints of a segment, one at a time. If an element contains only one segment, then the entire element is deleted when that segment is deleted.

Segments can be deleted only from polylines, excluding the rectangles, and leader lines (see Figure 5.17.2.1). To delete points from other types of drafting elements, they must first be exploded.

Figure 5.17.2.1: Deleting segments from drafting elements: (a) a closed hexagonal polyline and (b) an open polyline. (c) Deleting the first segment of an open polyline. Deleting segments from (d) a leader line and (e) the boundary shape of an area.
5.17.3 Deleting elements

All types of drafting elements can be deleted when executing the Delete tool with the topological level set to Element. Deleting an element eliminates it from the system’s memory and erases it from the screen. The Delete tool at the Element Level can also be used to delete the hatch pattern of a shape.

Recall that polylines, arcs, dimensions, leader lines, text, and the boundary shape of an area are picked by clicking the mouse on one of their segments. Image elements, and symbols are picked by clicking within their bounding boxes. A hatch pattern is picked by clicking within the element to which it is attached. A point element is picked by clicking on its screen point.

The deletion of elements with which hatch patterns are associated also causes the deletion of these hatch patterns. The deletion of an associative element does not affect the element that it references. Deleting an element which is referenced by a dimension does not delete the dimension as well, but it converts the status of the dimension to non-associated. The deletion of the boundary shape of an area element (executed with topological level at Element), closes that area, without affecting the elements that may be selected by the area.

The Delete tool is complemented by two menu commands, found under the Edit menu: Clear and Clear All Ghosted. The first deletes all unghosted and the latter all ghosted elements. This is the only direct method for eliminating ghosted elements from the system’s memory. The Delete tool has no effect on ghosted elements. To delete ghosted elements using the Delete tool, they must first be unghosted.
5.17.4 Deleting with the area

The Delete tool can also be executed by picking an area element, while the topological level is at Area. It deletes the elements that are selected by the area element. Which elements are selected depends on the options selected in the **Area Options** dialog (see subsection 5.3.3). Recall that the following groups of elements may be selected: those (a) completely inside, (b) inside together with those crossing the area boundary, (c) inside and the inside clipped portions of those crossing the area boundary, (d) completely outside, (e) outside together with those crossing the area boundary, and (f) outside and the outside clipped portions of those crossing the area boundary. The latter two cases result when the **Clip To Area** option is selected. These cases are illustrated in Figure 5.17.4.1.

![Figure 5.17.4.1](image)

*Figure 5.17.4.1:* Deletions using the area element: (a) all inside elements, (b) all inside and those crossing or touching the area boundary, (c) all inside parts of clipped elements, (d) all outside, (e) all outside and crossing the area boundary, and (f) all outside parts of clipped elements.

Note that deleting the boundary shape of an area and deleting the elements that are selected by the area are two distinct operations. The first is executed with the topological level at **Element**, the area is picked by clicking on its boundary shape, and the operation causes the area itself to be deleted. The second is executed with topological level at **Area**, the area is picked by clicking inside its bounding shape, and the operation causes the elements selected by the area (clipped or unclipped) to be deleted. The area element itself is not deleted.
5.18 Importing image elements and modeling views

Drafting elements can also be created by importing images from the modeling environment. There are two distinct types of elements that can be imported: image elements and polyline versions of modeling views.

The image elements can be imported into drafting after they have been captured in the operating system’s Clipboard.

The form•Z Clipboard may be used to import modeling images as polylines. These images may be axonometric or perspective views of a modeling configuration, or the orthographic projection of a model, or simply two dimensional displays of surface objects.

Recall that, even in the 3D world of modeling, in order to be able to display the image of a 3D scene on the inherently 2D screen, a viewing transformation is applied, which reduces the three dimensions to two. These two dimensions are then used to display the view on the screen. That view may be a perspective, an axonometric, or simply an orthographic projection to a plane. When an image is transported from the modeling to the drafting environment, the same transformation is applied in order to reduce the three dimensions of modeling to the two which the drafting environment can handle. What is actually transported to the drafting environment is the view that is displayed on the screen.

All modeling views are brought into the drafting environment as polylines which can be freely edited. When the corresponding source shape of the modeling environment is a connected sequence of segments, it can be brought into drafting in the same fashion, which is as a polyline consisting of a number of connected segments, or it may be brought in as a number of single segment polylines, one for each segment in the original shape. This is controlled by the option selected in the Paste From Modeling dialog (see subsection 3.2.2).
5.18.1 Importing image elements

The process of importing an image from the modeling environment into the drafting environment involves two steps (see subsection 3.2.2). First, the image is captured using the operating system’s Clipboard. This places it in the system Clipboard from which it can be brought into the drafting window by executing the Paste Image command. Images can also be placed in the Clipboard by an application other than the operating system. Once in the Clipboard, they are brought into the drafting window as an image element. The placement of an image, captured from modeling and placed into a drafting window, is illustrated in Figure 5.18.1.1.

Figure 5.18.1.1: An image which was captured in modeling placed in the operating system’s Clipboard and placed on a drafting window.

Once in the drafting window, image elements can be moved or scaled. They can not be rotated or reflected. The Move and the Scale tools can be executed at any topological level (except Area) and with any of the Self/Copy modifiers. The image element always has a rectangular shape, and behaves as a rectangle when it is geometrically transformed.
5.18.2 Importing modeling views

The process of importing an image from the modeling to the drafting environment involves two steps. First, the modeling object, whose image is to be transported, is placed in the form-Z Clipboard. Then it is brought into the drafting window from the Clipboard. As the latter step is executed, the modeling representation of an object is transformed to a drafting representation, which changes the view of the object to one or more polylines.

To place one or more objects in the form-Z Clipboard, they are first picked and highlighted in the modeling window. The objects in the modeling window may be displayed in wire frame or hidden line, which will affect the image that will be transported. After objects have been selected either the Copy or the Cut command, from the Edit menu, is selected. The operation is executed immediately. If the Cut command is used, the picked objects are also deleted from the modeling window. If the Copy command is used, then a copy is made and is placed in the form-Z Clipboard. The original objects remain in the project window.

Once an object has been placed in the Clipboard, the drafting window, where it will be transported, is activated. That drafting window may be from the same project or a different project. Once a drafting window is active, the Paste From Modeling* command, from the Edit menu, is selected. The images of the objects that were placed in the Clipboard appear in the drafting window, immediately. They are highlighted, and can be moved if appropriate.

The drafting elements are created in exactly the same position where the images of the modeling objects were at the time they were placed in the Clipboard. That position is always interpreted relative to the position of the origins of the modeling and the drafting windows. If both origins are in their default positions, which are at the centers of the windows, the transported image appears in exactly the same position. This is illustrated in Figure 5.18.2.1. If one or both of the origins have been moved, then the position of the transported image reflects these moves.

![Figure 5.18.2.1: Transporting (a) the view of a model into (b) a drafting window.](image)
The import process also preserves the scale of the image. If the scale of the drafting window where the element is imported is the same as the scale of the modeling window from which it was captured, then the two images appear identical in size. If they are at different scales, then the transported image appears smaller or larger.

The drafting elements that are generated from modeling images are always polylines. Whenever the source shapes are connected sequences of segments, such as the faces of solids, they may be transported as polylines that consist of exactly the same sequence of connected segments or as collections of single segment polylines. This option is controlled by the Import 3D Faces As option in the Paste From Modeling dialog that is invoked from the Paste From Modeling* item (Edit menu). Default is Each Face/Outline As A Polyline, which produces connected polylines. When Each 3D Segment As A Single Line is selected, one polyline is created for each pair of the reversely coincident segments of the 3D modeling representation. The two ways in which drafting elements can be produced from the modeling images are illustrated in Figure 5.18.2.2.

*Figure 5.18.2.2: A 3D view transported into drafting as (a) connected polylines and (b) single segment polylines.
5.18.3 Using modeling operations for drafting purposes

There are certain operations that are available in the modeling but not in the drafting environment. Yet, they represent capabilities which can facilitate a variety of drafting tasks. It is possible to execute such drafting tasks in the modeling environment, and then transport the resulting drawing into the drafting environment for further refinement or inclusion in another drawing. Such operations may be executed in exactly the same way as the importation of modeling views. However, their functionality is different, which makes it appropriate to discuss and illustrate them separately.

The most notable modeling operations that can be used advantageously for drafting purposes are the attachments, and the macro transformations. In the modeling environment they can all be applied to both 3D solids and 2D surface objects. Given the requirement for transporting the created shapes into the drafting environment, the discussion here focuses exclusively on their application to surface objects.

The remainder of this section presents two examples where some work is first done in the modeling environment, and then the resulting objects are transported as polylines into the drafting environment and used for the production of a drawing.

The first example, illustrated in Figure 5.18.3.1, uses the Attachment tool in the modeling environment to construct a “tile” pattern using surface objects. That pattern is then transported into the drafting environment where it is transformed into a symbol and used to fill a surface.

A surface octagon (8-sided polygon) is drawn and then point-to-point attachments are applied to derive a 4x4 arrangement of octagons. The scale at which the pattern is initially created does not matter. However, after it has been created it should be positioned and scaled so that its boundary edges lie on integer increments of the grid. The resulting pattern is shown in Figure 5.18.3.1(a). Its left and bottom edges have been positioned on the Y and X axes, respectively, and it has been scaled to a dimension of 4’x4’. The drawing is in 1/4" scale.

The pattern is transported into a drafting window in the usual manner. When it appears in the drafting window (Figure 5.18.3.1(b)), it is in a picked state, which allows us to proceed immediately and transform it to a symbol. The Create Symbol tool is selected and the mouse is clicked on the lower left corner of the pattern. We can now repeat that pattern as many times as we desire. In the example, we drew an irregular shape and then filled it with the pattern (Figure 5.18.3.1(c)). Next we generated an area element with the same shape, selected the Clip To Area and the Outside options in the Area Options dialog, picked the area (used the Pick tool with topological level at Area), and executed a Delete operation. The result is shown in Figure 5.18.3.1(d).
The second example, which is illustrated in Figure 5.18.3.2, again uses a modeling feature that is unavailable in drafting, namely the Macro Transformations, to create a composite shape that is then transported into the drafting environment.

Using the Scale and the Rotation tools, a macro transformation is defined and is next applied to a rectangle to create the composite shape shown in Figure 5.18.3.2(a). That shape is transported into the drafting environment in the usual way, and is transformed into a symbol. Then it can be repeated as many times as appropriate. In the example shown in Figure 5.18.3.2(b), the symbol is placed four times, using different values for the parameters in the Symbols Place Options dialog.

Figure 5.18.3.2: A composite shape (a) created in modeling using a Macro Transformation (b) is transported and displayed in the drafting environment.
6.0 Introduction

form•Z RenderZone Plus is the version of form•Z which includes photorealistic rendering provided by LightWorks®, a rendering engine developed by LightWork Design Limited. It offers three levels of rendering: Simple, z-buffer, and raytrace. A user can start developing the image of a 3D model at the Simple level, and gradually turn on features and develop it into an image that is rendered at the most photorealistic levels.

Included in form•Z RenderZone Plus is the ability to produce images based on global illumination. This technique creates renderings that have the highest level of realism, as the illumination of a scene takes into account the accurate distribution of light in an environment. Global illumination is a term, that summarizes a number of individual techniques, including final gather, ambient occlusion, and radiosity.

While color is the essence of rendering, the images included in this volume are in black and white. Images in full color will be found in the electronic version of this manual available on the documentation CD that comes with your package. It is strongly recommended that you inspect the color images as you read this manual.
6.0.1 The rendering levels and their rendering effects

The Simple rendering level results in the fastest renderings, with limited quality and rendering effects, while Raytrace gives the best image quality, but takes the most time to execute. The simple and z-buffer rendering modes use the z-buffer algorithm to determine the visibility of a surface at a given pixel. That is, each face to be rendered is scanned pixel by pixel, and the distance of each pixel from the view point is recorded in the z-buffer. The pixel which is nearest the viewpoint is used to determine the color at that location in the buffer.

Raytrace determines the closest pixel by casting a ray from the view point through each pixel and intersecting it with each face. The intersection point which is closest to the viewer is used to calculate the pixel color. Ray tracing can easily generate accurate reflections and transparencies on a surface, since rays can bounce off reflective surfaces and pass through transparent surfaces. However, ray tracing is a costly algorithm, and rendering times may increase significantly, especially if a scene contains a large number of reflective or transparent surfaces. On the other hand, raytracing can be accelerated when multiple processors are available, which will offset the slower algorithm. When rendered on a machine with 2 or more processors Raytrace renderings will produce the best results at a speed that is even faster than z-buffer.

To generate a quickly rendered image, the z-buffer modes are usually more efficient. However, due to the nature of the algorithm, accurate transparencies and reflections cannot be generated directly. To address this issue, form•Z RenderZone Plus applies a mixed method where the z-buffer algorithm is combined with ray tracing. While the efficiency of the z-buffer is used to determine the closest pixel, reflections and transparencies are generated by casting rays from pixels on reflective and transparent surfaces. Since the majority of the pixels in an image are usually not located on such surfaces, only a limited amount of time is spent in ray tracing.
6.0.2 How the rendering effects are created

Rendering with form•Z RenderZone Plus involves the following steps:

- A surface style is created or selected from a set of predefined surface styles. The surface styles are displayed in the Surface Styles palette, which contains style definitions with an extended set of parameters, compared to their counterparts in the regular version of form•Z. Surface styles can be edited and changed by double clicking on them in the Surface Styles palette.

- The active surface style is assigned to an object at the time it is created. A surface style can also be assigned to a complete object or to individual faces of an object using the Color tool, as in the regular version. A surface style can be thought of as a color which is assigned to the surfaces of an object; flat colors are actually the simplest forms of surface styles.

- The textures that may be contained in a surface style are mapped onto objects by default when objects are created. The Texture Map tool may then be used to specify the exact position of a texture, the type of texture mapping (flat, cubic, cylindrical, spherical, parametric, or UV coordinates), the size of the texture tile and other parameters. This is done through a preview dialog which displays the effects of the parameters at different levels of detail.

- Surface styles can also be attached to surfaces of objects as decals on top of and in addition to previously assigned surface styles. This is done using the Decal tool, which works in a manner similar to the Texture Map tool.

- One or more lights are generated and positioned.

- Once the surface styles are defined and assigned to the objects and/or their surfaces, and the lights are positioned, a rendered image is created by selecting the RenderZone* item from the Display menu. The RenderZone Options dialog can also be invoked directly from this item in order to select the desired type of rendering and to turn rendering effects on or off (see next section). The rendered image is then generated in the active window where the 3D model is currently displayed.

Note that a surface style, once defined and displayed in the Surface Styles palette, is of global character. That is, the same surface style can be assigned to any number of objects. A surface style is individualized when it is mapped to objects either as a basic surface style or as a decal. The same surface style may appear differently on different objects, depending on how it is mapped and what scale is used.

Following are a few specific examples previewing some of the steps involved in generating photorealistic renderings. Note however that, in this section, no attempt is made to fully discuss the dialogs and tools used in the examples. These are discussed in more detail later in this manual. The examples are complemented by a few files that are supplied with form•Z. You can import them and render the objects they contain with preset rendering parameters.
Refining the surface styles

By default the surface styles generate a matte shaded surface with a uniform color. The image in Figure 6.0.2.1 shows a scene in which all the objects are rendered with default surface styles. The realism of the surfaces can be improved by choosing surface style parameters that simulate specific materials. In the rendering displayed in Figure 6.0.2.2, the surface style for the car body was given a reflective property, which causes the objects next to the car’s body to show up as reflections. Likewise, the surface style for the tires has been given a plastic like reflective property, and the surface style for the windshield was assigned glass like reflection parameters. The Surface Style Parameters dialog, which is discussed in further detail in section 6.2.2, is shown in Figure 6.0.2.3 with the parameters used for the reflective car body. You may recreate the renderings shown in Figures 6.0.2.1 and 6.0.2.2 by loading the sample files labeled “car_1.fmz” and “car_2.fmz”, respectively. The former contains the simple rendering and the latter has the modified surface styles. You may want to double click on the surface styles to observe the different parameters. Choose RenderZone* from the Display menu to create the images.
Mapping textures

Textures are patterns derived from image files or procedural algorithms. They can be used in a surface style, for example, to turn the default uniform color into a color pattern. Textures can change the color, transparency, and bumpiness of a surface style. These textures are mapped onto an object with the Texture Map and Decal tools, with which the exact size and orientation of the texture is determined. In the rendering of Figure 6.0.2.4 additional textures were mapped on the car. For example, a bump texture was mapped onto each tire. The texture is mapped in a cylindrical fashion around the object, which can be previewed in the Texture Map Controls dialog shown in Figure 6.0.2.5. The Texture Map and Decal tools are discussed in further detail in sections 6.3 and 6.4. The sample file labeled “car_3.fmz” contains the above scene with additional textures defined and mapped onto various objects. You can choose the Texture Map tool, located in the Attributes tool palette and click on one of the tires. This will open the Texture Map Controls dialog for this object and will show you how the bump texture is wrapped around the tire.
Defining light sources

By default a form•Z project contains a single light source, which represents the sun. Additional light sources can be defined, as discussed in further detail in section 6.5. In the image shown in Figure 6.0.2.6, three additional light sources were added, which simulate spot lights. Their shadow casting attribute was turned on. This scene is located in sample file “car_4.fmz.” After opening the file, you can see the additional light sources in the Lights palette. Double clicking on any of the names in the palette opens the Light Parameters dialog, as shown in Figure 6.0.2.7, where the parameters of the light are displayed.

Figure 6.0.2.6: Three lights with shadow casting attributes were added to the rendering.

Figure 6.0.2.7: The Light Parameters dialog.
Additional rendering effects

Additional rendering effects can be added to a scene by turning on the respective options in the RenderZone Options dialog, shown in Figure 6.1.0.1, in the next section, where its options are also discussed in detail. In the rendering of Figure 6.0.2.8, a different background was selected and a depth effect was added, which blends the ground plane and the background in the far distance. The sample file “car_5.fmz” can be used to create this rendering. While selecting RenderZone* from the Display menu, hold down the option key (Macintosh) or cntrl/shift (Windows). This opens the RenderZone Options dialog. The background and depth effect options used for the rendering are located in their respective categories in the dialog.

*Figure 6.0.2.8: A new background with a depth effect is added.*
6.0.3 Multiprocessor support

form•Z RenderZone Plus supports multiprocessor machines running the Windows and MacOS X operating systems.

Multiprocessor support only affects RenderZone renderings when using the Raytrace rendering mode. For most scenes, a rendering using two processors should increase the rendering performance by a factor of 1.6 to 1.8 when compared to a single processor machine. form•Z will take advantage of as many processors as are installed, so if you have a quad processor machine, the increase will be even more dramatic.

During a multiprocessor rendering you may notice that the rhythm of the bands drawn on the screen during the rendering is less regular. The rendering is split up into bands which are each handled by one of the available processors and cached until drawn on the screen.
6.1 Producing a rendered image

The modeling scene displayed in the active window of form•Z can be rendered by clicking on the RenderZone* item in the Display menu. The rendering parameters are set in the RenderZone Options dialog (Figure 6.1.0.1). It is invoked by selecting the RenderZone Options... button in the Display Options dialog, or by clicking on the RenderZone* item (Display menu) while pressing the option key on the Macintosh or ctrl+shift on Windows.

The RenderZone Options dialog contains five tabs and a few options outside the tabs. The tabs are labeled Shading, Global Illumination, Geometry, Scene, and Postprocess. The options in these tabs are discussed in this section.

![Figure 6.1.0.1: The RenderZone Options dialog with the Shading tab open.](image)
6.1.1 Selecting rendering type and parameters

One of three available rendering types is selected from the Rendering Type pop up menu available at the top of the RenderZone Options dialog. When the Z-Buffer or Raytrace rendering method is selected, the Options... button next to it is active. It is not active for the Simple rendering method. This button invokes a dialog in which raytrace parameters may be set. The RenderZone Options dialog also contains a variety of other options for setting the image size and other rendering parameters. An overview of all these parameters is offered in this subsection and are discussed in more detail in the next three sections.

Rendering type

The following three rendering types can be selected from the Rendering Type pop up menu:

**Simple**: This is the fastest rendering method. It produces images very quickly at the cost of relatively low image quality. It is meant to preview the early stages of a design, when it is more important to achieve a feeling for the three dimensional quality of a design, rather than realistic images. Each face is rendered as a single, opaque surface in a uniform color.

**Z-Buffer**: This mode is capable of generating high quality images and is intended for final output of rendered images. It has the ability to show shadows, textures, reflections, bumps, backgrounds, foregrounds, depth effects, transparencies, and antialiasing.

**Raytrace**: This mode determines the visibility of a surface by raytracing and is capable of generating high quality photorealistic images that include all the rendering features. Pixels throughout the image are sampled at a rate sufficient to suppress the appearance of aliased artifacts. This rendering mode will use multiple processors if they are available. The Preferences dialog, which is invoked from the Edit menu shows in the System: General section how many processors are allocated for multiprocessing during raytrace rendering.

Setting the raytrace parameters

**Options...**: Clicking on this button when the rendering mode is set to Z-Buffer or Raytrace invokes the Raytrace Options dialog (Figure 6.1.1.1). The parameters in this dialog control the performance of raytraced images. By default, they are set to their optimal values for an average scene and it is generally not necessary to change them. However, increased performance or quality can be achieved by carefully adjusting these parameters.

![Raytrace Options dialog](image)

**Figure 6.1.1.1**: The Raytrace Options dialog.
**Maximum Number Of Recursive Rays**: The number in this field determines how often a ray can be reflected from a surface, or pass through a transparent surface. Since reflected and transmitted rays contribute to the final color of a pixel, increasing this value will result in a more accurate rendering. However, a large number of recursive rays increases the rendering times. In general, values above 8 will not produce a significant improvement in image quality. As a matter of fact, a value of 4 produces acceptable results. Only in a scene with a large number of reflecting surfaces would a higher value be needed for photorealistic results.

**Maximum Number Of Polygons Per Bounding Volume**: In order to optimize the intersection calculation of rays with all faces, bounding volumes are formed around clusters of faces. If a ray does not pass through a bounding volume, no intersection checks are performed with the faces inside this volume. Bounding volumes are constructed so that they contain the maximum number of faces, in the smallest possible volume. A bounding volume is further subdivided if more faces than indicated in this field are enclosed. Increasing this number will cause bounding volumes to become larger, while smaller numbers will make bounding volumes smaller. If a scene contains objects which are spatially separated, a larger number may provide some speed improvements.

**Maximum Depth Of Bounding Volume Subdivision**: This value indicates how often a bounding volume can be subdivided. Each subdivision yields eight new bounding volumes. That is, with each level of subdivision the number of bounding volumes can grow by its square. While large numbers of bounding volumes increases the efficiency of the raytrace intersection calculation, it also consumes more memory.

**Minimum Ray Contribution**: If a reflected or transmitted ray contributes less light than the amount indicated in this field, no further rays are cast. For example, consider two opposing mirrors with a reflectivity of 60% and a minimum ray contribution of 10%. A ray which bounces between those mirrors contributes only 5% to the initial pixel after five reflections. No further rays would be cast, since the contribution falls below the 10% minimum.

**Antialias Sampling Threshold**: When rendering in Raytrace, pixels are super sampled to eliminate aliasing artifacts. Initially, each corner of a pixel is assigned a color by casting a ray from the view point through the corresponding pixel’s corner point. If these colors differ by at least the amount indicated by this field (showing that the pixel covers a pattern in the image), the pixel is subdivided and sampled again. This process is repeated until no adjacent pairs of color values differ above the indicated amount. The final pixel color is a weighted average of all the calculated colors.
Setting the image size

Normally, the complete modeling scene displayed in a window will be rendered. The option to render only a portion of the scene is also available.

**Set Image Size:** When this option is selected, before starting the execution of the rendering, *form•Z* allows you to delineate the area of the model you wish to be rendered. This feature works as with the **Shaded Render** item.

Pre-estimating the memory required to render an image

Normally, the rendering of a modeling scene will proceed and complete, unless the program runs out of memory. The option to request the program to pre-estimate the memory requirements for rendering a scene is also available. Note that these estimates are approximations. The exact required quantity of memory is known only when the rendering is actually executed.

**Estimate Memory Usage:** When this button is pressed, the memory required by textures, environments, and shadows to complete a rendering is pre-estimated and posted through a dialog.
Setting project level rendering options

The following button duplicates a menu item with the same name found in the Display menu.

Project Rendering Options...: Clicking on this button invokes the Project Rendering Options dialog, which contains two tabs: Smooth Shading and Default Texture Map Control. The content of the first tab is as for the Shaded Render* display mode and is discussed in section 3.6.6. The latter tab is discussed here.

Recall that a Project Rendering Options dialog can be invoked from the Options dialogs of all the rendering modes. However, not all of them have two tabs. Each has the tabs that apply to the type of rendering it performs. Thus only the RenderZone Options and the Interactive Shaded Options dialogs include the Default Texture Map Control tab and the latter only when it runs under the RenderZone Plus version. The plain form-Z version does not support texture mapping for the interactive shaded rendering mode:

The Project Rendering Options dialog with the Default Texture Map Control tab open is shown in Figure 6.1.1.2. Most of its options are the same with those found in the Texture Map Options dialog, invoked from the Texture Map tool, and are discussed in detail in section 6.3. Even though the same the two sets of options perform different functions:

• Given that all objects are generated without texture map control (their texture map attribute is off), the options in the Default Texture Map Control tab are used as defaults when rendering objects to which no specific texture mapping has been applied but have been assigned a surface style that contains textures.

• The options in the Texture Map Options dialog are used when applying textures to an object, as discussed in section 6.3.

![Figure 6.1.1.2: The Default Texture Map Control tab of the Project Rendering Options dialog.](image)
Saving a rendered image

It is possible to instruct the program to automatically save images as soon as they are rendered. This is done by a proper selection from the following pop up menu:

**Save Image**: The three items in this pop up menu determine whether or not a rendered image will be saved after it is completed (Figure 6.1.1.3).

**Off**: When this item is selected, which is the default, the rendering is not saved automatically. It can be saved using the **Export Image** item in the **File** menu.

**Manual**: When this item is selected, the file name, format, and location are determined by invoking the standard file save dialog prior to the execution of each rendering.

**Auto**: When this item is on, the file name is generated automatically by combining the project’s name and the name of the current view. The file is then saved at a specified location. The format and location of the file can be selected in the **Image Autosave Options** dialog (Figure 6.1.1.4), invoked by pressing the **Options...** button next to the menu.

**Format**: The file format in which the rendering will be saved is chosen from this menu. **Options...** invokes the dialog of the respective format.

**Destination**: The image file can be saved **With Project** or in a defined location, which is done by clicking on the **Set Destination Folder...** button to invoke the standard folder location dialog.

**Generate Unique Image File Names**: When this option is selected, the file name is constructed so that existing files will not be overwritten by new image files. This is done by appending incremental numbers to the file name.

**File Info**: This section displays the file name and location of the image file to be saved.

Applying the settings to all windows

**All Windows**: When this option (which is off by default) is selected, the current settings in the **RenderZone Options** dialog will be applied to all the windows of a project. This option is as with the dialogs of the other rendering modes.
6.1.2 The Shading tab

The options in the **Shading** tab (Figure 6.1.0.1) of the **RenderZone Options** dialog allow you to turn on and apply a variety of rendering effects for the active window. Note that for these options to produce a result, the respective effects must also be applied to objects in the scene. For example, to produce reflections, reflective shaders must be assigned to some of the surfaces in the scene and the **Reflections** option must be turned on. If this option is off, no reflections will be produced even though reflective shaders have been assigned to some of the surfaces. In a way, many of the options in the **RenderZone Options** dialog represent a quick way for turning on/off rendering effects globally.

**Shadows**: When this option is selected, and the rendering level is set to **Z-Buffer** or **Raytrace**, objects that have been assigned the shadow casting attribute will cast shadows from lights whose shining and shadow attributes are also on. The shadows will be cast only on objects whose shadow receiving attribute is on. If this option is off, it will override the attributes set for the lights, and no shadows will be cast or received in the rendering.

The **Shadows** option in the **RenderZone Options** dialog has a pop up menu that contains three items (Figure 6.1.2.1).

- **All Opaque**: When this item is selected, all the shadows of all the lights are black, regardless of the transparency or transmission parameters of the surface style that is assigned to an object or face.

- **All Transparent**: When this item is selected, transparent or transmissive objects or faces will cast shadows that take on the color of the object through which a light shines. All the lights in the scene will cast transparent shadows. If a light is set to use **Soft (Mapped)** shadows, raytraced shadows will be used instead to enable transparent shadows.

- **Per Light**: When this item is selected, a light casts transparent or opaque shadows depending on an option selected individually for that light in the **Light Parameters** dialog. That is, while the other two options in this pop up menu override the individual light settings and globally cast the same type of shadows for all the lights, this item allows the type of shadow to vary by light.
When transparent shadows are created using either the **All Transparent** or the **Per Light** option, the following results can be expected:

- If an object is assigned a surface style which uses the **Glass, Accurate, Glass, Simple**, or **Generic** reflection shader, the color of the shadow will be equal to the object color, if the **Transmission** parameter of those two shaders is set to 100% and the shadow is cast from a white light on a perfectly white surface. The shadow color will fall off towards black as the **Transmission** parameter approaches 0%.

- If an object is assigned a surface style which uses any of the transparency shaders except **None**, the shadow will be invisible where the object shows 100% transparency, black where the object is 100% opaque, and a **mixture** of black and the object color where the transparency on the object is between 0 and 100%.

![Figure 6.1.2.2: Rendering opaque and transparent shadows.](image)

Note that transparent shadows may increase the rendering time by a noticeable amount. If a scene does not contain any transparent or transmissive objects, the **All Transparent** shadows item should not be selected. An example with three renderings of a bottle is shown in Figure 6.1.2.2. Image (a) shows the bottle rendered with opaque shadows. Image (b) shows the bottle using the **Glass, Accurate** reflection shader with 100% **Transmission**, rendered with transparent shadows. Image (c) shows the bottle using the **Simple** transparency shader with 50% **Transparency**, rendered with transparent shadows.

**Texture Maps:** When this option is on, the rendering is created using image based color, transparency, and bump maps. When this option is off, only an average color is used and no image, transparency, or bump maps will be included in the rendering. Since texture maps may use large amounts of memory, turning this option off will avoid memory consumption.

**Reflections:** When this option is off, all surfaces are rendered without reflections. If this option is selected, objects in a scene can be rendered with reflections which originate from other objects in the scene or from environment maps. Reflection shaders are discussed further in subsection 6.2.4.

**Transparencies:** When this option is selected, all surfaces which have been assigned a surface style that includes transparencies will be rendered transparent. All surfaces will be rendered opaque when this option is off. Transparency shaders are discussed further in subsection 6.2.5.
**Bump Mapping:** When this option is on, all surfaces which have been assigned a surface style that includes bumps will be rendered with bumps. No bumps will be rendered when this option is off. Bump mapping and the bump shaders are discussed further in subsection 6.2.6.

For the **Z-Buffer** and **Raytrace** rendering types, two options are available which remove aliasing artifacts: **Antialiasing** and **Super Sampling**. Both options are mutually exclusive. For the **Simple** rendering type, only the **Super Sampling** option is available.

**Antialiasing:** When this option is on, antialiasing is performed along edges of faces and inside a face along areas of sharp contrast. Antialiasing removes the "stair stepping" effect common on a pixel based display. The following antialiasing effects are performed:

- Removal of stair stepping artifacts on sloping edges of faces (raytrace and z-buffer).
- Area sampling of shader patterns if the shader also has the **Area Sampling** option selected. **Shader Area Sampling** is discussed in more detail in Section 6.2.2 (raytrace and z-buffer).
- Additional antialiasing on surfaces, wherever there are areas of strong contrast, such as along the boundary of hard shadows (raytrace only).
- Antialiasing of areas of strong contrast in mirrored reflections and glass like refractions (raytrace only).

**Super Sampling:** When this option is selected, the entire image is super sampled. For each pixel on the screen, additional samples are rendered. The amount of samples taken per pixel depends on the option selected from the menu. **Low** renders a grid of 2 by 2, **Medium** 3 by 3, and **High** 4 by 4 samples. Note that super sampling an image leads to increased rendering times. For example, if **High** is used, 16 additional samples are rendered per pixel. This may increase the rendering time by a factor of 16.
6.1.3 Illumination

The options in this group of the Shading tab in the RenderZone Options dialog control the illumination of the scene that is being rendered.

**Light Intensity**: The intensity of lights in a scene can be specified in three ways: as a basic global intensity expressed in a percentage figure, globally in physically accurate units, or these options can be applied separately per light. The **Light Intensity** menu contains three items as follows:

- **All Simple**: When this item is selected, simple light intensity, which is defined as a percentage, is used for all the light sources in a rendered scene. Selection of this item overrides any intensity selection made individually for the lights.

- **All Accurate**: When this item is selected, accurate intensity is used for all the lights. Accurate intensity is defined using physically accurate intensity units. Selection of this item overrides any intensity selection made individually for the lights.

- **Per Light**: When this item is selected, the intensity type set for each individual light source is used. Whether a light emits with simple or accurate intensity is selected in the **Light Parameters** dialog, as described in section 6.6.2.

**Area/Line Lights**: The three options in this pop up menu determine how area/line lights will be treated in the rendering. Note that rendering area and line lights can significantly affect rendering speed, which is why the options in this menu allow you to bypass them, especially during preliminary rendering stages. A **Quality** slider bar for area and line lights can be found in the **Light Parameters** dialog when an area or line light is selected. Depending on the setting of the slider, the quality of illumination and shadows generated by area and line lights can be controlled, at the expense of rendering time. Area and line lights are discussed in detail in section 6.6.

- **Ignore**: When this option is selected, the area/line lights that may be in a scene are ignored and not included in the rendering.

- **Approximate**: When this option is selected, any area/line lights encountered in a scene are processed by RenderZone* as point lights.

- **Use**: When this option is selected, area/line lights are processed as area/line lights. Selecting this particular option affects the rendering speed as mentioned above.

**Light Glow**: Selecting this option allows you to globally control the use of glowing lights in a rendering. For glowing lights to be rendered, the glow options for each individual light must be turned on in the **Light Parameters** dialog, in addition to selecting this option. Deselecting this option will disable the light glow effect for all lights, including those for which it has been individually set.
**Precompute Exposure Correction**: When this option is selected, the intensity of the final rendering is corrected automatically to not produce any over or underexposed images. This is achieved by first measuring the level of exposure of the image before it is rendered. This step happens before the first scanline is rendered and may take a short while to complete. Once the actual image is rendered, areas which would otherwise be overexposed will be corrected to show better illumination. Likewise underexposed areas are rendered with increased illumination. The purpose of this option is to avoid a rendering which is completed washed out with strong light or completely dark because of lack of lighting. It may be necessary to further fine tune the final image exposure using the **Exposure Correction** option of the **RenderZone Postprocess** feature. This is discussed in more detail in section 6.1.10.

**Analysis**: If this option is selected, an illumination analysis is available. With this option on, the current rendering will not show the colors from the surface styles associated with each object in the scene, but the light intensity on a surface is visualized using a color coding scheme. In architectural design it is a common task to design the lighting of a space so that certain illumination criteria are met for the activities performed in the space. For example, in an office space light intensities on the desktop surface should be around 250 Lux. Properly lit scenes rendered with accurate intensities can show whether such lighting criteria are met. Recall that accurate light intensities are used by choosing **All Accurate** from the **Light Intensity** menu. In this example, surfaces which are 250 Lux or higher can be rendered in a specific set of colors which immediately show if the criteria have been met. The resulting image is not unlike a picture taken with a heat sensitive camera as shown in Figure 6.1.3.1.

**Figure 6.1.3.1**: A rendering with (a) no analysis, (b) **Illuminance**, and (c) **Luminance**.
Clicking on the **Options**... button next to **Analysis** invokes the **Illumination Analysis Options** dialog shown in Figure 6.1.3.2. The colors to be used for the light intensities are set in this dialog.

**Color/Value**: This list on the left of the dialog contains colors assigned to different values. These colors are used to represent the respective value when the illuminance or luminance diagrams are created. If the intensity of a surface pixel is no larger than the closest value in the list, it is rendered with the color shown next to the value. For example, a pixel with intensity of 650 would render in red, according to the options shown in Figure 6.1.3.2. A value in the list can be edited by clicking on the text in the highlighted row. This will display the value in the text field next to the list where it can be edited. The color of the selected item in the list is shown in the color field next to the list. Clicking on it invokes the standard Color Picker dialog.

**Illuminance**: If this option is selected, the rendering shows the illuminance of all surfaces. Illuminance is the amount of light intensity received on a square unit area and is measured in Lux. Rendering the illuminance of a scene gives a measure of how much light is received on a surface and can be used to determine proper lighting conditions in a space.

**Luminance**: If this option is selected, the rendering shows the luminance of all surfaces. Luminance is the amount of light intensity reflected by a surface, also measured in Lux. Rendering the luminance of a scene gives a measure of how bright a surface appears to an observer. It may be used for example to find areas with a high degree of glare.

**Figure 6.1.3.2**: The **Illumination Analysis Options** dialog.

<table>
<thead>
<tr>
<th>Color</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Color</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td></td>
</tr>
<tr>
<td>900</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 6.1.3.3**: A rendering (a) with and (b) without blended colors.
Blend Colors: If this option is selected, the color value of a pixel is interpolated smoothly between the colors of the closest color/value pair. An example of a rendering with and without blended colors is shown in Figure 6.1.3.3.

Set Range...: Clicking on this button invokes the Illuminance/Luminance Range dialog (Figure 6.1.3.4), where the values in the Color/Value list can be set using uniform increments. If the maximum value is specified, the values in the list are divided into equal increments, starting at the minimum and ending at the maximum. If Increment is selected, the values in the list start at the minimum and are incremented in each row by the increment value. Note that after the increments are set uniformly through this dialog, they can still be changed individually by editing them.

Insert: If this button is selected, a new color/value pair is inserted above the current row.

Delete: If this button is selected, the current row is deleted.

Save...: If this button is selected, a set of color/value pairs can be saved to a text file.

Load...: If this button is selected, a previously saved file with color/value pairs can be loaded.

Default: If this button is selected, the default set of color/value pairs is loaded.

Note that there is a significant difference in rendering illuminance versus rendering luminance. For example a black surface that absorbs most of the incoming light may have a high degree of illuminance, but has very little luminance, since no or little light is reflected back into the environment.

For accurate studies of the luminance/illuminance of a scene it is necessary to execute a reasonably complete radiosity solution. Radiosity accurately simulates the light reflecting off surfaces and illuminating other surfaces in the scene. This secondary illumination is necessary to accurately describe how a scene is illuminated in areas which are not reached by light sources directly. A RenderZone rendering which is not based on a radiosity solution, although using accurate light intensities, only applies a constant ambient light in areas not illuminated directly or a one bounce light reflection through Final Gather. Thus, a color analysis of this rendering will not be accurate. However, it may be sufficient to get a basic impression of whether the light sources in the scene are defined with the proper intensities.
6.1.4 Global illumination and its tab

What is global illumination

In a standard rendering, a surface is normally lit by *direct illumination*. For a given point of a surface a ray is cast in a straight line (directly) toward a light source. If the ray intersects any other objects on its way to the light source, the point is in shadow and no illumination occurs. If the ray reaches the light source without obstruction, the distance of the point to the light source and the light’s intensity, falloff and color determine the illumination of the surface. Figure 6.1.4.1 illustrates a simple example. The scene shows a room with a table and a few spot light sources shining down from the ceiling and the wall. These light sources reach the walls and table top directly, providing adequate illumination. However, the floor below the table, upper walls and the ceiling are not reached by the light and subsequently become uniformly dark.

In a real world scenario a point on a surface is not only reached by light directly, as described above, but also by light, which is reflected off other surfaces in the scene. This effect is called *indirect illumination*. From a rendering standpoint, direct illumination can be calculated rather quickly. Indirect illumination, due to its complex nature, is much more difficult to compute. Looking at the simple scene, it is easy to see that a point on the ceiling can really be reached by an infinite number of reflected light rays. Computer graphic has therefore developed a series of different algorithms, which simulate this complex phenomenon of bouncing light more or less precisely. These algorithms as a whole are referred to as *global illumination* (or GI for short), since they consider all light that exists globally in a scene, rather than only the light, which reaches a point on a surface directly. The simple scene from above has been rendered using GI algorithms in Figure 6.1.4.2. One can see, that the areas previously rendered uniformly dark now receive some light and show subtle shading effects, which gives it a much more realistic look.

![Figure 6.1.4.1: A scene with direct illumination only.](image1)

![Figure 6.1.4.2: A scene with global illumination.](image2)
form•Z supports GI through three rendering and illumination algorithms: ambient occlusion, final gather, and radiosity. Each of these algorithms can be used separately, but give the best results when applied in combination with each other. In other words, rather than computing the same effect, each algorithm provides a complimentary part of calculating global illumination in a scene. Depending on the type of scene and how much computation time can be afforded, it is quite possible to only use one or two of the methods and still get very good rendering results.

In addition to these methods form•Z offers special light sources that are considered part of global illumination. They are the Environment light type and the Atmospheric Light option of the Distant light. One can also consider the Ambient Light, which is set in the Lights dialog, as part of global illumination. Although it is the most inaccurate representation of bouncing light, it can be used effectively in conjunction with ambient occlusion.

The Global Illumination tab

The options in this tab of the RenderZone Options dialog determine whether a global illumination method will be applied and which one. The Global Illumination tab is shown in Figure 6.1.4.3. Its options are as follows:

Use Ambient Occlusion: When this option is on, ambient occlusion is applied to a rendering. See section 6.1.5 for a detailed discussion of this global illumination method.

Use Final Gather: As above, when this option is on, final gather is applied to a rendering. See section 6.1.6 for a detailed discussion of this global illumination method.

Both these options share the following two options:

Scene Size: This pop up menu contains 7 items: Small Object (6’), Large Object (3’), Room (20’), Building (100’), City (2000’), From View, and Custom, as shown in Figure 6.1.4.4. Next to the pop up menu is a numeric field that displays the numeric value corresponding to each item. For the top 5 items this value is fixed, not editable, and is as shown in the parentheses next to the items, above.

Figure 6.1.4.3: The Global Illumination tab of the RenderZone Options dialog.
**From View** is a value corresponding to the diameter of the cone of vision pyramid at the center of interest of the current view. In other words, this is the width of a scene at the view's center of interest. While this number is not editable, it varies as the view changes. For the last item, **Custom**, while its value initially defaults to 30', it is editable and it is intended to be set by the user, whenever a value other than those represented by the other menu items is desired.

It is very important to choose the appropriate item in the **Scene Size** menu as both **Ambient Occlusion** and **Final Gather** automatically determine certain settings based on the size of the scene being rendered. While it is not so critical to get the dimension exactly correct, the scene should roughly correspond to the size chosen. For example, if a view showing a room is 30' wide, it is still OK to choose the **Room** item of the **Scene Size** menu, which sets the dimension to 20 feet. It is important not to choose **Small Object** or **City**.

**Quality**: This pop up menu contains 5 levels of quality ranging from 1 (low) to 5 (high) and an item labeled **Custom**, which allows you to set detailed **Quality** parameters for **Final Gather** and **Ambient Occlusion** manually (Figure 6.1.4.5). The higher the quality chosen from the menu, the longer the rendering will take, but the better it will look. Quick preview images can be derived with level 1 or 2 rendering, while higher settings will produce more photorealistic images.

**Show Details**: When this button is pressed, detailed parameters for **Ambient Occlusion** and **Final Gather** are shown in their respective tabs. By default, these detailed parameters are hidden and only the parameters that are essential to the behavior of the algorithm are shown. All of the hidden detailed parameters are still computed automatically based on the settings of the **Scene Size** and **Quality** menus. For the majority of the scenes rendered, it is not necessary to worry about the exact values of these parameters. However, should an experienced user wish to fine tune a rendering, the detailed parameters can be displayed and edited by pressing the **Show Details** button, which changes to **Hide Details**. Likewise, the detailed parameters can be hidden by pressing the button again, which reverts back to **Show Details**. If a user chooses to change any of the detailed parameters from the pre-computed defaults, the **Quality** menu changes to **Custom**, indicating that the parameters have now been edited. It is also possible to choose the **Custom** item from the **Quality** menu directly, in which case the detailed parameters are also revealed.
6.1.5 Ambient occlusion and its tab

Ambient occlusion is the simplest of all GI methods. It can be used very effectively for interior scenes to quickly add contrast to areas, where no direct illumination occurs. This method assumes that there is uniformly distributed ambient light in a scene. One could think of this ambient light as all the light which is reflected from surfaces and which is scattered equally strong in all directions. When a surface point is illuminated, ambient occlusion determines whether there is any obstruction close by, which would prevent some of the ambient light from reaching the point. If this is the case, the point receives less ambient light, and therefore is rendered darker. If there is no obstruction, it is assumed that all of the ambient light reaches the point and therefore it is rendered lighter. This method of occluding the point from the ambient light requires that a number of randomly directed rays are cast from the point and tested whether they intersect any objects within a given distance. The result is that in corners of a room or in tight spaces, the image becomes darker, whereas large unobstructed surfaces are rendered lighter. This is illustrated in a simple scene in Figure 6.1.5.1(a), which is rendered without ambient occlusion, and Figure 6.1.5.1(b), which is rendered with ambient occlusion.

Figure 6.1.5.1: A scene rendered (a) without and (b) with ambient occlusion.
The Ambient Occlusion tab

The options in this tab become active when Use Ambient Occlusion is on in the RenderZone Options dialog (Figure 6.1.5.2). They are as follows:

**Ambient Light:** The two options in this group (Color and Intensity) are the same as in the Lights dialog. They are repeated here in order to become more accessible as different global illumination parameters may be tried.

**Distance:** This pop up menu contains four items: Small, Medium, Large, and Custom. The option selected from this menu determines how far away from a point another object can be to qualify as an occluder. If one of the top three items is selected, the distance is set automatically, depending on the selection from the Scene Size menu. It is also possible to enter a different value by choosing Custom from the menu. One can think of the distance value as an indicator of how far from a corner the dark occluded region will reach into the illuminated surface. The larger the value, the further the dark region will stretch into light areas. A good value for interior scenes is a distance that is about half the ceiling height.

**Contrast:** This slider determines how dark a point becomes, if it has maximum occlusion. When set to 100% and a point is completely occluded, it is rendered as dark as if there were no ambient light. When set to 50%, the same point gets 50% of the ambient light. In other words, with this slider it is possible to determine how dark occluded areas become.

**Falloff:** This slider allows for additional control of how quickly the dark region in a corner becomes lighter as it reaches into a surface. If it is set to 0%, the dark region stays dark further out, giving the impression of stronger contrast. If it is set to 100%, the dark region becomes light very fast and only very tight spaces remain dark. For most scenes, the default value of 50% works well.

**Additional Contrast:** This slider adds extra darkness to occluded areas. By default it is set to 0%, which produces no additional contrast. If it is set to 100%, a fully occluded point would render black. Values between 0 and 100 produce analogous darkness.

**Add Noise:** When this option is on, ambient occlusion produces a subtle stipple pattern that enhances the hand drawn look of a rendering.

![Figure 6.1.5.2](image-url): The Ambient Occlusion tab with details shown.
Detailed parameters

As already mentioned, pressing the **Show Details** button reveals additional detailed parameters, which, for **Ambient Occlusion**, is only one:

**# Of Rays:** The value entered in this field determines how many rays are cast from each point for the calculation of how much ambient occlusion occurs. The lowest quality sets this value to 36, whereas the highest quality sets it to 196. Note that these numbers are the squares of 6 and 14. The algorithm always uses a square value. Therefore if a non square number is entered, it is automatically rounded to the next square number. For example, entering 130 or 140 will make no difference as, for both, 144 is the next square number, which will be used. Only on rare occasions will it be necessary to set this parameter to values above 196. It is unlikely that increasing the parameter from 300 to 600 will make any difference, other than the rendering taking twice as long to complete.
How to use ambient occlusion

There are two different scenarios, where ambient occlusion can be helpful. The first is a scene where there is not enough light, such as an interior, where only sun light shines through windows. A hidden line rendering of such a scene is shown in Figure 6.1.5.3. Improving this rendering with ambient occlusion is a simple three step process.

The first step is to render the scene with just the sun light and Ambient Light set to 0%. This rendering is shown in Figure 6.1.5.4.

In the second step the Ambient Light is set to a much higher value. It may require one or two quick test renderings, but the goal is to determine, how intense the brightest (least occluded) surfaces in the scene should appear. Typically, the Ambient Light is set to values between 50 and 80%. The rendering in Figure 6.1.5.5 has Ambient Light set to 70%.

For the first two renderings it is OK to use methods for creating fast renderings. For example, one can use a small window, turn off layers with objects that are not critical to the rendering, but have a high number of faces, and/or turn off Reflections and Transparencies in the RenderZone Options dialog. The reason for these two renderings is to determine how the scene looks without any ambient light and with maximum ambient light.

In the third step, ambient occlusion is added. Any shortcuts for faster renderings in step 1 and 2 should now be reversed. In the RenderZone Options dialog, Use Ambient Occlusion is selected, in the Global Illumination tab. The appropriate items are selected from the Scene Size and Quality menus. The default quality level of 3 works well for average scenes. More detailed parameters can be found and set in the Ambient Occlusion tab (Figure 6.1.5.2).
Recall that the **Contrast** slider determines how dark a point becomes, if it has maximum occlusion. When set to 100% and a point is completely occluded, it is rendered as dark as if there were no ambient light, as in step 1. When set to 50%, the same point gets 50% of the ambient light that was added in step 2. In other words, with the **Contrast** slider it is possible to determine how dark occluded areas become. The renderings in step 1 and step 2 represent the extremes, full ambient illumination and no ambient illumination.

The option selected from the **Distance** menu determines how far away from a point another object can be to qualify as an occluder. If **Small**, **Medium**, or **Large** is selected, the distance is set automatically, depending on the selection from the **Scene Size** menu. It is also possible to enter a different value by choosing **Custom** from the menu. One can think of the **Distance** value as an indicator of how far from a corner the dark, occluded region will reach into the illuminated surface. The larger the value the further the dark region will stretch into light areas. A good value for interior scenes is a distance that is about half the ceiling height.

For example, for a room with 10' ceilings, 6' works well. It is best to choose the **Large** option from the menu for interiors. Figure 6.1.5.6 shows the scene now rendered with ambient occlusion, using 100% **Contrast** and the **Large Distance** option. Note that the corners of the room now show darker regions, whereas the large wall areas remain bright. The darkest area is inside the shelf on the bottom right of the image, since the small openings have the most occlusion.

A second scenario may be a scene that already has sufficient light before adding any extra ambient light. This may be the case in a rendering of an exterior of a building. Such a rendering may suffer from the lack of contrast in areas of small detail, as is shown in Figure 6.1.5.7(a).

The sun light illuminating the scene causes equally bright areas everywhere, which makes it difficult to distinguish elements on the facade. Since little or no ambient light was added to the scene, the **Contrast** slider cannot be used. For this scene, the **Additional Contrast** slider needs to be increased. Instead of illuminating a point between no and maximum ambient light, the **Additional Contrast** simply adds extra darkness to occluded areas. A fully occluded point would render as black, if the **Additional Contrast** is set to 100%. The exterior scene shown in Figure 6.1.5.7(b) was rendered with the **Distance** set to **Medium** and **Additional Contrast** set to 50%.

---

**Figure 6.1.5.6**: A scene rendered with ambient occlusion.
The **Falloff** slider allows for additional control of how quickly the dark region in a corner becomes lighter as it reaches into a surface. If it is set to 0%, the dark region stays dark further out, giving the impression of stronger contrast. If it is set to 100%, the dark region becomes light very fast and only very tight spaces remain dark. For most scenes, the default value of 50% works well. The interior scene from earlier, rendered with 50% **Falloff**, is shown again with values of 0% and 100% in Figures 6.1.5.8(a) and (b).
Controlling ambient occlusion on individual surfaces

The amount of ambient occlusion that occurs on a surface is directly influenced by the **Ambient Reflection Factor** parameter of the **Reflection** shader of the surface style that is assigned to an object or face. By default, the reflection shaders that represent rough and opaque materials, such as **Matte** or **Plastic**, have their **Ambient Factor** set to 100%. In this case, ambient occlusion is fully added to such a surface. However, a material that is highly transparent or very shiny is hardly affected by ambient light. These reflection shaders have their **Ambient Factor** set to lower values by default. Therefore ambient occlusion affects these surfaces to a lesser degree. If a reflection shader has its **Ambient Factor** set to 0% or is missing the **Ambient Factor** parameter, no ambient occlusion will be shown on a surface. The one exception is the **Catcher** shader. Although it does not have an **Ambient Factor** parameter, it determines the amount of ambient occlusion through its **Shadow Intensity** slider.

![Plastic Options](image1.png)

![Accurate Glass Options](image2.png)

**Figure 6.1.5.9**: The default **Ambient Reflection** parameter of (a) **Plastic** and (b) **Glass**.
Whether an object receives ambient occlusion and causes occlusion on other surfaces can also be controlled by an object level attribute. By default a new object always receives and causes ambient occlusion. The defaults can be changed by using the Render Attributes tool. It contains two options, Receives Ambient Occlusion and Causes Ambient Occlusion as shown in Figure 6.1.5.10. The same setting are also accessible through the Query Object Attributes dialog. If an object is set to not receive ambient occlusion, it will not show any shading effects from AO. However, it will still act as an occluder for other surfaces. If the object is set to not cause ambient occlusion, it will be ignored as an occluder. Note, that setting an object to not receive ambient occlusion will make a rendering faster, whereas setting it to not cause ambient occlusion makes it slightly slower.

Ordinarily it is not necessary to change these two setting form their defaults. Only if the globally applied ambient occlusion has an undesired effect in parts of a scene, should those settings be used.

Figure 6.1.5.10:
The Ambient Occlusion settings in the Render Attributes Options dialog.
Ambient occlusion in abstract renderings

While ambient occlusion can greatly enhance the realism of a rendering, it is also very useful in producing abstract, sketch-like illustrations. For example, it is possible to create an architectural drawing that appears to be pencil rendered. In such an image all the surfaces are colored white and only shadows and shading is used to articulate forms and spaces. Without ambient occlusion, these white surfaces would all blend together and reveal little or no detail, as shown in Figure 6.1.5.11(a). The surface style used on all objects has a pure white color and the default Matte reflection shader. To make all surfaces appear bright, Ambient Light is increased to 80%. Ambient Occlusion is turned on with 100% Contrast and 30% Additional Contrast to give a more dramatic occlusion effect. Finally the Add Noise option is turned on. This causes ambient occlusion to produce a subtle stipple pattern, which enhances the hand drawn look of the rendering, as shown in Figure 6.1.5.11(b).

Performance and limitations

Since ambient occlusion relies heavily on raytracing techniques, the same rules apply to speed it up as do for a rendering using the raytrace rendering method. In general it is recommended to use the raytrace rendering type whenever ambient occlusion is also used. If multiple processors are available, the raytrace rendering type will typically be faster than the z-buffer mode, since it takes advantage of multi processing, whereas z-buffer does not. In addition, ambient occlusion used with the z-buffer mode may show artifacts along edges of faces as shown in Figure 6.1.5.12. These can be avoided using the raytrace render mode instead.
Tutorial

This tutorial illustrates how to quickly improve the illumination of an interior rendering using ambient occlusion.

• Default rendering
  - Load sample file AO_tutorial.fmz.
  - Select RenderZone* from the Display menu.

The rendered image (Figure 6.1.5.13) shows an interior scene illuminated by the sun shining through a series of windows and skylights. In order to simulate the effect of the incoming light bouncing off the floor and walls, a number of fill lights are also set up. These are point lights that distribute extra light into areas that are not reached directly by the sun. This is a common technique when we wish to get a quick effect of bouncing light. However, it is also quite inaccurate and the image does not look very realistic.

• Establishing ambient brightness

In the next step, we will turn off the fill lights.

  • In the Lights palette, click in the header bar next to “Light Name” to open the Lights dialog.
  • Click in the Shining column behind the light group called Fill Lights. This will turn off the shining attribute of those lights. Also turn the Ambient Light Intensity all the way down to 0% (Figure 6.1.5.14).
  • Click OK to exit the dialog.
Now the scene will be rendered only with the direct illumination of the sun light (Figure 6.1.5.15). Notice that all surfaces that are not reached by the sun are black. The purpose of this rendering is to determine how dark the scene is without any ambient light.

- Open the Lights dialog as before. Increase the Ambient Light Intensity to 70% and exit the dialog. The subsequent rendering shows the scene illuminated with a large amount of ambient light, as shown in Figure 6.1.5.16. You may want to repeat this step a few times with different values of the ambient light intensity. The goal is to find a brightness that will render the areas that were previously black, so that they appear with their maximum desired intensity.

The two renderings, dark and bright, serve as the extremes between which ambient occlusion will choose and create soft shadows and gradual shading.

Figure 6.1.5.15: The scene rendered only with the direct illumination of the sun light.

Figure 6.1.5.16: The scene illuminated with a large amount of ambient light.
• Adding ambient occlusion

• Invoke the RenderZone Options dialog.

• Select the Global Illumination tab. In it choose the Use Ambient Occlusion check box.

• Set the Scene Size menu to From View (Figure 6.1.5.17).

• Select Large from the Distance menu in the Ambient Occlusion tab.

• Click OK to close the dialog and complete another rendering, as shown in Figure 6.1.5.18.

Notice, that the corners and niches in the scene are now showing a gradual falloff from the level of darkness established in the black rendering to the level of illumination created in the bright rendering. Areas, that are very tight, such as the shelf below the window, create the most occlusion and are rendered the darkest. Areas, where there are no objects in front and are therefore not occluded are rendered with the maximum brightness.

The two major parameters that control the shading in this example are the Contrast and Distance options in the Ambient Occlusion tab in the RenderZone Options dialog. At 100% Contrast, the maximum darkness that can be reached is that of the black rendering, which would be in areas that are entirely occluded. If Contrast is set to a lower number, the maximum level of blackness would be less. In other words, the Contrast parameter determines how dark the shading becomes. The Distance parameter determines how far from a dark corner the gradual shading reaches.

Notice that in a corner the shading slowly bleeds into the open wall. The Distance of about 10’ chosen for our example causes this shading to reach no further than this distance. In the rendering it actually appears much less, since the corner is not fully occluded and the lower levels of shading, further out in the open wall, are not as noticeable as the dark areas closer to the corner. 10’, roughly the distance from floor to ceiling, is a good value for interior architectural scenes.
6.1.6 Final gather and its tab

While ambient occlusion computes how much ambient light does not reach a given point, the final gather algorithm calculates how much of the reflected light in a scene reaches a point. Recall that increasing the Ambient Light slider in the Lights dialog makes the scene uniformly lighter, because it is assumed that this ambient light is equally strong everywhere. Final gather on the other hand creates ambient light by distributing it non uniformly. It does this by placing many sample points in the scene before a rendering is started. For each sample point final gather calculates how much indirect illumination the point receives from light bouncing off other surfaces. The sample points are clustered closer together in areas where there is a sharp change of contrast and are spread further apart where the illumination does not change much. After this first stage of calculating the final gather sample points is complete, a rendering is started. Each pixel is lit by direct illumination, as in a regular rendering. To compute indirect illumination for the pixel, the final gather sample points are used. Instead of calculating the amount of indirect light hitting the pixel precisely, final gather collects (gathers) the pre computed indirect light from close by sample points and averages the indirect illumination for the pixel from those samples. More sample points may be added to the scene even during a rendering, if it turns out that there are not enough sample points close enough to a pixel to average adequate indirect illumination. After such a rendering is complete, the final gather sample points are maintained and do not have to be recalculated anymore for subsequent renderings, as long as no changes occur to the objects or lights in a scene. If only the view is changed, the samples are also maintained, but new samples may be added. In general, any rendering after the first that calculated the final gather samples is faster.

One can also think of final gather as simply doing a better job of calculating the ambient light. It will make it brighter in areas where more light is reflected from adjacent surfaces and darker in areas with little reflected light.

When to use final gather

Final gather can greatly improve the realism in a rendering through a better distribution of indirect light. While ambient occlusion is a first, quick step for better illumination, final gather adds further improvements. For a final rendering it is usually advisable to also use ambient occlusion together with final gather, since both methods compute complimentary information. Final gather works best in scenes, that have a reasonable amount of light, but are not completely flooded with light. For example, a rendering of an interior with typical light fixtures or sun light shining through openings benefits greatly from final gather. An exterior rendering, that is well illuminated by sun light may show only little change in illumination when final gather is used, although it may still be beneficial. Figure 6.1.6.1 shows an interior scene with just direct illumination and some constant ambient light. Figure 6.1.6.2 shows the same scene after final gather was added. Ambient occlusion was not used in either rendering to highlight the improvement added by final gather alone.
Figure 6.1.6.1: An interior scene rendered with direct illumination only.

Figure 6.1.6.2: The scene after final gather was added.
The Final Gather tab

The options in this tab become active when **Use Final Gather** is on in the **RenderZone Options** dialog (Figure 6.1.6.1). They are as follows:

**Level Of Detail:** A selection made from this pop up menu determines the size of objects that are desired to cast shadows from indirect illumination. It contains five items: **1** (Coarse), **2**, **3**, **4** (Fine), and **Custom**. A numeric field next to the menu displays the size value that corresponds to the selected item. The top four items are levels that are set automatically, while the last item offers the option for the user to enter a value.

**Add Radiosity:** When this option is on, a radiosity solution will be added to the final gather procedures. Exactly how much is determined by the following options.

- **Time:** The value entered in this field determines for how long the solution is processed.
- **Time Spent:** The value in this non editable field indicates the time that has been spent on an existing radiosity solution, if there is one. To add more time, the value entered in the **Time** field must be greater than the time shown in this field.
- **Use Bounding Box:** When this option is on, the bounding box set up with the Radiosity Bounding Box tool is used.
- **Keep Solution:** When this option is on, the radiosity solution calculated for the final gather rendering is saved, after the rendering is finished. If no changes to objects or lights are made and another final gather rendering is made, the radiosity solution is reused. When this option is off, a new radiosity solution needs to be calculated for each rendering.

**Specular Illumination:** This menu contains three items that determine how specular effects are calculated during a final gather based rendering. They are **Direct Without Dome Lights**, **Direct With Dome Lights** and **Indirect From Final Gather**. The first creates specular effects only from direct lights as in a regular rendering and also ignores specular effects from dome lights. It is the option with the fastest rendering speed. The second item works as the first, except that specular effects are added from direct illumination from dome lights. The last option calculates indirect specular effects through final gather, which makes the image most accurate, but also takes the longest to compute. The last option requires that the **Calculate Specular Final Gather** parameter of a surface style’s reflection shader is also turned on. This is discussed in more detail later in this section.

**Scale Output:** The indirect illumination calculated by final gather may be artificially amplified (or reduced) by increasing (or decreasing) this parameter. It should not be abused, however. Setting it to values above 200% to brighten an underexposed image may create unrealistic results.
**Color Saturation**: When light bounces off a surface, it will be filtered by the color of that surface. For example, when light bounces off a red surface, it becomes pink. Final gather also creates this effect. This parameter determines how strong the coloring of the reflected light will be. At 0%, the reflected light will not be colored by the surface. At 100%, the coloring effect will be most dramatic.

**How to use final gather**

To test how final gather affects a scene, it is advisable to preform a few quick test renderings. The image generation can be sped up by reducing the window size, turning off reflection, transparencies and other time consuming effects, before adding final gather. With more experience, this step may be omitted and a final rendering can be attempted right away.

As already mentioned, the final gather method is turned on by selecting the **Use Final Gather** check box in the **Global Illumination** tab of the **RenderZone Options** dialog. As with ambient occlusion, it is important to choose the correct item from the **Scene Size** menu. This selection is even more critical for final gather, as the wrong choice may result in poor renderings, or renderings that take too much time for little gain in quality. From the **Quality** menu, a low setting should be chosen first for a simple test rendering. In the **Final Gather** tab, the **Level Of Detail** menu needs to be set to the appropriate value. This menu sets a size parameter, which determines how large an object or part of an object should be in order to cast shadows from indirect illumination. The smaller this size, the more work final gather has to do and the longer the rendering will take. The menu offers four automatic levels and the option to enter a custom value. Note that the sizes of the automatic levels are determined based on the setting of the **Scene Size** menu. Therefore, if one renders a scene that is, for example, 100’ feet wide, but **Small Object** is selected from the **Scene Size** menu, the automatic levels are set to values that are much too small, which will result in renderings that will take unnecessarily long. It is important to understand that the level of detail is not necessarily the smallest size in the scene that could cast shadows from indirect light.

For example, in a rendering of a room with furniture, it is not important to have final gather calculate shadows cast from a thin rail in the back of a chair, but rather the whole chair. An appropriate level of detail would therefore be 12”, rather than 1”. Subtle detail of contrast can be calculated much more efficiently by ambient occlusion, which is usually added to a final gather rendering. Figure 6.1.6.4 shows a rendering of an interior without final gather. Figure 6.1.6.5 has final gather turned on. For this rendering the **Scene Size** is set to **Room** (20’), the **Quality** is set to 2, and **Level Of Detail** is set to 2. This results in a quick rendering, sufficient to evaluate the overall brightness of the scene.

For the final rendering in Figure 6.1.6.6, **Ambient Light** was increased from the default 10% to 40%, since the first final gather rendering shows that more light needs to be added to the scene. The final rendering also has **Ambient Occlusion** turned on with 100% **Contrast** and 25% **Additional Contrast**. The **Final Gather** settings were increased to **Quality** 3 and **Level Of Detail** 3. The indirect illumination calculated by final gather may be artificially amplified by increasing the **Scale Output** parameter in the **Final Gather** tab. This parameter, however, should not be abused. Setting it to values above 200% may create unrealistic results.
Figure 6.1.6.4: An interior rendered with one distant light and direct illumination only.

Figure 6.1.6.5: The same scene rendered with low quality final gather.

Figure 6.1.6.6: The scene rendered with quality 3 final gather and ambient occlusion.
Improving final gather by using dome lights

There are two light types referred to as dome lights: the Environment Light and the Atmospheric Light option of a Distant light. Both lights use a large number of individual light samples to illuminate a scene. When used in a regular rendering the number of samples needs to be set to a high value to avoid strong artifacts. Unfortunately, a high number of samples also causes a rendering to slow down dramatically, since every pixel in the image needs to be illuminated by each sample of the light. Figure 6.1.6.7 shows a regular rendering using a Distant light with the Atmospheric Light option selected and a low number of samples (100). The artifacts are very apparent. The same scene is shown in Figure 6.1.6.8, but with the # Of Samples set to 2000. The image in Figure 6.1.6.8 took 20 times longer to compute than the one in Figure 6.1.6.7. In a final gather based rendering, the artifacts seen in Figure 6.1.6.7 can be avoided without having to raise the number of samples. Therefore, it normally makes sense to always use the Atmospheric Light option of a Distant light in a final gather rendering. Using this light serves two purposes: First, the image quality is improved by the mere presence of the atmospheric light, which will cause soft shadows and more light to enter the scene. Second, the final gather process has more light to work with.

Recall that during the sampling phase of final gather, a sample point receives indirect illumination from surfaces around the scene. However, if there are only a few bright surfaces, even final gather can not improve the illumination of the scene much. An atmospheric light, however, adds more light and subsequently final gather does a better job distributing this light further into the scene. This is especially useful in an interior scene, where the distant light shining through a relatively small opening illuminates only a small area on the floor or wall. Going back to the initial example of Figure 6.1.6.6, adding the atmospheric light option with an intensity of 1000% contributes much more light and gives final gather a chance to create even better indirect illumination. Another quick preview rendering with such a light is shown in Figure 6.1.6.9. Comparing it to Figure 6.1.6.5, which was rendered without atmospheric light, one can see that much less (if any) additional ambient light needs to be added to achieve a good final scene brightness. The image shown in Figure 6.1.6.10 is a complete GI rendering with final gather, ambient occlusion, and atmospheric light. The Quality of the GI rendering was set to 4.

The environment light works in essence like the atmospheric light. However, it is its own light type, rather than an option of the distant light. HDRI images can be used effectively to provide global light, that is concentrated around bright areas in the image. This works well for rendering objects in an exterior setting. Two images of an exterior scene are shown in Figure 6.1.6.11 and Figure 6.1.6.12. The first is rendered without final gather or environment light, the second uses both.
Figure 6.1.6.7: A scene rendered using atmospheric light with 100 samples.

Figure 6.1.6.8: The same scene rendered with 2000 samples.

Figure 6.1.6.9: A low quality final gather rendering using 100 samples.

Figure 6.1.6.10: A high quality final gather and ambient occlusion rendering.

Figure 6.1.6.11: A dune buggy rendered with a distant light and direct illumination only.

Figure 6.1.6.12: A dune buggy rendered with an environment light and final gather.
Improving final gather by using radiosity

Radiosity is the third of the global illumination methods mentioned at the beginning of this section. It is by far the most complex algorithm and, as such, it is discussed in more detail later. At this point, only the basic concepts are introduced and discussed as they relate to final gather.

Radiosity computes how light travels throughout a scene. It does so by first dividing all objects in the scene into a reasonably dense mesh of triangle patches, called the radiosity mesh. Each light source in the scene then illuminates each patch in a first phase. This calculates the direct illumination of the scene. The second phase computes the indirect illumination. After the first phase, the patch that has received the strongest direct illumination sends the portion of the light that is not absorbed (i.e. which is reflected by the material), back into the scene, illuminating all other patches. After that the second strongest patch illuminates all other patches. This phase continuously cycles, where the strongest patch, not having illuminated the scene, sends its light back into the scene. Little by little more light travels throughout the scene. If this cycle continues long enough a very accurate distribution of indirect light in an environment can be derived. This is called a radiosity solution. The radiosity algorithm has two major disadvantages that make it impractical as the base for a rendering. First, the radiosity mesh that records the light intensities, when rendered, inevitably shows artifacts. Second, a radiosity solution may take a long time to distribute enough light in a scene to give it enough brightness, especially if the scene is complex.

When used in conjunction with final gather, both of these disadvantages can be offset. If a radiosity solution has been calculated and a final gather based rendering is executed, the radiosity mesh itself is not shown. Instead, the mesh adds more final gather samples with indirect illumination. The radiosity solution simply contributes more light for final gather to work with. Therefore, the artifacts that would otherwise result from rendering a radiosity mesh will not be present in a final gather rendering that uses radiosity. A radiosity solution also does not have to run for a very long time to contribute some extra light to a scene. Recall, that final gather starts from a point to be illuminated and looks backwards (with respect to the direction in which light travels in a scene) to determine how much light reaches the point from other surfaces. Radiosity on the other hand, distributes light in a forward direction. Even a few cycles of radiosity send some light into the scene. Together, the backward looking of final gather and forward distribution of light by radiosity can result in a good illumination of the scene. In many instances, it is sufficient to add a few minutes of radiosity processing to a final gather rendering to see a significant improvement in illumination.
Recall the scene shown in Figure 6.1.6.9 that started with a single distant light and was improved by adding the atmospheric light. With the atmospheric light still on, 2 minutes of radiosity processing is added to the scene prior to executing another final gather rendering. An intermediate rendering of the radiosity mesh is shown in Figure 6.1.6.13. Note, that the artifacts from the radiosity mesh are quite dramatic. However, they won’t affect the quality of the final rendering. What’s important is to see that the scene increases in brightness as the radiosity solution progresses. Figure 6.1.6.14 shows another low quality final gather rendering that uses the existing radiosity solution. Note, that the artifacts are gone and the scene is better illuminated than using just the distant and atmospheric lights. Figure 6.1.6.15 shows a higher quality final gather rendering. The **Quality** menu was set to 4, **Ambient Occlusion** was turned on again, and the additional **Ambient Light** was reduced to 25%.

**Figure 6.1.6.13:** A rendering of the radiosity solution.

**Figure 6.1.6.14:** The radiosity solution rendered with final gather.

**Figure 6.1.6.15:** A high quality final gather and ambient occlusion rendering of the radiosity solution.
Radiosity can be added to final gather in two ways. In the steps described above, the Display menu command Generate Radiosity Solution was selected. By default, the radiosity options are configured to operate as final gather support and the solution is set to run for 2 minutes. Every 20 seconds a quick preview rendering is produced, that gives the user the opportunity to evaluate the progress of the illumination. If the solution progresses quickly, it can be cancelled earlier. If it needs to run longer, more time can be added in the Radiosity Options dialog. Once the solution is complete, a RenderZone rendering is executed.

A second way to add radiosity is to select the Add Radiosity check box in the Final Gather tab in the RenderZone Options dialog. Below this check box, a few options are presented that allow the user to control the solution. The value in the Time field determines how long the solution is processed. The Time Spent So Far field below indicates the time that has been spent on an existing radiosity solution, if one already exists. Therefore, to add more time, the value entered in the Time field must be greater than the value shown in the Time Spent So Far field. If the Use Bounding Box option is selected, the bounding box set up by the Radiosity Bounding Box tool is used. This option is also available for the first method and is discussed in more detail later. If the Keep Solution option is selected, the radiosity solution calculated for the final gather rendering is saved after the rendering is finished. If no changes to objects or lights are made and another final gather rendering is selected, the radiosity solution is reused. If this option is off, a new radiosity solution needs to be calculated for each rendering. If the detailed parameters are shown in the Global Illumination tab, a Quality slider is displayed, that allows a custom quality setting for the radiosity solution. With the Add Radiosity option, the generation of the radiosity solution, the calculation of the final gather sample points and the rendering itself are all executed in a single step.

form•Z offers a few tools that can be used to make sure that radiosity works as efficiently as possible. With the Radiosity Bounding Box tool, one can limit the region in which radiosity is calculated. For example, if a rendering of a lobby of a high rise building is produced, there is no sense in calculating the radiosity solution of any of the rooms not visible in the rendering. The Radiosity Bounding Box tool allows the user to define a box tightly around the area, where radiosity is desired. Any surfaces outside this box are simply skipped. After sizing the bounding box with this tool, the Use Bounding Box option needs to be selected in the Radiosity Options dialog or in the Final Gather tab of the RenderZone Options dialog. A second condition that may slow down radiosity is when there are small objects in a scene that are made up of many faces. Because of their small size, such objects hardly reflect any significant amount of light, but, because of their high polygon count, slow down the radiosity process. With the Radiosity Attributes tool, such objects can be specifically excluded form a radiosity solution. For example, a room and furniture may consist of a few thousand polygons. A small statue located on a table in the room, however, may be modeled with tens of thousands of polygons. Excluding the statue from the radiosity solution will allow the processing time to be spent computing how the light bounces off the walls and furniture, which will have a much greater visual impact than the light reflected off the statue. These two tools are discussed in more detail in their own sections.
Specular effects with final gather

Diffuse illumination describes light that hits a surface and is reflected equally strong in all directions. A surface, which only has diffuse reflection appears matte and dull. Roughly sanded wood, or a flat wall paint are examples of materials with a high degree of diffuse reflection. Specular illumination describes light, that is reflected off a surface in about the same angle as the incident (or incoming) angle. On a curved surface, only those light rays, that are reflected off the surface and are hitting the eye point can be seen. Surfaces with specular reflection appear shiny, showing typical hot spots. A polished plastic or a glossy varnish paint are examples of materials with high specular reflection. Some materials have traits of both. For example smooth leather has some diffuse and some specular reflection.

Final gather presents three options in the Specular Illumination menu, which determine how specular effects are handled.

**Direct Without Dome Lights**: When this item is selected, specular effects are only created though direct illumination. This is the same as in a rendering not based on final gather. For example, a cone light shining on a glossy sphere will produce a bright hot spot. In addition, the specular effects of dome lights are not computed. For most architectural scenes, the highly non uniform light emitted by atmospheric or environment lights creates nice diffuse illumination, but little or no specular effects on flat surfaces. For example, a shiny floor won’t show any specular effects from dome lights. Thus, one might as well save the time computing them.

**Direct With Dome Lights**: When this item is selected, the specular effects are calculated as with the first option, except that dome lights now are included in the calculation of specular effect form direct illumination. This option is useful, if a scene consists of objects with curved surfaces, that produce specular highlights, such as the rendering of the dune buggy shown in Figure 6.1.6.12. Using this option will cause the rendering to be slower.

**Indirect From Final Gather**: With the first two options, a rendering shows only indirect diffuse illumination and direct specular illumination. When this option is selected, specular effects are calculated from all light sources though direct and indirect illumination. It is most useful to create subtle shine on glossy surfaces, such as furniture in an architectural rendering.

The reflection shader of each surface style, that has a Specular Reflection parameter also has a check box called *Calculate Specular Final Gather*, as shown in Figure 6.1.6.16. This check box also needs be turned on for the *Indirect From Final Gather* option to work. By default, the Brushed, Simple Metal, Plastic, and Woven Reflection shader have this option turned on, whereas all others have it turned off.
Calculating specular indirect illumination in final gather is a time consuming process. It is therefore recommended to only compute specular indirect illumination for those objects in a scene, where this effect is noticeable and visually desirable. Highly reflective or transparent materials, such as a mirror or window glass will not benefit from the effect. Nevertheless it would be calculated, unnecessarily slowing down a final gather rendering, if this option were selected in the respective reflection shader.

Therefore, the ray trace based reflection shaders, such as **Glass**, **Accurate Glass**, **Mirror** or **Paint** have the **Calculate Specular Final Gather** option turned off by default. A good use for this option would be to assign a surface style with the **Plastic** shader and the **Calculate Specular Final Gather** option enabled to a sofa object that shows a leather like material. Only when this object is rendered will final gather spend the extra time to compute the specular effect. For the glossy floor, the **Glossy** reflection shader is used. It has the **Calculate Specular Final Gather** option turned off. Since the specular effect is not necessary for the floor, the rendering won’t be slowed down. A final gather rendering without and with specular indirect illumination are shown in Figure 6.1.6.17 (a) and (b).

To compute specular indirect illumination, final gather sends a number of rays from the pixel into the scene. The rays will intersect other surfaces and final gather calculates the illumination at each intersection. From the total illumination at all intersections, the specular effect is generated. The rays cast from the pixel are not uniformly distributed, but are focused around the direction of the reflected view vector. When using the preset **Quality**, the number of rays is set automatically. When editing the final gather parameters manually, the **# Of Rays** field is shown below the **Specular Illumination** menu, as shown in Figure 6.1.6.18. A reasonable range for this parameter is 10 to 400. Note that very smooth and shiny surfaces require a higher number of rays than rough surfaces.
Quality settings

The Quality menu in the Global Illumination section determines how detailed the shadows appear that are generated by indirect illumination from final gather. At a low setting, final gather will execute fast, but the surfaces will show little contrast. The higher the setting the more the contrast will vary, revealing shading detail and generating subtle shadows around objects. For most renderings, a Quality setting of 3 is sufficient. Recall, that the Scene Size, Quality and Level Of Detail all affect the speed at which final gather executes. For a given Scene Size, raising the Quality will slow down rendering speed. But also if the Quality remains and the Level Of Detail is increased, rendering speed will also be affected. Therefore, if the final gather sampling process, that occurs at the beginning of a rendering takes too long, either the Quality is too high, or the Level Of Detail is too fine. Since the Level Of Detail is automatically computed from the Scene Size, it is also possible that a size was chosen that is too small compared to the real scene.

Detailed parameters

Ordinarily it is enough to control the execution of final gather with just the Scene Size, Quality, and Level Of Detail menus. These three setting determine the values of a larger set of detailed parameters that are not necessarily of interest to the average user. Only in rare occasions should the parameters be edited manually. By default, the detailed parameters are hidden. They can be displayed by hitting the Show Details button, as shown in Figure 6.1.6.18. Once at least one detailed parameter was changed manually, the Quality menu changes to Custom. The exact definition of each parameter and its impact on the final gather computation are described next. For each parameter a reasonable range is given, where the first value causes a final gather process to execute fast with lower quality results. The second value is an upper limit, which causes final gather to execute slower but with best rendering quality.

Figure 6.1.6.18: The Final Gather tab with details shown.
**Initial Sampling Distance**: Before a rendering, final gather places sample points in a scene at which the indirect illumination is calculated. The value in this field determines the closest possible distance in screen space of these sample points from each other, expressed as pixels. That is, two initial sample points cannot be closer than the value entered in this field. In reality, most sample points will be further apart and only get closer in areas, where light changes more rapidly across a surface. Reasonable values range from 10 to 2 pixels.

**Sampling Min Radius**: This parameter expresses the minimum radius to the next closest final gather sample point, in real world linear distance units. It means that two sample points cannot get any closer than this parameter. A very small value, relative to the scene size, potentially allows final gather to add many sample points. If there is a high degree of detail that needs to be rendered, this may be necessary. However, if too many samples are placed, final gather will slow down for no apparent gain in quality. A good value for this parameter is the size of the smallest object that one cares about to generate soft shadow. This parameters is actually set directly to the value indicated by the **Level Of Detail** menu. Reasonable values are scene size divided by 10 (coarse but fast) to scene size divided by 100 (fine but slow).

**Sampling Max Radius**: This parameter defines the maximum radius at which final gather sample points are spaced apart, in real world linear distance units. In other words, two sample points cannot get further apart than this parameter. It guarantees a minimum density distribution of sample points across a surface. If the parameter gets small and close to the **Sampling Min Radius** value, the total number of samples placed gets very high. This will result in a slow final gather computation, but will generate high levels of detail. If this parameter is set to large values, such as the size of the scene, sample point density is low and final gather will execute faster, but reveal little detail. Reasonable values are scene size times 2 (coarse but fast) to scene size divided by 25 (fine but slow).

**Max Ray Length**: In order to determine which other surfaces in the scene reflect light onto a sample point, final gather casts a number of rays from the point. At the intersection of a ray with another surface the illumination is computed, or gathered from other sample points already close to that intersection point. The average of all those values calculated at the ray intersections is stored with the sample point as its indirect illumination value. The **Max Ray Length** parameter indicates the maximum distance a ray can travel to check for intersections with other surfaces. This parameter should always be set to the size of the scene. If it is set to a value that is significantly smaller than the scene, the sample point distribution becomes coarser, where many samples are closer to **Sampling Max Radius** apart. If the parameter becomes larger than the scene, sample points are spaced denser, up to the **Sampling Min Radius** distance apart.
# Of Rays: This parameter determines how many rays are cast from a sample point into the scene to test for intersection with other surfaces and to compute the intensity reflected off these surfaces. The goal of these rays is, to find bright spots on other surfaces resulting from direct illumination. Only if these rays actually find lit surfaces can final gather compute indirect illumination. Therefore this parameter should be set to a reasonable high number, especially if the lit area in a scene is small. This would be the case of a small window opening in a wall, allowing just a small patch of sunlight to illuminate the floor. The rays cast from a sample point on the ceiling above will mostly hit dark areas, except for a few, if any, that happen to hit the small sunlight patch on the floor. As described earlier, final gather works best if enough initial light is supplied in a scene. In the case of the sun light patch, final gather will be much more successful, if the atmospheric light of the sun is also used and if a short radiosity solution is calculated. Both of those methods will add more light, causing many more rays to hit surface that have at least some light. A good range for the # Of Rays parameter is 500 to 1000.

Interpolation Quality: This parameter determines how much effort final gather spends at rendering time to gather indirect illumination for a each pixel. Once the initial phase of calculating the sample points is finished, the actual rendering of the scene can proceed. To get the direct illumination of a pixel, it is shaded as in a regular rendering. To get the indirect illumination, those sample points that are close to the pixel are considered. Their pre-computed indirect illumination is averaged through an interpolation process and used for the pixel. The Interpolation Quality parameter determines how precise the interpolation process is executed. A low value, such as 20%, causes the rendering to proceed faster, but leaves noticeable patches of light and dark regions. For values above 50% the patches disappear, but the rendering will take a little longer. In a practical application, even a value of 50% renders reasonably fast and can be considered the lower value of a good range. 90% Interpolation Quality is a good upper range setting.
**Best Dome Light Quality**: As described earlier, the environment and atmospheric light types, also referred to as dome lights, are optimized for final gather. That is, their # Of Samples setting has no influence on the quality of a final gather rendering and their lighting contribution always creates smooth shadows. The **Best Dome Light Quality** parameter further determines how much effort should be spent to compute illumination from these lights. If the parameter is off, dome lights only contribute light to a sample point for those rays that leave the scene without hitting any surface. That is, they end up directly in the sky, where they collect the appropriate light intensity for the sample point. If the **Best Dome Light Quality** option is on, the intersection point of those rays that hit other surfaces are also illuminated by the dome light. This type of illumination is executed by casting light from each sample point of the dome light onto the intersection point. In this case, the # Of Samples parameter of some light will matter. However, a low # Of Samples is still a valid option, since final gather will smooth any rough shadows through its interpolation process. Therefore, if the # Of Samples parameter of any dome light in the scene is set to values larger than 100, the setting is reduced to 100. Using the **Best Dome Light Quality** option will make the scene brighter and add better illumination quality. However, the rendering will become significantly slower. If using the automated **Quality**, settings of 3 and lower turn this option off, whereas a **Quality** of 4 and higher turns it on. A scene rendered with the **Best Dome Light Quality** off and on is shown in Figure 6.1.6.19 (a) and (b).

![Figure 6.1.6.19: An interior scene rendered (a) without and (b) with the Best Dome Light Quality option.](image)
Global Illumination Tutorial

This tutorial illustrates the steps necessary to improve a simple interior rendering by applying a number of global illumination techniques.

• **Default rendering**
  - Load the sample file GI_tutorial.fmz. In this first step, we produce a base line rendering.
  - Click RenderZone* in the Display menu.
  This will render the scene with the default RenderZone options, without any global illumination features. The room is illuminated by a single light source, a distant light, which represents the sun. You may want to save this rendered image, as shown in Figure 6.1.6.20, for comparison with the final image, which includes full global illumination effects.

• **Adding final gather**
  - Invoke the RenderZone Options dialog.
  - Select the Global Illumination tab.
  - In this tab, choose the Final Gather check box.
  - Set the Scene Size menu to From View. All other settings may be left at their default values (Figure 6.1.6.21).
  - Select OK to exit the dialog and start a rendering. The image produced will show improved distribution of the reflected light off the floor, as shown in Figure 6.1.6.22.
Adding atmospheric light

While the rendering in Figure 6.1.6.22 shows some lighting improvements, more needs to be done to get realistic illumination. The distant light source used in the scene simulates the sun by sending parallel light rays into the room. In reality a significant amount of light also comes from the sky, clouds, and the outside environment. The distant light alone does not produce this kind of illumination.

- Double click on the light named “sun” in the Lights palette to open the Light Parameters dialog.

- Click on the Parameters tab and select the Apply Atmospheric Light option (Figure 6.1.6.23). Increase the Intensity of the atmospheric light to 600%.

- Select OK to exit the dialog.

- Select RenderZone from the Display menu to start a new rendering. Observe that the room became noticeably brighter, especially on the floor and walls (Figure 6.1.6.24).

Figure 6.1.6.23: The Apply Atmospheric Light option in the Parameters tab of the Light Parameters dialog.

Figure 6.1.6.24: The room becomes noticeably brighter.
• **Applying radiosity**

So far, final gather has distributed direct light from the sun and atmospheric light from the sky by bouncing it once off surfaces into the room. The image can be improved even further by adding a small amount of radiosity to it. Before starting a radiosity solution, it is advisable to perform a few steps to make sure radiosity executes efficiently.

• **Using the Radiosity Bounding Box tool**

  • Change the rendering back to **Wireframe** and open a new modeling window.

  • Set the view to **Top** and zoom fit the building.

  • Next, select the Radiosity Bounding Box tool from the Attributes tool palette and click in the new window. This will show a bounding box with arrow handles as shown in Figure 6.1.6.25.

  • Working in top and side views, resize the bounding box to include only the top right portion of the building, which is where the view is in the original modeling window. The final state of the bounding box should be as shown in the **Interactive Shaded** rendering of Figure 6.1.6.26.

  • Close the new modeling window to return to the original window.

In these steps, we reduced the area in which radiosity occurs. Since the view only shows a portion of the whole building, there is no need to compute radiosity in the other parts. This will speed up radiosity.

*Figure 6.1.6.25:* Bounding box with arrow handles.  
*Figure 6.1.6.26:* An **Interactive Shaded** rendering.
• **Setting object radiosity attributes**

To further speed up radiosity, we will exclude the furniture from any radiosity calculations. Since the furniture pieces are rather small compared to the size of the scene, they will not contribute any noticeable amount of reflected light. But since they are composed of many faces, radiosity would spend much time processing the surfaces of those objects.

- Prepick the chaise, couch, and chair.
- Select the Radiosity Attributes tool from the Attributes tool palette.
- In the Tool Option palette, set the **Object Radiosity** menu option to **No Receive And No Bounce** (Figure 6.1.6.27).
- Click anywhere in the project window to assign the attribute.

• **Executing a radiosity solution**

To distribute more light in the scene we will now execute a quick radiosity solution.

- For rendering mode select **Shaded Render** or **Interactive Shaded**.
- Select **Radiosity Options** from the **Display** menu.
- In the **Radiosity Options** dialog (Figure 6.1.6.28), increase the **Quality Slider** in the **Options** tab to 75%. This will make the solution a bit better than the 50% default. We can afford the increased quality since we took extra steps to speed up the solution.

![Figure 6.1.6.27: The Object Radiosity menu option set to No Receive And No Bounce.](image)

![Figure 6.1.6.28: The Quality Slider in the Options tab of the Radiosity Options dialog.](image)
• Select the **Use Bounding Box** option. This will activate the box defined earlier.

In the **Termination** tab, the solution is set to run for 2 minutes. This is most likely too much, but you can keep it unchanged.

In the **Preview** tab, the solution is set to be rendered every 20 seconds.

• Close the **Radiosity Options** dialog by clicking **OK**.

• Now select **Generate Radiosity Solution** from the **Display** menu.

After 20 seconds the first preview will be drawn (Figure 6.1.6.29). You can see that the scene has brightened up significantly since the first rendering. Radiosity already has distributed light through many bounces in the scene. You may also notice that the shadows created by radiosity are not very accurate. This is not of much concern right now, as **Final Gather** will later on produce proper crisp shadows.

You may let the solution run a while longer, but after the second or third preview drawing, no visible improvements will occur anymore. You may stop the solution by hitting **Cancel** in the progress bar.

• Now execute another **RenderZone** rendering. Final gather will now incorporate the radiosity lighting into its own illumination calculations. The result is shown in Figure 6.1.6.30.
• **Adding ambient occlusion**

In a final step we will add some ambient occlusion to the scene. Unlike final gather and radiosity, which distribute light in a forward direction through bounces, ambient occlusion looks backwards from a point to be shaded and determines if there is any light that can not reach the point. If so, the point is rendered a bit darker. This method works well for generating subtle drop shadows under and behind furniture and adds a little contrast in the corners of the room.

• Invoke the **RenderZone Options** dialog.
• Select the **Ambient Occlusion** option in the **Global Illumination** tab.
• Increase the **Additional Contrast** slider to 50%.

Regular contrast is generated by ambient occlusion by creating shades of darkness between 0% ambient light and the amount of ambient light shown in the **Ambient Occlusion** tab. Since the ambient light setting is only 10%, additional contrast needs to be created to darken the extra light created by radiosity and final gather.

• Click **OK** to generate the rendering shown in Figure 6.1.6.31.

*Figure 6.1.6.31: Adding Ambient Occlusion.*
• The final touch : Exposure correction

It is quite possible, that after illumination has been calculated by radiosity and final gather the scene has become too bright. Considering the darkness in the first rendering, one may easily increase the light intensities too much, or add too many lights to make the image brighter. Instead of repeating all the steps above after making adjustments to the light intensities of the distant and atmospheric lights, it is much easier to correct the image exposure through the RenderZone Postprocess command. While this feature is discussed in detail later (section 6.1.10), to complete our tutorial example, will also cover the basics here.

• Select RenderZone Postprocess from the Display menu.

• In the RenderZone Postprocess Options dialog select the Exposure Correction check box and click on the Options button next to it. This invokes the Exposure Correction Option dialog shown in Figure 6.1.6.32.

• From the Type menu select Automatic and adjust the Bias and Brightness sliders until a satisfactory image is shown in the preview area. A Bias of 60% and a Brightness of 15% were used for the image shown in Figure 6.1.6.33.

Figure 6.1.6.32: The first preview drawn after 20 seconds.

Figure 6.1.6.33: A Bias of 60% and a Brightness of 15% were used for the image shown.
6.1.7 The Geometry tab

The options in this tab of the RenderZone Options dialog determine how non planar faces will be decomposed, how objects that appear as wire frames in a shaded rendering will be plotted, and how fine smooth objects are rendered. The Geometry tab is shown in Figure 6.1.7.1.

Decomposing non planar faces

Decompose Non Planar Faces:
When this option is selected, all non planar faces contained in the modeling scene will be decomposed into triangles. If not selected, all surfaces will be rendered as they are.

form•Z RenderZone Plus will render non planar surfaces regardless of whether or not they are triangulated, and it will not triangulate them automatically. However, non planar surfaces are not guaranteed to always produce correct renderings because the math calculations are not accurate when non planar surfaces are involved. This is particularly true when non planar surfaces intersect, or when textures are mapped onto them. It is therefore strongly recommended that the Decompose Non Planar Faces option in the RenderZone Options dialog be turned on when a scene is known to contain non planar surfaces. On the other hand, if all the surfaces in a modeling scene are planar, turning this option off typically results in some speed gains, since the program does not have to check the planarity of all the faces in the scene.

Figure 6.1.7.1: The Geometry tab of the RenderZone Options dialog.
Rendering wire frame objects

Two options in a box labeled **Wireframe Width** allow you to globally control wire frame widths when mixing wire frames with shaded renderings.

**Scale By:** When this option is selected, the line widths of the objects that will be rendered as wire frames are scaled by the amount indicated in the text field. This is useful when, for example, in a final rendering the image resolution is increased from 72 to 300 dpi in the **Image Options** dialog. In this case, scaling the line width by a factor of $300/72 = 4.16$ will preserve the line width relative to the image size.

**Set To n Pixels:** When this option is selected, all the objects rendered as wire frames are displayed with the line width entered in this text field. In other words, this option globally overrides the line width attribute assigned to individual objects with the Render Attributes tool.

The two options are illustrated in Figure 6.1.7.2.

---

**Figure 6.1.7.2:** Two renderings of a blimp using the two **Wireframe Width** options:
(a) the **Scale By** option is selected with a factor of 1 entered in its box and (b) the **Set To** option is selected with a factor of 3 pixels entered in its box.
Rendering quality of smooth objects

The **Geometry** tab contains two slider bars intended specifically for controlling the rendering of smooth objects, as follows:

**Silhouette Quality**: This slider bar controls the quality of shading of smooth objects along their silhouette edges. At rendering time, **form•Z** will generate a facetted representation of smooth objects. If this slider is set to its highest value, the facetting is done so densely, that no facetting artifacts can be observed along the silhouette edges of curved surfaces. If the slider is moved to the left, less facets will be created, which results in memory savings and an increased rendering speed, but will also produce visible facetting artifacts.

**Surface Quality**: This slider bar controls the quality of shading of smooth objects on their surfaces. This setting controls the facetting of curved surfaces on their insides independently from the **Silhouette Quality** option. In general, the **Surface Quality** slider can be set to a lower value than the **Silhouette Quality** and still result in high quality renderings. Only for highly reflective surfaces that mirror geometric patterns, such as stripes, is it necessary to also increase the **Surface Quality** setting to a higher value.

Examples of a smooth object rendered with different quality settings are shown in Figure 6.1.7.3.

*Figure 6.1.7.3:* A smooth object with (a) low **Silhouette Quality**, (b) high **Silhouette Quality**, (c) low **Surface Quality**, and (d) high **Surface Quality**.
6.1.8 The Scene tab

The Scene tab of the RenderZone Options dialog, shown in Figure 6.1.8.1, contains three groups of options, which allow you to set the world within which a modeling scene may be rendered. They are Background, Environment, and Depth Effect. The former two are discussed in this subsection. The latter in the next.

![Figure 6.1.8.1: The Scene tab of the RenderZone Options dialog.](image)
Background

A background pattern or image can be generated by selecting one of the items in the Background pop up menu, shown in Figure 6.1.8.2. Note that there is no check box in front of this menu, therefore a background is always generated. However, a plain white or black color may be thought of as being equivalent to no background. Also note that, unless a background other than the default, Project Color, is selected from the Background menu, the background color set in the Project Colors dialog (invoked from the Options menu) is used.

Background colors or textures, which can be either procedural or precaptured, are generated by shaders, the same way shaders are used to define surface styles that are mapped onto objects. The background can be thought of as a surface style that consists of a single shader, as opposed to the four shaders that are overlaid to produce the surface styles that are assigned to objects.

The menu is organized in 5 groups:

- The first group contains shaders that generate a plain color or use the project background.

- The second group contains procedural shaders. Two of these, Graduated and Sky, are unique to the Background menu and are discussed in this subsection. The others are similar to the color shaders of the surface styles and are discussed in detail in subsection 6.2.3. Their only difference is that the Background shaders do not have the Area Sampling parameter.

- The third, fourth, and fifth groups contain one shader each, each offering a unique feature. They are also discussed in this subsection.

All the shaders, including those used for the background, are controlled by parameters selected in their respective dialogs. The dialog for the item currently selected in the Background menu is invoked by clicking on the Options... button which is next to the menu. All shader dialogs follow a similar format, and include a preview area on their right portion. The color or image resulting from the current parameters in the dialog is displayed in the preview area, and is regenerated each time a parameter is changed. All the shader dialogs use similar types of parameters, such as color boxes, sliding bars, and editable size fields. These types are discussed in more detail in section 6.2.2.
**Plain Color**: When this item is selected, the background is set to a uniform color. This color can be changed by clicking in the **Color** box next to the menu, which invokes the Color Picker dialog.

**Project Color**: This item is the default, and resembles the **Plain Color** item in that it also sets the background to a plain and uniform color. However, this color is the same as that used as background for all the other rendering methods, as selected from the **Project Colors** dialog invoked from the **Options** menu. When the **Project Color** item is selected and another color is picked from the Color Picker dialog, invoked by clicking in the **Color** box next to it, this color will also be used as background for all the other rendering methods.

**Project Background**: This item causes the background to be rendered as it is in any of the other rendering modes, including grid, axes, and underlay. Its one option can be selected from the **Project Background Options** dialog it invokes (Figure 6.1.8.3).

**Render Grid And Axes As 3D Lines**: When this option is on, the grid and axes are rendered as 3D entities. As a result, if the grid is at a location where it passes through the middle of an object, the rendering will show the grid lines intersecting properly with the object, as illustrated in Figure 6.1.8.4(a). If this option is off, the grid and axes are generated as a 2D background image. That is, a 3D object will always cover the grid and axes, even if they pass through the object. This is illustrated in Figure 6.1.8.4(b).

**Graduated**: When this item is selected, the background is a mix of two colors, which are set in the **Graduated Options** dialog shown in Figure 6.1.8.5. These colors can be changed by clicking on their boxes to invoke the Color Picker dialog. The background starts with the **Top Color** at the top of the image and ends with the **Bottom Color** at the bottom of the image. The two colors gradually phase into each other between the two ends of the image. The preview at the right of the dialog shows how the background will be rendered using the current selection of colors.
**Sky:** When this item is selected, a background with clouds, sun, sky, and horizon is rendered. This item invokes the **Sky Options** dialog, shown in Figure 6.1.8.6. It contains five boxes for setting respective parameters.

**Sky, Earth:** These groups of parameters determine the color mixture for the sky and earth portions of the background image.

**Color: Top, Bottom:** The colors displayed in these boxes determine a color gradation that will be rendered for the respective area of the background.

**Falloff:** These slider bars control how fast the **Top** color changes to the **Bottom** color.

**Clouds:** When this checkbox is selected, clouds are rendered in the background image.

**Horizontal / Vertical Scale:** These parameters determine the size of the clouds. The horizontal cloud direction corresponds to the X axis of the reference plane, or the horizontal screen direction, if the **Match Horizon With Perspective** option is not selected. The vertical cloud direction corresponds to the Y axis or vertical screen direction, respectively.

**Color: Top, Bottom; Falloff:** As for **Sky** and **Earth**.

**Detail:** This slider controls the sharpness of the clouds. Higher values produce fuzzier clouds.

**Coverage:** The cloud coverage of the sky can be set through this slider bar. 100% coverage will generate an overcast sky, while 0% will show no clouds at all.
Density: This slider bar sets the density. Clouds with 100% density are totally opaque. 50% density shows 50% of the sky color through the clouds. 0% density generates invisible clouds.

At Horizon Fall Off Towards: This pop up menu contains two items. If Sky is selected, as clouds approach the horizon line, they tend to disappear into the sky color. This simulates atmospheric conditions where the visibility is very high and the sky can be seen at the horizon. If Clouds is selected, the area above the horizon falls off towards the bottom cloud color. This simulates foggy atmospheric conditions.

Amount Of Sky To Mix With: This parameter determines how much of the area above the horizon is mixed with either the sky or cloud color.

Sun: When this checkbox is selected, a sun is rendered in the background image.

Diameter n % Of Image Width: This parameter determines the size of the sun relative to the overall image size.

Color: The color displayed in the box is the color used for the sun. It can be changed as usual.

Fuzz: This parameter determines how fuzzy the edge of the sun appears. 0% generates a perfectly crisp edge. 100% generates a fuzzy area equal to the radius of the sun. It falls off towards the sky color.

Horizontal n % Along Horizon, Vertical n % Above Horizon: These parameters control the position of the sun horizontally and vertically. Horizontal 50 % places the sun in the middle of the image, horizontally. Vertical 50% places it half way between the horizon and the top of the image.

Match Horizon With Perspective: When this option is selected, the horizon line of the background image is automatically placed at the horizon implied by the current perspective view. If this option is off, the horizon is a horizontal line at the height indicated by the Horizon slider bar.

Reference Plane: This pop up menu contains four items: XY (default), YZ, ZX, and Active Plane. The item selected from this menu determines which plane is considered the ground plane. The horizon line will always be parallel to this plane.

Clouds Are n Above Plane: This value determines how high the clouds are from the ground.

Ground Is n Below Plane: The value entered in this field determines how low the ground is below the Reference Plane.

Horizon: This slider bar determines how far above the bottom of the image the horizon line is located, when the current view is not a perspective view, or Match Horizon With Perspective is off.
Background Map: This item generates a background from an image file. Clicking on the **Options...** button invokes the **Background Map Options** dialog (Figure 6.1.8.7), which contains the typical options of the image mapping dialogs and, in addition, three options that determine how the background image will be sized.

**Fit Image to Window**: When this option is selected, the background image is sized to fit the size of the window where the image will be rendered. Note that with this option, if the image is significantly smaller than the window, the background may show a significant decrease in resolution.

**Keep Image Size**: When this option is selected, the background image will be rendered at its original size. If the image is smaller than the rendering window, it is tiled automatically. If it is larger, it is cropped. In both cases, the image is centered in the rendering window.

**Scale Image**: When this option is selected, a scale for the image, relative to the rendering window, can be entered in the text field next to it. For example, if 25% is entered, the image is scaled to 1/4 of the size of the window and is repeated 4 times.

**Maintain Proportions**: This option is only available when **Scale Image** is selected. When checked, the original proportions of the scaled image are maintained.
Environment: This item can be selected from the Background menu when the Environment option is selected. Otherwise this item is dimmed and unavailable. This background shader will render the pattern of the current environment sphere or cube as the background. If this shader is not selected, the environment sphere or cube is visible only in reflective surfaces and the background pattern is rendered according to the current background shader. The environment background is shown as a portion of a sphere or cube. How much of the sphere or cube is visible, depends on the view angle of the current perspective as illustrated in Figure 6.1.8.8. If the view is not a perspective, the entire environment is shown in the background. Environments are discussed in more detail in the next section.

Figure 6.1.8.8: How the environment is rendered as the background.

The Environment background shader is especially useful when creating an animation with a perspective view. Since the portion of the background rendered depends on the current view angle and direction, an animation will render each frame with a slightly different background. This gives the impression that the view pans around the environment. If a regular background is used instead, each rendered frame will have the same background. Although the view changes during the animation, the background will appear static.

Rendering a background from a portion of the environment sphere will introduce a certain amount of distortion in the background. This distortion is the smallest around the equator of the sphere and most dramatic at the poles. When using the Image environment shader, the image will by default be wrapped around the sphere from pole to pole and once around the equator. This is illustrated in Figure 6.1.8.20. If an animation is known not to look at the pole areas of the environment sphere, there is no need to cover that area with the image, leaving the image to be projected on only the visible area of the sphere. Which portion of the environment sphere is covered by the environment image can be determined in the Spherical Environment Map Limits dialog (Figure 6.1.8.19), invoked from the Spherical Environment Map Options dialog, as discussed later in this subsection.
**Alpha Channel:** When this item is selected, the background is rendered as a solid black color, but a transparency value is also stored in a fourth channel called the *alpha channel*. This fourth channel can be used to smoothly blend the edges of images when they are pasted on top of other images.

An alpha value is computed for each pixel. This value represents how transparent a pixel is relative to the background. This transparency may result from one of two conditions. First, pixels which are located on the edges of polygons may be partially transparent because they were antialiased when rendered in the z-buffer or raytrace rendering modes. Second, a pixel may be transparent because it is located on a polygon that is rendered with a surface style, which uses any of the transparency shaders, except *None*. A transparency shader can be selected from the *Transparency* pop up menu in the *Surface Style Parameters* dialog, which is invoked by double clicking on a surface style icon in the Surface Styles palette.

When such an image is saved in an image format that supports alpha channel, the alpha values for each pixel are stored in addition to the three channels for the red, green, and blue color components in a fourth channel. This is illustrated in Figure 6.1.8.9, where the RGB colors are displayed in (a) and the alpha channel in (b). Note that the RGB image displays the glow of the lamp opaque, or in its full intensity. This is necessary for producing appropriate results when the alpha channel is applied as a filter. Filtering the RGB image through the alpha channel image produces the result in Figure 6.1.8.9(c).

![Image](image1.png)

**Figure 6.1.8.9:** Rendering with *Alpha Channel*: (a) the RGB channels, (b) the alpha channel, and (c) the RGB image filtered through the alpha channel.

When rendering an image in *form-Z RenderZone Plus* with *Alpha Channel* used as background, you can display it with or without the alpha channel filter applied to it. This is controlled by the *Apply Alpha Channel* option in the postprocess tab of the *RenderZone Options* dialog. This option is also available in the *RenderZone Postprocess* dialog and is discussed in more detail in section 6.1.10. When off, the alpha channel filter is not applied and the image is displayed as in Figure 6.1.8.9(a). When on, the filter is applied and the image is displayed as in Figure 6.1.8.9(c). While turning this option on is useful for viewing images, it should be off when saving an image to a 2D image file.
When editing an image rendered with **Alpha Channel** in an image processing software like Adobe Photoshop™, the alpha channel can be used to correctly superimpose the color channels of the image over a new background image. When this is done, the alpha values determine how much the color of the image is mixed with the new background. For example, if a pixel carries an alpha value of 30%, 70% of the background color and 30% of the image color at that pixel are mixed together. The image appears as if it were rendered with the new background, including antialiased edges and transparent surfaces. This is illustrated in Figure 6.1.8.10. The room shown in (a) is rendered separately and then the lamp from Figure 6.1.8.9 is pasted on it. The alpha channel produces a smooth blend of the two images as if they were rendered together. One of the advantages of this method is that a number of images can be generated, which show the same scene on different backgrounds, without having to render each image individually.

As further discussed in sections 6.2.4 and 6.2.5, transparency effects can be achieved in two different ways in **form•Z RenderZone Plus**. Objects rendered with surface styles which include the **Glass, Accurate, Glass, Simple**, or **Generic** reflection shaders can accurately simulate the change in view direction associated with transparent materials, which is caused by refraction. Transparent surfaces can also be generated by selecting any of the transparency shaders. This kind of transparency acts as a compositing mechanism, where the color of a transparent surface is superimposed over the existing background, without handling refraction. Only transparencies from transparency shaders generate alpha values other than 100%. Transparencies from the **Glass, Accurate, Glass, Simple**, or **Generic** reflection shaders generate an opaque (100%) alpha value.
Setting the environment

Objects in a scene can be rendered with reflections, which originate from other objects in the scene or from an environment. This environment is assumed to be surrounding the scene, but is not rendered as part of the scene, unless the **Environment** shader is selected in the **Background** pop up menu as discussed earlier. Recall that reflections are rendered when the **Reflections** option is selected from the **Shading** tab.

In the **Environment** group of options, two distinctly different environments can be defined: **mapped** and **rendered**. A mapped environment uses one or six textures, depending on whether the mapping type is spherical or cubic. The pattern of these textures is shown in reflective surfaces. The rendered environment is used in combination with the **Environment** reflection shader of a surface style. Instead of using precaptured textures, the reflective pattern consists of six rendered images of the scene, one for each of the orthogonal directions. This is described in more detail later in this section.

**Environment**: When this option is off, faces to which a reflective shader has been assigned produce reflections from other objects but not from an environment. This is illustrated in Figure 6.1.8.11. The scene consists of a teapot in a room. Note that the room and its walls are the objects in the scene. The **Mirror** reflection shader (see section 6.2.4) has been assigned to the pot and **Environment** is off. The image shown has resulted from “pure” raytraced reflections causing the room to be reflected on the surface of the pot. The reflections are accurate but require additional time to calculate. Later in this section the same scene is shown rendered using different types of environment reflections. When **Environment** is on, the two environments, mapped and rendered, can be set up.

The group labeled **Type** contains the parameters for the mapped environment. It contains two pop up menus and a button. One menu controls the mapping type and the other the selection of a shader.

**Shader**: This pop up menu, shown in Figure 6.1.8.12, allows you to select the particular environment shader. This can be of a plain color, procedural, or image based. The dialog containing the settings of the currently selected shader can be invoked by clicking on the **Options...** button next to the menu. Note that this pop up menu is a subset of the **Background** menu and the parameters of its shaders are set as for the background.
**Type:** This pop up menu, shown in Figure 6.1.8.13, allows you to specify how the environment shader will be mapped. It has two items, as follows:

**Cubic:** When this item is selected, the shader picked for the environment is mapped onto an imaginary cube from which it is reflected on the surfaces to which reflective shaders have been assigned. If a procedural shader is selected, the respective pattern will be automatically placed on each of the surfaces of the environment cube. If **Environment Map** is selected, then either a single map or six individual maps are selected to define the image for each side of an imaginary cube surrounding the scene. When choosing six individual maps, a different image map can be used for each side of the cube, or the same map can be repeated for all sides. When choosing a single map, the image map needs to contain all the images for the six sides of the cube in a specific layout, which resembles a cross. Therefore these maps are also referred to as **cubic cross maps.** An example of such a map is shown in Figure 6.1.8.14. Note that, unlike the horizontal layout of the six individual images of the environment cube, the cubic cross map presents the sub images in a vertical format. Defining the images and choosing between the two different map types can be done in the **Environment Maps** dialog (shown in Figure 6.1.8.15), which is invoked by clicking on the **Options...** button next to the **Environment Map** item that is displayed in the **Shader** pop up menu.

The **Environment Maps** dialog has two options at the top.

**Six Individual Maps:** When this option is selected, the dialog displays six squares, each corresponding to a side of the unfolded imaginary cube that surrounds the scene. Initially, the system default texture map is assigned to each side of the cube (Figure 6.1.8.15(a)).

A new image file can be loaded in each of the squares, as shown in Figure 6.1.8.15(b). To work properly, these images should be correlated to present a panoramic view of an environment extending around the scene to be rendered. When using a z-buffer or raytrace rendering type, all reflective surfaces will show the portion of the environment toward which they are oriented. Usually, the environment cube itself is not rendered, but only appears in the reflections of the surfaces. However, if the **Environment** background shader is selected from the **Background** menu, the portion of the environment which is visible in the current view is rendered as the background. This is discussed in more detail earlier in this subsection.
Images are loaded into the squares of the environment cube by clicking inside each square. This invokes the **Cubic Environment Map Options** dialog, shown in Figure 6.1.8.16. The general options for this type of dialog were discussed earlier. This particular dialog has an additional parameter and an additional button:

**Scale**: The value entered in this field determines the size of the image as a percentage of the face of the environment cube. At 100% the whole image will fit exactly from one end of the face to the other. At 25% the image will be repeated four times. Percentages greater than 100% result in partial images being mapped onto the faces of the environment cube.

**Apply To All**: Closing the **Cubic Environment Map Options** dialog by pressing the **OK** button places the image in the currently edited square of the environment cube, and each of the squares needs to be assigned individually to load other images onto them. Clicking on this button will apply the currently loaded image and all of its parameters to all six sides of the environment cube.
**Single Cross Map**: When this option is selected, the Environment Maps dialog shows a single cross map (Figure 6.1.8.17). Initially, this is also a system default texture map, which looks different from the one used for the six individual images. It is properly formatted as a cubic cross map. A new map can be selected in the usual fashion. Note, it is important that the map selected follows the layout convention for the cubic cross map. That is, the six sub images must be ordered in the vertical format, with the top image at the middle top of the map and the remaining sub images unfolding below. Therefore, the width to height pixel ratio of such a map must always be 3 : 4. For example, it may be 768 to 1024 pixels. If an image map with a ratio other than 3 : 4 is selected, an error will be posted when the dialog is closed. Many of the cubic environments distributed as maps are now available as cross maps. While there is some waste of space in the texture map itself, it is more convenient to define the environment cube through a single image map rather than selecting and properly placing six maps. An example of a cubic cross map is shown in Figure 6.1.8.14.
**Spherical**: When this item is selected, faces to which a reflective surface style has been assigned will reflect the pattern generated by an environment shader mapped onto an imaginary sphere that surrounds the modeling scene. That is, this environment is similar to **Cubic**, except that it applies a spherical rather than a cubic reflection. With this environment, the pattern is wrapped around an imaginary sphere in a way such that the horizontal direction of the pattern runs parallel to the equator of the sphere. The axis of the sphere is always parallel to the world Z axis.

A particular environment shader can be selected from the **Environment** type pop up menu, as for **Cubic**. When the **Environment Map** item is selected, a precaptured image can be mapped onto the environment sphere through the **Spherical Environment Map Options** dialog shown in Figure 6.1.8.18. It is invoked by clicking on the **Options...** button next to the **Environment Map** item shown in the menu.

The content of the **Spherical Environment Map Options** dialog is the same as that of the **Cubic Environment Map Options** dialog, except that the **Apply To All** option is not available (since only one image is selected and mapped onto the environment sphere), and it has an additional button and menu.

**Set Limits...**: Clicking on this button invokes the **Spherical Environment Map Limits** dialog shown in Figure 6.1.8.19. It contains options for determining which portion of the environment sphere is covered by the environment map. The default values in this dialog result in a completely curved environment sphere, as shown in Figure 6.1.8.20.

**Horizontal**: The options in this box control how the environment image is positioned horizontally.

**Start/End**: These angles indicate where the horizontal extents of the image start and end. Defaults are 0 and 360 degrees respectively.

**Rotation**: This value indicates the rotation of the environment sphere around the world Z axis. A value of 0 degrees, which is the default, places the start angle of 0 at the world X axis.
**Vertical:** The options in this box control how the environment map is placed vertically.

**Top/Bottom:** These angles indicate where the vertical extents of the image start and end.

Alternatively to typing the angles in the text edit fields, the red circles on the left of the dialog allow for graphic editing of the limits. Clicking in one of the handles the angle can be dragged graphically to its desired position. Holding the **shift** key down while dragging will adjust the horizontal start and end angle or the vertical top and bottom angle symmetrically. Holding the **option** (Macintosh) or **ctrl+shift** (Windows) key allows you to adjust one of the handles at a time without affecting the others.

If a portion of the environment sphere is rendered which is not covered by the image, the colors at the edge of the image are stretched to continue beyond the image limits.

**Figure 6.1.8.20:** (a) A panoramic environment image, (b) wrapped around the environment sphere.

**Map Layout:** One of two available spherical map functions is selected from this pop up menu:

**Panoramic:** When this item is selected, the environment map represents an unfolded and flattened sphere, not unlike a map of the world. Ideally, the proportions of such a map should be 2:1, with the width dimension representing the distance along the equator of the environment sphere and the height representing the distance between the two poles. Once mapped, the area of the image which is along the horizontal middle section will appear least distorted, whereas the area at the top and bottom of the image will be distorted most, since this area will be compressed at the poles. An example of a panoramic environment map is shown in Figure 6.1.8.20.

**Spherical:** When this item is selected, the environment map should show an image that resembles a photograph of a mirror ball that is placed in the middle of a scene. The image reflected in the ball should show the environment to be used in the rendering. An example of such an image is shown in Figure 6.1.8.21. Environment maps are now commonly distributed in such a format. Note that the area in the map outside of the mirror ball may be black and is not used in the rendering. The width to height ratio of the angular map must be 1:1. If a map of a different proportion is chosen, an error will be posted when the dialog is closed.
Spherical and cubic environment reflections generate slightly different patterns of reflective surfaces. Which one is preferable depends on the particular reflective effect desired and the type of the environment. For example, to generate the interior reflections of a square room one could produce a single image which contains a panoramic view of the entire space. This image can then be mapped onto an environment sphere. With such a mapping, the reflections of the areas that are close to the poles of the sphere will display significant distortions. An alternative and possibly preferable approach in this case would be to capture six images of the room, one for each of the four walls, one looking straight up at the ceiling, and one looking straight down at the floor. The resulting images would then be placed on the respective sides of an environment cube. Reflections from such a cubic environment will show less distortions. On the other hand, if an open environment such as a landscape or sky needs to be reflected, then a spherical environment may produce more appropriate results. Regarding memory requirements, spherical environments are more efficient than cubic environments since the latter requires six images to be stored, versus the one stored for spherical environments.

Figure 6.1.8.21: An Spherical environment map, courtesy of Sachform Technology GbR (www.sachform.de) from the LightWorks HDRI Starter Collection (www.lightworkdesign.com).
Examples of environment reflection using cubic and spherical environments are shown in Figures 6.1.8.22, and 6.1.8.23, respectively. Note that this is the same scene that earlier (Figure 6.1.8.11) was shown rendered with no environment, but raytraced reflections.

For the cubic environment, six images of the room are rendered first and then mapped onto the environment cube as shown in Figure 6.1.8.22 (a). A surface style that includes the Environment shader with the Mapped item selected from the Global pop up menu (Reflection pop up menu in the Surface Style Parameters dialog) is assigned to the teapot. Cubic is selected from the Environment Type pop up menu in the RenderZone Options dialog. When the image is rendered, the reflections from the cubic environment appear as in Figure 6.1.8.22 (b).

For the spherical environment, the panoramic view of the room shown in Figure 6.1.8.23(a) is rendered and mapped onto the environment sphere. The Environment shader is assigned to the teapot as before, and Spherical is selected from the Environment Type menu. The resulting image with environment reflections is shown in Figure 6.1.8.23(b). Observe the similarities and differences between the cubic and spherical reflections.
**Center And Size:** When this button is selected, the **Environment Center And Size** dialog is invoked, as shown in Figure 6.1.8.24. It contains two major options:

**Infinite:** When this option is selected, the environment cube or sphere is considered infinitely large. The reflection pattern on a surface is calculated from the direction of a ray as it bounces off a surface. As a side effect of this approach, flat surfaces will show little or no variation in the reflected environment, since all the rays reflected off such a surface are exactly parallel in an axonometric view or nearly parallel in a perspective view. Parallel rays will always yield the same color for a pixel in an infinite environment.

**Fixed:** When this option is selected, the environment has a fixed size and is centered around a given point, which can be entered in the **Center** and **Size** fields. For rays reflected off a surface, the true intersection point with a cube or sphere of the given size and location is calculated. The color value of the pattern at that intersection point is used. Parallel rays reflected off a flat surface will now intersect the environment at different points, showing more of the pattern. The smaller the **Fixed Size**, the larger the reflected pattern will be. Care should be taken that the rendered scene fits inside the extent of the fixed size environment. Surfaces which fall outside of the environment will not show correct reflections. Pressing the Extent of Scene button automatically sets the size field, so that the scene fits entirely inside the environment.

![Environment Center And Size dialog](image)
**Rendered**: The option in this group affects reflections on those objects in the scene that have been assigned the Environment reflection shader that uses the Global Rendered option. The reflections come from objects in the scene to which the Environment shader has not been assigned. Surfaces to which the Environment shader has been assigned do not reflect each other. While the reflective shaders, such as Mirror, produce accurate reflections, they also require additional computing time. In contrast the Environment shader produces simulated reflections and is generally faster. This is described in more detail in subsection 6.2.4.

When an Environment reflection shader with the Global Rendered option has been included in the surface styles assigned to one or more objects in a scene, reflections are simulated through the use of an environment cube. Using as viewpoint the center (average coordinates) of all the surfaces to which the Environment reflection shader has been assigned, views of the other objects (those that are not assigned the Environment reflection shader) are generated looking into the direction of each of the squares of the environment cube. These directions are the X, Y, and Z axes of the world space, and views are taken in both their positive and negative directions. These images are then mapped onto the squares of the environment cube, the same way in which precaptured images are mapped when the Cubic mapped environment is used. From this point on, the reflections are calculated from the environment cube as for Cubic.

The resolution of the rendered images is controlled by the Render At n Pixels parameter. The value \( n \) entered in its numeric field determines the size at which the images of the objects to be reflected will be rendered.

An example of an image using rendered environment reflection is shown in Figure 6.1.8.25. The surface style assigned to the teapot uses the Environment reflection shader with the Global Rendered option. The surface style assigned to the room does not. Six rendered images of the room are created and attached to the environment cube. The image is then rendered as with the Cubic environment.

Which portion of an environment is reflected depends on the direction of a reflected ray. This is why reflections generated from spherical, cubic, or rendered environments work best on curved surfaces. For example, a spherical object will reflect a large portion of the environment, since rays bouncing off its surface will point in almost all directions. A cube on the other hand will reflect only a very small portion of the environment, since all reflected rays off each of the sides of the cube are pointing in the same direction, resulting in the same pixel of the environment being assigned to the surface.

![Figure 6.1.8.25: The teapot rendered using Rendered environment reflections.](image-url)
6.1.9 Depth effects

This section of the Scene tab in the RenderZone Options dialog contains the Depth Effect menu, which has five items (Depth Cue, Fog, Rain, Snow, and Snow 3D). They control different depth effects through parameters set in dialogs invoked by clicking on the Options... buttons.

**Depth Cue**: This depth effect blends a color with the surfaces of objects that lie within a region which may be defined either relative to the viewpoint or relative to the clipping (hither and yon) planes. One of two methods can be selected from the **Depth Cue Options** dialog (Figure 6.1.9.1).

**Distance From View Point**: When this option is selected, the positions of the beginning and end of the depth region are defined as distances from the viewpoint, through the values entered in the Start Distance and End Distance fields.

**Distance From Hither/Yon**: When this option is selected, the range of the depth region is defined relative to the clipping planes. The Start From Hither parameter, which is a percentage, defines the position of the beginning of the depth region relative to the hither plane. **End From Yon** defines the position of the end of the depth region relative to the yon plane. Both of these parameters can be set through slider bars or typed in the text fields.

The **Distance From View Point** and **Distance From Hither/Yon** groups of parameters are correlated. When a parameter in one group changes, the values displayed in the other group are automatically adjusted. If, after the values are set in this dialog, the positions of the hither and yon planes are changed, whether new values are calculated on the basis of the hither/yon planes or the viewpoint depends on the option selected in this dialog.

**Color**: When this option is selected, the color with which surface pixels are mixed is specified by the color in the box next to it, which can be changed by clicking on it.

**Background Mix**: This parameter indicates what percentage of the depth effect’s color will be mixed with the background. A mix of 0% renders the background as if no depth effect were specified. A mix of 100% renders the background with the depth effect color, regardless of which background shader is selected. Values between these two mix the depth effect color and the background at the specified percentage.

**Current Background**: When this option is selected, the colors with which surface pixels are mixed are taken from the current background shader. That is, the further back from the viewpoint an object is located, the more it will appear to blend into the background image. If the current background shader is **Alpha Channel**, the amount of blending of a pixel is contained in the alpha channel of the rendered image.
Figure 6.1.9.2 shows an example of how a depth effect can be applied. In (a) a scene is rendered without depth effects. In (b) the **Depth Cue** shader has been used, as follows:

- The **Sky** background shader is applied, generating a white cloud pattern on a dark sky.
- From the **Depth Effects** pop up menu **Depth Cue** is selected.
- In the **Depth Cue Options** dialog, the **Current Background** option is selected.

When the image is rendered, each pixel is blended with whatever color is assigned to the background at that image location. As a result, the model appears to blend in with the background cloud pattern. The further away a surface is from the viewer, the less visible the image is.

*Figure 6.1.9.2: A scene rendered (a) without and (b) with depth effects.*
**Fog**: When this option is selected, the current image will be rendered with fog effects. In the **Fog Options** dialog (Figure 6.1.9.3), the basis of the fog may be set as a distance from the viewpoint, or relative to the position of the hither plane. The fog effect is the strongest when its base is set close to the hither plane (**Distance From Hither** = 0%). The farther from the hither plane (and the viewer) it is set, the weaker the fog effect becomes.

The **Distance From Viewpoint**, **Distance From Hither**, **Color** and **Background Mix** options work similar to those found with the **Depth Cue** shader.

The fog can also be limited to a specified height above the ground plane if you select the **Ground Fog** option in the lower half of the dialog.

**Ground Fog**: Selecting this option enables the ground fog.

**Ground Plane**: The reference plane selected from the pop up menu determines the orientation of the ground plane.

**Ground Is n Above Plane**: The value entered in this numeric field indicates how high the ground plane is above the reference plane.

**Fog Height**: The value entered in this numeric field indicates how high the fog is above the ground plane.

Figure 6.1.9.4 shows a scene rendered with ground fog.
**Rain:** This depth effect adds streaks of rain drops to a scene. Its dialog is shown in Figure 6.1.9.5.

**Visibility Of Drops:** This determines how translucent the rain drops are. Smaller values yield less visible drops, and higher values yield more visible, almost snow-like drops.

**Direction Angle:** This angle determines the direction the rain drops are falling. Negative values cause the rain to fall to the left and positive values cause the rain to fall to the right.

**Density:** This value decreases or increases the overall number of rain drops present in the scene.

**Max Distance:** This value is the distance in the scene where the maximum number of rain drops will be seen.

**Snow:** When you select this option from the pop up menu, simple snow flakes are created in a near and far distance, relative to the view position. The parameters for using this depth effect are set in the **Snow Options** dialog (Figure 6.1.9.7).

**Near/Far Scale:** These parameters determine the size of the snow flakes close to, and far away, from the viewer.

**Flake Density:** A high density value creates more snow flakes in the same area, simulating a heavy snowfall. Lower density values simulate lighter snowfall.

**Flake Color:** Clicking in the color box invokes the standard color dialog in which you can select a color for the snow.

**Noise:** This option is the same as for all shaders with random patterns. The **Noise** option is discussed in detail in subsection 6.2.2.

Figure 6.1.9.8 shows a scene rendered with snow.
**Snow 3D**: This depth effect takes a volumetric approach to placing snow flakes. Hence, the snow flakes are distributed throughout the volume of the modelling scene, not just painted on top of the final image. Its dialog is shown in Figure 6.1.9.9.

**Visibility Of Snow Flakes**: This value determines how white or translucent the snow flakes are.

**Density**: This value decreases or increases the overall number of snow flakes present in the scene.

**Snow Flake Size**: This value decreases or increases a size of each snow flake.

**Blizzardy Noise**: This gives the snow flakes a more random look.

**Max Distance**: Snow flakes are calculated up to this distance in the scene.

**Accumulation**: If this option is on, the upward surfaces collect snow.

**Accumulation Scale**: This value determines how dense are the patches of snow that accumulate. If no patches are desired, set this to 100%.

**Accumulation Amount**: This is the percentage that the patches are solidly covered with snow.

**Accumulation Fuzz Amount**: This is the percentage at which solid snow cover fades to nothing for the patches. Should be set higher than the **Accumulation Amount**.
6.1.10 Postprocessed rendering effects

The Postprocess tab in the RenderZone Options dialog, shown in Figure 6.1.10.1 contains options that determine whether any postprocessing of a rendering scene will occur and how.

Postprocessed effects are applied to an image after a rendering has been completed. Unlike other rendering effects, changing the parameters of a postprocessed effect does not require the image to be rerendered. Postprocessed effects start with an already generated image and yield a new image which incorporates the postprocessed effects. The significant implication of this is that the postprocessed effects can be changed after their initial generation without rerendering the image. The implementation of the postprocessed effects is aimed at maximizing the efficiency gained by an interactive evaluation of postprocessed effects.

In form•Z RenderZone Plus, postprocessed effects can be set up before a rendering has been generated. This is done from the RenderZone Options dialog. Postprocessed effects can also be revised after a rendering has been completed. This is done through the RenderZone Postprocess... item in the Display menu.

Figure 6.1.10.1: The Postprocess tab in the RenderZone Options dialog.
Applying the Alpha Channel

Apply Alpha Channel: When this option is on, the rendered image is displayed with the alpha channel filter applied to it. This is discussed in more detail in section 6.1.8.

Note that this option is only available if the Alpha Channel item is selected from the Background shader menu.

Enabling postprocessing

In order for postprocessed effects to be applied, certain information about a rendering must be maintained after the rendering is completed. This is done by turning on the following options in the RenderZone Options dialog:

Generate Postprocess Data: When this option is on in the Postprocess tab of the RenderZone Options dialog, the data necessary to apply postprocessing is maintained and saved with the image. Also, the three postprocess options Exposure Correction, Depth Blur, and Lens Flares are made available when this option is on. Note that these options are not available when partial images are rendered.

If any of the postprocessing options are selected, the respective effect is automatically applied to the image after the rendering is completed. The order in which the effects are applied is the same as the order of the options in the dialog.
Postprocessing after a rendering is completed

In addition to setting up postprocessed effects prior to a rendering, they can also be reset and regenerated after the rendering is completed, which is actually quite common. Because the postprocessed effects greatly depend on the outcome of the rendering, it is typically difficult to predict how they will turn out. Consequently, regenerating them until they appear satisfactory is usually desirable and is made practical and possible by the fact that postprocessed effects compute relatively fast. After their initial generation, the parameters of the postprocessed effects can be adjusted and the final image regenerated at a reasonable speed, without having to rerender the basic image. This is accomplished by using the RenderZone Postprocess... item found in the Display menu (Figure 6.1.10.2).

This item is only available when the RenderZone* display type is active in the Display menu and the Generate Postprocess Data option was on in the RenderZone Options dialog when the image was rendered. It is not necessary that any of the options under Generate Postprocess Data are actually turned on in order to be able to recompute the postprocessed effects.

Selecting the RenderZone Postprocess... item invokes the RenderZone Postprocess Options dialog, shown in Figure 6.1.10.3. This dialog contains the same options as the Postprocess section in the RenderZone Options dialog plus Background, which allows you to change the background of an image through postprocessing.

The options that are common in the RenderZone Options and RenderZone Postprocess Options dialogs work the same, except that the dialogs invoked by the Options... buttons in the latter include a preview area that displays the rendered image on a smaller scale with the current settings of the postprocessed effects applied to the image. Every time a parameter is changed, the image is regenerated. Parameters can be edited until a satisfactory result is reached. The postprocessed effects are applied permanently to the image when exiting the RenderZone Postprocess Options dialog.

For an option to be active for postprocessing in the RenderZone Postprocess Options dialog, it must have also been active in the RenderZone Options dialog when the basic image was rendered. Note that Background and Apply Alpha Channel can not both be active at the same time. The latter will be active when Alpha Channel was applied as background to the rendering. The former will be active for all other backgrounds.
Postprocessing the background

**Background**: When this option is selected, it is possible to change the parameters of the current background, or even select an entirely new background without rerendering the image. With this option on, pressing the **Options...** button in the **RenderZone Postprocess Options** dialog invokes the **Background Options** dialog, shown in Figure 6.1.10.4.

Assuming a scene was rendered with a **Graduated** background, when **Options...** is pressed, the **Background Options** dialog displays in its preview area the image with the graduated background, as shown in Figure 6.1.10.4(a). The parameters of the graduated background can be changed or the background may be changed to a different one, such as **Sky**. When it is, the preview area is refreshed to display the image with the new background, as shown in Figure 6.1.10.4(b). This way a variety of different backgrounds may be tried until the most suitable one is selected.

New backgrounds are selected from the **Background** pop up menu, located under the preview area. The parameters of the background currently selected may be set by pressing the **Options...** button next to it, which invokes the respective options dialog. These dialogs are the same as the background dialogs invoked from the **RenderZone Options** dialog. Each time a parameter is changed or a new background is selected, the preview is regenerated to show the effects of the new parameters or new background.

Note that the **Background** option is not available when the image is rendered with the **Alpha Channel** background. In the latter case the **Apply Alpha Channel** option is available for postprocessing.

*Figure 6.1.10.4*: The **Background Options** dialog with (a) **Graduated** and (b) **Sky** backgrounds.
Adjusting the image exposure

Through postprocessing, the level of exposure of an image can be adjusted, after it has been rendered, using the following option in the Postprocess tab:

**Exposure Correction**: When this option is on, clicking on the Options... button next to it invokes the Exposure Correction Options dialog, shown in Figure 6.1.10.5.

![Exposure Correction Options dialog](image)

**Figure 6.1.10.5**: The Exposure Correction Options dialog.

The human eye is able to adapt to a wide range of light intensities. For example, we can see details in a dark room illuminated by a candle as well as in bright sunlight. The difference of light intensities between the two environments spans over nine orders of magnitude. Photo paper or electronic display devices, such as TV monitors or computer screens, have a much more limited range. This is why photographs become overexposed when trying to capture details in scenes with areas of bright and dim regions.

In order to show the detail in the dim region, a long exposure needs to be selected, which in turn overexposes the photograph in the bright region. The same artifact can be observed with rendered images displayed on a computer screen. To correct this problem, it is possible to map a large intensity range to the smaller range of the screen by compressing intensities in overexposed areas and by inflating intensities in underexposed areas.

The dialog contains a pop up menu from where one of three methods of correction can be selected.

**Type**: This pop up menu contains three exposure correction methods: Spline, Automatic, and Brightness & Contrast. The default is Automatic, which has only two parameters to set up and is rather simple to use. So is Brightness & Contrast. The Spline method is more involved. It has more parameters to set up but offers significantly more control of the result.
Spline: When this item is selected from the Type menu, a diagram and a number of parameters for controlling Luminance and Brightness are displayed in the dialog, as shown in Figure 6.1.10.6.

The graph shows to which screen brightness value a particular image intensity is mapped (Figure 6.1.10.7). The horizontal axis represents the image intensity ranging from 0 to the value entered in the Mapping Range field. The vertical axis defines the screen brightness. The curve is drawn as a 4-point spline.

- First control point (Min), at the lower left corner, sets to which screen brightness the darkest intensity in the image is mapped.
- Second point (Dark) maps a dark intensity to a screen brightness.
- Third control point (Light) maps a light intensity to a brightness.
- Fourth control point (Max) defines the mapping of the maximum intensity of the image to a screen brightness.

What is perceived as a dark or light intensity can be changed by moving the second or third point horizontally. The brightness can be changed by moving a point vertically. The corresponding numerical values for intensity and brightness are displayed in the text fields under the graph. The intensities of the first and of last points are always 0% and 100%, which cannot be edited. The brightness for any intensity in the image can be found by determining the percentage of the intensity along the horizontal axis, tracing vertically to the mapping curve and then tracing horizontally to the vertical axis, where the brightness is found (Figure 6.1.10.8).

If an image is generally overexposed, it is necessary to turn down the brightness in light areas, while maintaining the detail in dark areas. This is done by moving the Dark point up and the Light point down. A high brightness value for the Dark point lightens dark areas in the image and a low brightness for the Light point dampens bright areas. This is illustrated in Figure 6.1.10.9.
Figure 6.1.10.9: Exposure correction for generally overexposed images: (a) original image, (b) corrected image, and (c) mapping curve of corrected image.

If an image contains areas of both very high and very low intensities, the contrast in the image needs to be decreased, while brightening the dark areas. Moving the Dark control point up high along the brightness scale lightens the image in dark areas. A high contrast image and the corrected version are illustrated in Figure 6.1.10.

Figure 6.1.10.10: Exposure correction for high contrast images: (a) original image, (b) corrected image, and (c) mapping curve of corrected image.

If an image has very little contrast, the control points can be adjusted in such a way that intensities which are just below a certain value become darker, whereas intensities which are above this threshold become lighter. This results in an increased contrast, as illustrated in Figure 6.1.10.11.

Figure 6.1.10.11: Exposure correction for low contrast images: (a) original image, (b) corrected image, and (c) mapping curve of corrected image.
**Max Luminance**: The top of two information fields located under the preview area shows the luminance value in lux of the brightest spot in the image.

**Sample Luminance**: The second field under the preview area shows the luminance value of the pixel to which the mouse currently points. This allows for interactive measuring of the luminance values in an image and helps determine which luminance value will be mapped to which screen brightness value on the mapping curve. Red lines are drawn from the luminance scale on the bottom to the mapping curve to the brightness scale, as shown in Figure 6.1.10.12.

![Exposure Correction Options](image)

**Mapping Range**: The value entered in this field determines the range of luminance values into which the horizontal scale on the bottom is subdivided. It always starts with 0 at its left and ends with the value entered in this field at its right. Values in the image that fall within this range are mapped to the matching screen brightness value on the vertical brightness scale. Combined with the ability to interactively sample luminance values as mentioned above, it is easy to find a suitable exposure correction for an under or overexposed image. A process of how this might be done is summarized next, using the scenes in Figures 6.1.10.12 and 6.1.10.13.

- Through interactive luminance sampling, find the lowest luminance spot in the image that should be shown with maximum screen brightness.

- Set the mapping range value to this luminance. Now, all the pixels in the image having a lower luminance than this spot are mapped between the darkest and brightest screen brightness, which is the visible range of the display monitor. You may also want to set the mapping range value slightly lower, which will help preserve bright hot spots.

Luminance values that are higher than the mapping range will be clamped to the maximum screen brightness. Changing the **Max Brightness** field to a value larger than 100% may help to further refine the exposure. Now, the vertical brightness scale on the left extends beyond the display capabilities of the monitor, as shown in Figure 6.1.10.13. It may be desirable that only the left portion of the mapping curve matches luminance values to the displayable portion of the brightness scale. Consider the example below, which shows an interior rendering, where the wall is illuminated through a regular light fixture, and the floor shows illumination from the sun light shining through the window. If physically correct light intensity values are used, the sun light is several orders of magnitude stronger than the artificial light. This leads to overexposure on the floor and underexposure on the wall. To correct this the following steps can be taken:
• Set the mapping range to 1200 lux. The sun lit area, which has a luminance of about 1000 lux, can now be mapped to a visible brightness.

• Set Max Brightness to 200. This leaves some room for the mapping curve to extend beyond the visible brightness.

• Set Dark Luminance to 5% and Dark Brightness to 100%. This means that pixels with at least 5% of 1200 lux (= 60 lux) are mapped to 90% screen brightness, which brightens all the areas illuminated by the artificial light that have luminance values between 0 and 100 lux.

• Now set Light Luminance to 90% and Light Brightness to 110%. This brings the pixels of the sun lit area into visible range, since their luminance of around 1000 lux is now mapped to between 90% and 110% of the screen brightness.

Automatic: This method automatically compresses all overexposed regions in an image into the display range of the monitor. Likewise, underexposed areas are amplified to become brighter. This is achieved by first measuring all intensities in the image and then creating a histogram from which the automatic correction is generated. The two parameters shown in Figure 6.1.10.14 allow the user to further fine tune the exposure.

Bias: At 50% this parameter has a neutral effect on the image exposure. If it is set to values larger than 50%, the middle range of the image intensities will be brightened, which will decrease the contrast between bright and medium bright areas. It also increases the contrast between medium dark and very dark areas. Values below 50% have the opposite effect. Contrast between bright and medium bright areas is increased and decreased between medium and dark regions.

Brightness: Some images may become too dark with the default correction. Increasing the value of this slider lifts the overall brightness of the image, without changing the contrast between light and dark regions. What the appropriate value of this parameter is, depends greatly on the image. In some instances it may be necessary to increase the Brightness value above 100%.
**Brightness & Contrast:** As with Automatic, this method also maps the range of true image intensities to the display range of the monitor. The two parameters shown in Figure 6.1.10.15, **Brightness** and **Contrast** are straightforward. Increasing **Brightness** makes the whole image uniformly brighter and makes it darker, if it is decreased. Manipulating the **Contrast** parameter allows for control over decreasing or increasing the difference between dark and light regions.

Unfortunately, there is no single good setting that works well for all images. Nor is either one method better suited than the other. In general, the two latter methods are more automated than the Spline method and are easier to control. Since the latter two methods only have two parameters, it is fairly easy to experiment with the slider settings and quickly find which values improve the image exposure. Note that in some cases even small value changes can cause significant differences in the image exposure, especially towards the high range (100%) of the parameter. More importantly though, the simpler two methods allow even significantly overexposed images to be corrected yielding usable final results. Therefore, one does not have to worry too much in the initial phases of a final gather and radiosity based renderings about light intensities. As long as the image has a decent amount of contrast in it, overexposed or not, the new exposure correction methods will almost always create a usable result. Figure 6.1.10.16(a) shows a scene that was rendered with final gather and radiosity support, but has too much light, causing overexposure. The same image is shown in Figure 6.1.10.16(b) with the **Automatic** method applied.

**Figure 6.1.10.15:**
The Exposure Correction Options dialog when **Brightness & Contrast** is selected.

**Figure 6.1.10.16:**
(a) A Final Gather rendering with uncorrected exposure.
(b) The same image after the Automatic correction method was applied.
Depth blurring

**Depth Blur:** When this option is on, a blur effect, similar to that of a photograph can be generated. Objects which are in front or behind the focused area appear fuzzy. The respective parameters are selected from the **Depth Blur Options** dialog (Figure 6.1.10.17), invoked by clicking on the **Options...** button next to it.

**Distance From View Point:** When this option is selected, the start and end of the focused area are expressed as distances relative to the eye point of the current view.

**Distance From Hither:** When this option is selected, the start and end of the focused area are expressed as relative distances along the cone of vision between the hither and yon clipping planes.

**Maximum Near/Far Blur:** The values entered in these fields determines the amount of fuzziness of objects at the eye point and at the location farthest away from the eye point. A value of 100% will apply blur to an area of up to 10 pixels around an object.

**Blur Background:** When this option is selected, background pixels are blurred according to the value entered in the **Maximum Far Blur** field. If this option is not selected, no blur is applied to the background.

The image in Figure 6.1.10.18 illustrates depth blurring generated by the default blur parameters.

*Figure 6.1.10.17:* The **Depth Blur Options** dialog.

*Figure 6.1.10.18:* An image rendered with depth blur.
Generating flares effects

**Lens Flares**: When this option is on, clicking on the **Options** button invokes the **Lens Flare Options** dialog, shown in Figure 6.1.10.19. Through this dialog the lens flare parameters can be changed and their effects previewed in the dialog.

Lens flares are geometric patterns caused by light rays refracting in optical camera lenses. They are common, when a camera is pointed directly at a light source or at very bright spots in a scene. In **form•Z RenderZone Plus** such lens flare patterns can be added to images as postprocessed effects. Lens flares can be generated from any active light source, except distant lights. They can also be generated from hot spots on glossy surfaces. A variety of flare shapes are available.

The preview area of the **Lens Flare Options** dialog initially shows the rendered image without lens flares. It uses yellow, blue, and white bullets to identify the location of the lens flares about to be generated.

- **Yellow** bullets indicate the centers of lens flares created from light sources. Clicking on a yellow bullet turns it off. This is the same as clicking in the **Lens Flare** option in the **Light Parameters** dialog.

- **Blue** bullets show light sources, which have the **Lens Flare** option off in the **Light Parameters** dialog. Clicking on a blue bullet turns on a light’s lens flare.

- **White** bullets stand for lens flares from hot spots on glossy surfaces.

The number of lens flares occurring in an image depends on the **Range** values, as described in more detail on the following pages.

**Sample Luminance**: This field, located under the preview area, shows the luminance value of the pixel to which the mouse is pointing. It is blank when the mouse cursor is outside the preview area. If the mouse cursor is inside a yellow light bullet, the luminance value of a light source is shown. Note that the luminance value of a light source depends on the intensity setting and falloff option of the light, which is set in the **Light Parameters** dialog. If the yellow or blue bullet is drawn with a diagonal cross, the intensity of the light source is not strong enough to create a lens flare. Lens flares created from hot spots are generated in such a way that one flare is placed on a number of pixels, identifying a hot spot. For example, the hot spot of a sphere may have 200 bright pixels. The lens flare is placed in the center of these pixels, using the brightest pixel as the luminance input. Again, moving the mouse over the white bullet shows the exact luminance value used for the hot spot.
Show Original Image: When this option is selected, the image is shown without lens flares but with the bullet locations.

Show Lens Flare Preview: When this option is selected, the lens flares are generated and the preview image is shown. Next to the Show Lens Flare Preview option is a button called Update. Anytime a parameter in the dialog is changed, it becomes active and allows the preview to be regenerated when it is clicked. If the preview is up to date, the button is dimmed. Since lens flares may take longer to generate, any click in the dialog while the preview is computed will stop it.

Shape: A geometric shape for the lens flare can be selected from this pop up menu. A preview of the shape is shown in the area below the menu. The available shapes are shown in Figure 6.1.10.20.

Colors: This group displays the colors currently assigned to the center Glow, Halo rings, and lens flare Rays. These can be edited by clicking in their respective color boxes, which invokes the standard color picker dialog.

Range: The values entered in these three fields determine conditions under which lens flares are generated and whether lights will be amplified.

Min Luminance, Max Luminance: The values entered in these fields determine when a lens flare is generated from a light source or a hot spot, as follows:
• If a light or hot spot has a luminance less than Min Luminance, no flare is generated.
• If it is at least Max Luminance, a maximum size flare is created.
• Luminance values between Min and Max scale the size of the flare.

Light Amplifier: The value entered in this field amplifies the intensity of all light sources. This allows for light sources to become stronger, relative to the hot spots in the scene.
Intensity: The value entered in this field determines how intensely the lens flares are drawn. For example a white lens flare on a black background will show as white at 100% intensity, medium gray at 50%, and not at all at 0%.

Size: The value entered in this field determines the size of the flare, relative to the image size. At 100% the lens flare fills the entire image, if located at the image center.

Rotation: This value determines the rotation of the rays and polygonal ghosts around their respective centers.

Flares From Hot Spots: When this option is selected, lens flares will be generated from hot spots. Otherwise hot spots will be ignored.

Flares From Light Sources: When this option is selected, lens flares will be generated from light sources. Otherwise light sources will be ignored.

Ignore Light Falloff: When this option is selected, the falloff parameter of a light is ignored when determining the luminance of a light source. Since the luminance of a light source decreases as it moves away from the view point, light sources far away from the view may not show up. Turning this option on causes all light sources to have a constant luminance, regardless of their distance from the viewer.

Use Out Of View Lights: When this option is selected, light sources that are just outside the view are included in the lens flare process. While the center of these flares is not visible, the rays, glow, and halos may pour into the image. This option may be useful during an animation, which will softly move a lens flare in and out of the view, as opposed to a lens flare suddenly showing up as soon as the light enters the scene.

Use Obscured Lights: When this option is selected, light sources that are hidden behind objects still create a lens flare. This option may be useful for creating lens flares from lights that are behind a transparent surface. If this option is not selected, lights that are behind rendered surfaces, transparent or not, are ignored.

Note that the lens flares generated by rendering programs are typically exaggerated rendering effects and not accurate simulations of the reflective behavior of optical lenses or other glossy surfaces. This is also true with the lens flares generated by form•Z. They merely superimpose a predetermined shape over an existing rendering by considering the brightness and location of hot spots and light sources. As such, it is not possible, for example, to accurately compute the intensity and color of a light source located behind a colored transparent surface.
Some lens flare examples

Figure 6.1.10.21 shows samples of renderings with lens flares added. The models that are rendered are included with the form•Z CD, where they are saved together with the parameters used to generate the lens flares shown. These parameters are also listed below:

<table>
<thead>
<tr>
<th>Lantern</th>
<th>Sconce</th>
<th>Lamp</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shape</strong>: Sparks</td>
<td><strong>Shape</strong>: 50 mm Lens</td>
<td><strong>Shape</strong>: Stars</td>
</tr>
<tr>
<td><strong>Colors</strong>:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glow RGB=100, 66, 19</td>
<td>Glow RGB=100, 70, 88</td>
<td>Glow RGB=100, 91, 40</td>
</tr>
<tr>
<td>Rays RGB=100, 60, 59</td>
<td>Halo RGB= 25,100, 56</td>
<td>Rays RGB= 75,100, 99</td>
</tr>
<tr>
<td><strong>Range</strong>:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min Luminance: 300 Lux</td>
<td>Min Luminance: 300 Lux</td>
<td>Min Luminance: 250 Lux</td>
</tr>
<tr>
<td>Max Luminance: 1000 Lux</td>
<td>Max Luminance: 1000 Lux</td>
<td>Max Luminance: 600 Lux</td>
</tr>
<tr>
<td>Light Amplifier: 10</td>
<td>Light Amplifier: 10</td>
<td>Light Amplifier: 10</td>
</tr>
<tr>
<td><strong>Intensity</strong>: 100%</td>
<td><strong>Intensity</strong>: 85%</td>
<td><strong>Intensity</strong>: 100%</td>
</tr>
<tr>
<td><strong>Size</strong>: 200%</td>
<td><strong>Size</strong>: 55%</td>
<td><strong>Size</strong>: 75%</td>
</tr>
<tr>
<td><strong>Rotation</strong>: 15˚</td>
<td><strong>Rotation</strong>: 15˚</td>
<td><strong>Rotation</strong>: 15˚</td>
</tr>
<tr>
<td>Flares From Hot Spots:</td>
<td>Flares From Hot Spots:</td>
<td>Flares From Hot Spots:</td>
</tr>
<tr>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>Flares From Lights:</td>
<td>Flares From Lights:</td>
<td>USE Obscured Lights:</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>Ignore Light Falloff:</td>
<td>Ignore Light Falloff:</td>
<td>Ignore Light Falloff:</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>Use Out Of View Lights:</td>
<td>Use Out Of View Lights:</td>
<td>Use Out Of View Lights:</td>
</tr>
<tr>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>Use Obscured Lights:</td>
<td>Use Obscured Lights:</td>
<td>Use Obscured Lights:</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
</tbody>
</table>

![Lantern Image](image1)

![Sconce Image](image2)

![Lamp Image](image3)

Figure 6.1.10.21: Examples of lens flares.
6.1.11 Loading and manipulating image maps

form•Z RenderZone Plus is shipped with 75 precaptured textures (see Appendix of this volume), and additional texture images may be produced by the user. A texture image file may contain a single image or it may be a movie file. When loading a movie file as an image map, only one frame of the movie will be used as the image. Precaptured images may be used by a variety of form•Z RenderZone Plus procedures. They may be used to create environment maps by mapping them onto the squares of the environment cube or onto the surface of the environment sphere (see section 6.1.8); they may be used as a background (see section 6.1.8); they may be used to attach precaptured images to surfaces of objects by using the Color Map shader (see section 6.2.3); they may be used to apply reflection and transmission parameters by the respective shaders (see section 6.2.4); they may be used as transparency filters by the Transparency Map shader (see section 6.2.5); and they may be used to delineate bumps through the Bump Map shader (see section 6.2.6). In all these cases, the loading of an image and other available operations are executed through a dialog which contains most of the features of the Cubic Environment Map Options dialog, shown in Figure 6.1.11.1. This dialog is used to assign images to the environment cube discussed in section 6.1.8. Most of the image mapping dialogs contain additional options that apply to the particular use of the image map. The general features of the image mapping dialogs are discussed in this section. The additional features are discussed in the respective sections.

The image mapping dialogs (Figure 6.1.11.1) consist of an image view area, a file information area, the standard Cancel and OK buttons, and four button commands that allow you to locate an image file, reset the image, and view and crop the currently loaded image. The types of image files that can be used in form•Z RenderZone Plus partially vary between the Windows and Macintosh operating systems. For example, WMF is only available on the former and PICT only available on the latter. Targa and TIFF are available on both. You can inspect the complete list of acceptable types by clicking on Load... to invoke the Open File dialog.

File Info: This is a non editable text area that lists the name, type, size, resolution, and location of the loaded image file. If the texture file is a movie, the duration and frame rate information are also displayed.

Frame Time: When the texture file is a movie file, this field allows the user to specify which frame of the movie will be used as the image. The frame is specified as a time value, which forwards the movie to that time. The maximum time that can be entered is the length of the movie, which is shown in the File Info section above.

Default: Clicking on this button resets the loaded image to the system default image file.
Load...: Clicking on this button invokes the Open File dialog, from which you can select the image to be loaded. Closing the Open dialog causes the image to appear in the preview window.

View...: This button is used to view the loaded image in a window in which it can be cropped. Clicking on it invokes the View window shown in Figure 6.1.11.2. The View window can also be invoked by clicking in the preview area of the image mapping dialogs. This window displays the currently loaded image bounded by a rectangular frame marked with little square handles. On the top margin of the window, a tool palette containing three icons is displayed. These tools can be used to manipulate the image, as described below.

Clicking with this tool on a part of the bounding rectangle crops the image as follows:
- Clicking on a corner and dragging it repositions the corner.
- Clicking on a mid-segment handle and dragging it repositions the segment.
- Clicking on a segment (away from a handle) and dragging it moves the whole rectangle.

Clicking with this tool inside the image enlarges it.
This icon replaces the zoom tool when pressing the option (Macintosh) or ctrl+shift (Windows) keys. When selected, clicking inside the image reduces its size.

Clicking on the image with this tool and dragging the mouse pans the image.

Note that all moves are restricted to within the boundaries of the image and can also be restricted to certain proportions by using modifier keys. Pressing shift when a corner point or a segment of the bounding rectangle is moved, restricts the movement to the proportions of the image. Pressing option (Macintosh) or ctrl+shift (Windows) restricts the movement to square proportions. Pressing shift while moving the entire rectangle constrains the move to the horizontal and vertical directions.

The following information about how image maps are handled by the program should help you use them efficiently, since precaptured texture maps may easily consume significant quantities of memory. When an image file is loaded to generate a texture map, a bitmap representation of the image is stored in memory. The memory requirements for a texture map can be calculated as:

For Color, Transparency, Reflection, and Bump Maps:
- MIP map sampling using: RGB channels: \[\text{bytes required} = \text{width} \times \text{height} \times 1.33 \times 3\]
- alpha channel: \[\text{bytes required} = \text{width} \times \text{height} \times 1.33\]
- Summation table sampling using: RGB channels: \[\text{bytes required} = \text{width} \times \text{height} \times 4 \times 3\]
- alpha channel: \[\text{bytes required} = \text{width} \times \text{height} \times 4\]

For Background and Environment Maps:
\[\text{bytes required} = \text{width} \times \text{height}\]

A word of caution about using texture maps. It is best to choose the resolution of a texture map based on the area it occupies in a rendered image. The two should not differ by a large amount. That is, if a texture with a resolution of 1000 x 1000 pixels is used on an image area of 100 x 100 pixels, a lot of memory is wasted with no gain in image quality. On the other hand, a 100 x 100 pixel texture applied to a 1000 x 1000 area on the screen will show significant artifacts.

form•Z RenderZone Plus • Producing a rendered image
6.2 Surface styles

Surface styles can be thought of as colors, and plain color is actually the simplest form of a surface style. A complete surface style is displayed by superimposing on a color additional types of attributes which control how a surface interacts with lights and the environment. Specifically, a surface style consists of four classes of attributes applied through distinct sets of shaders: color, reflection, transparency, and bumps.

Color shaders are used to define the color of a surface. The simplest color shader assigns a uniform color to the entire surface. Other color shaders generate more complex patterns, such as marble or wood.

The behavior of a surface when illuminated by a light depends on a reflection shader, which determines how much light is reflected from the surface toward the view point and how much light is transmitted through the object. Shaders of this class may be thought of as defining the finish of a surface, and are used to simulate material properties such as matte, metal, plastic or glass.

Transparency shaders define the transparency or opacity of a surface. A transparency shader may give a surface a uniform transparency, or it may be used to create regularly or irregularly eroded patterns which would be difficult to generate using modeling techniques.

Small surface variations can be created using bump shaders. Typically, a bump shader gives an otherwise smooth surface an irregular or indented appearance. In other words, bump shaders produce three-dimensional effects which could potentially be made part of the 3D model, but are more efficiently produced by rendering. For example, rough metal castings and the regular indentations produced by pressed sheet metal can be simulated using bump shaders.
6.2.1 The Surface Styles palette and dialog

form•Z RenderZone Plus displays surface styles in the Surface Styles palette. When the program is launched for the first time, the surface styles appear in their default plain colors (Figure 6.2.1.1). If a previously saved project file, for which surface styles were defined, is opened, those styles will be displayed. A Surface Styles palette displaying surface styles other than the default is shown in Figure 6.2.1.2.

Three formats are available for showing the surface styles: a large image, a small image, and a name list display. You can switch from one format to another by selecting the respective button on the left end of the bottom margin of the palette. The images in the Surface Styles palette are three dimensional, and offer a preview of the surface styles rendered on the surfaces of planes, cubes, cylinders, or spheres. The form to be used is selected when the surface style is defined.

One of the surface styles is the active surface style and is highlighted. When a new object is generated, the active surface style is automatically applied to the object.

Surface styles are managed through the Surface Styles dialog (Figure 6.2.1.3), which can be invoked by clicking in the Surface Styles palette while pressing the option key on the Macintosh or ctrl+shift on Windows. This dialog contains a display area which is identical to the Surface Styles palette, and a number of button commands for managing the surface styles.

New...: When this button is selected, a new surface style is created. The Surface Style Parameters dialog is invoked, in which the shaders for the new surface style may be selected and fine tuned. This dialog is discussed in the next section.

Note that new surface styles can also be created directly from the Surface Styles palette by clicking once in the blank space of the palette. The new surface style is defined through the Surface Style Parameters dialog which is thus invoked.
**Delete...**: When this button is selected, the active surface style is deleted. All but one surface style can be deleted. If a surface style used by one or more objects in the project is deleted, the program presents the dialog shown in Figure 6.2.1.4, and requests that another surface style be selected for use in place of the style deleted.

**Edit...**: When this button is selected, the parameters of the active surface style are shown in the **Surface Style Parameters** dialog, where they can be changed.

**Copy**: When this button is selected, the active surface style is duplicated. The new copy is placed at the end of the surface styles list.

The surface styles are initially stored in the order in which they are created. This order can be changed through button commands available in the **Surface Styles** dialog.

**Top, Bottom**: Clicking on one of these buttons moves the active surface style to the top or bottom position in the list, respectively.

**Sort**: When this button is selected, the list of surface styles is sorted alphabetically, according to their names.

**Purge...**: When this button is selected, all surface styles which are not in use are deleted. An alert message is first posted to verify that the user truly intends to delete the unused surface styles.

**Load Surface Styles...**: When this button is selected, the standard Open File dialog is invoked. It allows you to select a **form•Z** file, whose surface styles will be brought in and added to the end of the Surface Styles palette.
6.2.2 Editing the parameters of a surface style

Surface styles are set in the Surface Style Parameters dialog (Figure 6.2.2.1), which is invoked when either the New... or Edit... button in the Surface Styles dialog is selected. It can also be invoked by double clicking on a surface style icon or name in the Surface Styles palette. It contains two tabs (RenderZone and Simple), as well as a few options outside the tabs.

Options outside the tabs

**Name:** A surface style name (up to 31 characters long) can be typed in this text field to replace that defaulted by the system.

**Preview:** This area displays the effects of the currently selected shaders, as discussed in more detail later in this section.

**Predefined.../Copy From...:** These buttons and their command functions are discussed in section 6.2.7.

The RenderZone tab

This tab (Figure 6.2.2.1(a)) contains four classes of shaders, each consisting of a pop up menu from which a shader can be selected (Figure 6.2.2.2). Next to each menu is a button command labeled Options.... Clicking on it invokes a dialog containing the parameters which affect the shader that is currently selected in the menu. The shaders and their dialogs are discussed in detail in sections 6.2.3 through 6.2.6.

The combination of shaders selected in this section determines how a surface is rendered using the RenderZone* display mode. Interactive Shaded* also uses these options but is able to display only a subset of the parameters defined by the shaders. This is discussed in more detail in section 3.6.2 of the form-Z manual.
The Simple tab

This tab (Figure 6.2.2.1(b)) contains the color and transparency value used by **Wire Frame**, **Quick Paint**, **Surface Render**, **Hidden Line**, and **Shaded Render**. If the **RenderZone Average** option is selected, the color and transparency are automatically computed as an average of the color and transparency shaders in the **RenderZone** tab. If **Custom** is selected the color and transparency can be edited manually. Clicking in the color box invokes the standard Color dialog. The **Transparency** slider and text field allow you to set a simple transparency value.

Types of shaders

There are four classes of shaders that control **color**, **reflection**, **transparency**, and **bumps**. For these classes of shaders, the defaults are **Plain Color**, **Matte**, **None**, and **None**, respectively. **Plain Color** generates a uniform surface color, **Matte** generates a simple shaded surface. **None** in the last two classes produces opaque objects and no bumps.

Shaders are either **procedural** or **precaptured texture maps**. Procedural shaders generate a pattern by executing a procedure or program. In other words, these effects are generated from scratch and do not use a prestored image. In contrast, the precaptured texture maps use an image file which is mapped onto a surface, either for painting the image on the surface, or for using the image as a transparency filter or as a bump map. The texture maps can be captured photographically, can be previously rendered images, or can be images created in a paint program.

The procedural shaders are further distinguished into two types: **wrapped** or **solid shaders**. In the shaders menus, wrapped and solid shaders are grouped together and are separated by a line. The grouping in each shader menu is discussed in more detail at the beginning of each respective section (Figure 6.2.2.2).

**Figure 6.2.2.2:** The shader menus: (a) Color, (b) Reflection, (c) Transparency, and (d) Bump.
Previewing the effects of the shaders

At the upper right corner of the **Surface Style Parameters** dialog, there is a preview image which displays the combined effects of the shaders and the parameters which are currently selected. Each time a new shader is selected or a shader parameter is changed, the image is regenerated and displays the new effects.

The currently defined surface style is previewed by applying it to the surface of a **Plane**, **Cube**, **Cylinder**, or **Sphere**, which is selected from a menu under the preview image. The preview object is shown on a checkered floor, which helps to show the effects of reflective and transparent surface styles. The preview image selected is used to display the surface style in the palette.
How procedural textures are mapped

The wrapped texture shaders produce two-dimensional patterns which are mapped onto the surfaces of objects. Exactly how they are mapped onto a surface can be determined by using the Texture Map tool, which is discussed in detail in section 6.3. If the Texture Map tool is not used to articulate the position of a texture on a particular object, a global default texture mapping is used to project a texture on that object. These default texture mapping parameters are defined in the Default Texture Map Control tab of the Project Rendering Options dialog, which is invoked from the Display menu and from within the RenderZone Options dialog.

Wrapped textures are mapped onto a surface by repeating a rectangular tile which carries a module of a texture. Arranging a number of modular tiles horizontally and vertically produces the overall pattern of a texture. These tiles can be of different sizes, which is determined by the number entered in the Wrapped Textures Horizontal and Vertical Tiling Size fields in the Project Rendering Options dialog. The values entered in these Size fields have an effect only when a wrapped surface is positioned by default. When the Texture Map tool is used to accurately position a wrapped texture on an individual object, the Size parameters that are set in the Texture Map Controls dialog for that object (discussed in section 6.3.2) override the global default Size parameters entered in the Project Rendering Options dialog.

Solid shaders produce textures in 3D space. The patterns that appear on the surfaces of objects result from the intersection of these surfaces with the 3D textures. The advantage of the solid textures is that they show continuity around edges, and objects using these shaders may be thought of as being carved out of a solid block of the material which the shader is generating. The unit size or scale of a solid texture is controlled by the Solid Textures: Size parameter in the Project Rendering Options dialog. Contrary to the tile size of the wrapped textures where independent X and Y dimensions can be defined, a single uniform size applies to the three dimensions of the solid textures. As with the wrapped textures, the value entered in the Solid Size field of the Project Rendering Options dialog has an effect only when a solid texture is positioned by default. When the Texture Map tool is used to accurately position a solid texture on an individual object, the Size parameter that is set in the Texture Map Controls dialog overrides the default global Size parameter entered in the Project Rendering Options dialog.

The origin of solid textures is, by default, positioned at the values entered in the Origin fields in the Texture Map Options dialog. This position, as well as the orientation of the texture axes, can be changed by using the Texture Map tool, as for the wrapped textures, except that the mapping types (flat, cubic, etc.) do not apply to the solid textures.
How surface styles are rendered by display types other than RenderZone

When plain color is used on surfaces, plotting and rendering types other than RenderZone can use this color, when the object is displayed in wire frame, quick paint, etc. When a complex surface style, which may even be getting its color exclusively from an image map, is used, there is no single color that can be used. In these cases the average color of a sample of pixels is calculated and that color is used for rendering types other than RenderZone. This color is displayed in the **Surface Style Parameters** dialog and can be changed using the following option.

**Color:** The box next to this option in the *Simple* tab of the **Surface Style Parameters** dialog displays the color which is used by rendering types other than RenderZone. By default, the **RenderZone Average** option is selected, which means the color is automatically computed as an average of the currently selected shader. The ability to edit the color is unavailable when **Custom** option is selected, and is therefore dimmed. If the **RenderZone Average** checkbox is deselected, the color becomes active and available for editing. Clicking on the box opens the common Color Picker dialog through which the color can be changed. The **Color** option is also used when objects are rendered as wire frames in RenderZone Plus, whereas the current shader selection determines the display of the object when rendered as a shaded surface. Whether an object is rendered as wire frame, shaded surface or both is determined by selecting the respective attribute in the **Rendering Options** dialog, invoked from the Render Attributes tool, described in section 4.25.3 of the modeling manual.

**Transparency:** As with color, the transparency parameter shows the level of transparency used by rendering types other than RenderZone. If the **RenderZone Average** option is on, the transparency value is computed as an average from the currently selected transparency shader or the **Glass, Simple, Glass, Accurate**, or **Generic** reflection shaders. If the **Custom** option is selected, the transparency parameters can be edited manually.

Note that the Interactive Shaded rendering mode is not affected by the **Color** option. These rendering modes always try to approximate the actual surface style of the object by taking as much information from the surface style as possible.
The shaders dialogs

Next to each of the four menus of shaders there is an Options... button, which invokes a dialog for the shader currently selected in the menu. In this dialog the parameters of the shader can be set. Each shader has a distinct dialog; however, all dialogs share some common characteristics.

Three types of dialogs are invoked. The Plain Color shaders found in the Color and Transparency menus invoke the Color Picker dialog, from which a new color can be selected. The texture map shaders, namely Color Map, Transparency Map and Bump Map, invoke a texture map dialog which includes a preview of the texture map and a command for loading a new texture. Each of these dialogs also contains its own additional buttons which are discussed in the respective sections.

The dialogs invoked by the procedural shaders contain a two dimensional preview of the texture they generate at their upper right corner. Under the preview box there is a slider bar which allows you to control the frequency at which the shader pattern is viewed in the preview window. Dragging the slider to the right increases the frequency of the rendered pattern, while dragging to the left decreases it. Certain procedural shaders require more than one iteration in order to see the complete pattern. This slider allows control of how much of the shader pattern is visible.

The preview image is regenerated each time a parameter is changed. The following five types of parameters are commonly used in these dialogs. Additional types of parameters are used in some of the dialogs and are discussed in the respective sections.

Scale: This parameter contains a scaling factor whose default value is 100%. When the scaling factor is 100%, then the size of the texture unit retains the dimensions specified in the Texture Map Controls dialog. When a factor other than 100% is used, the unit is scaled up or down accordingly. The practical implication of the unit sizing parameters is that, if the unit is scaled up, a larger texture will be generated, and if it is scaled down, a smaller texture will be generated. Using shaders from different categories with different scales allows you to superimpose patterns at different sizes. For example, a Brick pattern at 100% and a Rough bump pattern at 50% may be combined on a single surface.

Texture size: These parameters, found in dialogs of wrapped textures, are used to define the size of the texture module. They are expressed as percentages, and they are relative to the size of the tile. For example, the brick texture may be defined by Brick Width = 90, Brick Height = 40, and Mortar Thickness = 10, where all numbers are percentages relative to the tile size.

Note that, for wrapped textures, when one or more of the pattern sizing parameters is greater than 100, such as Brick Width = 150, the texture tile will be scaled up before it is applied. That is, the tile will be enlarged as much as necessary to fit the texture module that is used to create the overall texture. Consequently, using pattern sizing values greater than 100 is another method for sizing the texture tile.
**Colors:** Most of the shaders include the definition of one or more colors that are used to generate a texture. The currently set colors are displayed in Color boxes. They can be changed by clicking in the color box, which invokes the Color Picker.

**Slider parameters:** These are parameters which are set by sliding a knob on a bar. The value corresponding to the position of the slider knob is displayed in a text field next to the bar. The text field is also editable, and values can be typed directly into it. The slider bar typically corresponds to values ranging from 0 to 100%, which is generally the normal range of the respective parameter. However, values greater than 100 or less than 0 can be typed into the text field, and frequently produce interesting results. The slider bars are used for a variety of parameters, whose degree or frequency can vary along a scale, such as softness or roughness, detail, contrast, etc.

**Noise:** This is a pop up menu found in the shaders that generate patterns based on random algorithms. It adds noise at three different levels. Simple noise generates fast results, but may reveal an underlying rectangular grid along which the random pattern is aligned. The Better option still uses the rectangular grid, but it will not be as visible in the random pattern as with the Simple option. The Best option does not use a grid at all, but takes the longest to compute.

The quality of the random pattern when using Best can be further improved by selecting a higher value for the # Of Impulses, which is an option next to the Noise menu. Reasonable values lie in the range of 3 - 10, with higher values causing additional decrease in rendering speed. Figure 6.2.2.3 shows examples of noise at all three levels generated for the Mist shader.
**Point and area sampling:** An **Area Sample** option is available for all wrapped shaders in the Color, Transparency, and Bump categories. When it is off, one sample is calculated to represent the color value of the pixel, which is called **point sampling**. When a pattern becomes very small, point sampling leads to moiré pattern artifacts, as shown in Figure 6.2.2.4(a). When area sampling is on, an average value is calculated for the pixel; it represents how much area the pixel covers on the rendered surface. This eliminates the moiré artifacts (Figure 6.2.2.4(b)), but causes longer rendering times. Note that the averaging technique is only applied in the **Z-Buffer** or **Raytrace** rendering modes, or if the **Antialiasing** option is also turned on in the **RenderZone Options** dialog.

All shader dialogs contain the same four button commands, namely **OK**, **Cancel**, **Defaults**, and **Reset**. A few contain additional buttons which are discussed in their respective sections.

**Figure 6.2.2.4:** Rendering with **Area Sample** (a) off and (b) on.
### 6.2.3 Color shaders

The **Color** shader menu in the **Surface Style Parameters** dialog contains five groups of shaders: *plain color, shaders that are neither wrapped nor solid, wrapped (2D) procedural textures, solid (3D) procedural textures*, and *precaptured image based textures*. The **Color** shader menu is shown in Figure 6.2.3.1.

#### Plain color shader

**Plain Color**: This item is the default and, when selected, a plain color will be used for the surface style. To select another color, click in the **Color** box, which invokes the Color Picker dialog from which a color can be selected.

<table>
<thead>
<tr>
<th>Plain Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
</tr>
<tr>
<td>Streaks</td>
</tr>
<tr>
<td>Surface Evaluation</td>
</tr>
<tr>
<td>Brick, Paver</td>
</tr>
<tr>
<td>Brick, Simple</td>
</tr>
<tr>
<td>Brick, Textured</td>
</tr>
<tr>
<td>Checker</td>
</tr>
<tr>
<td>Fabric</td>
</tr>
<tr>
<td>Gradient</td>
</tr>
<tr>
<td>Grid</td>
</tr>
<tr>
<td>Paint Splat</td>
</tr>
<tr>
<td>Polka Dots</td>
</tr>
<tr>
<td>Roof Tiles</td>
</tr>
<tr>
<td>Sine Wave</td>
</tr>
<tr>
<td>Stripes</td>
</tr>
<tr>
<td>Wood Boards</td>
</tr>
<tr>
<td>Checker</td>
</tr>
<tr>
<td>Granite</td>
</tr>
<tr>
<td>Marble, Simple</td>
</tr>
<tr>
<td>Marble, Textured</td>
</tr>
<tr>
<td>Mist</td>
</tr>
<tr>
<td>Paving</td>
</tr>
<tr>
<td>Polka Dots</td>
</tr>
<tr>
<td>Wood, Accurate</td>
</tr>
<tr>
<td>Wood, Simple</td>
</tr>
</tbody>
</table>

*Figure 6.2.3.1*: The **Color** shader menu.
The background color shader

The second item in the Color shader menu is a special shader called Background. When selected the Options... button is dimmed and there are no parameters that need to be set. This shader assigns the color of the background at a given pixel, to a surface. It is useful to create scenes, where the background needs to overlap and interact with real 3D objects. An example of how this is done is illustrated in Figure 6.2.3.2.

The background image in (a) shows an urban scene. The 3D model, which outlines a building to be added between two existing buildings, needs to be rendered so that the existing building in front of it is not covered by the new structure. First, the view parameters of the new structure need to match the perspective view parameters of the background image. This can be achieved effectively using the Match Perspective View tool discussed in subsection 2.2.8 of the form•Z manual. Next, the new structure needs to be rendered in such a manner that part of it covers the background while another part is covered by the background.

If the scene is rendered as is, the 3D model will overlap both adjacent buildings (b). To bring one building forward, a massing model, which corresponds to that building, is generated and placed in the proper location. If this massing model is rendered with a simple color, it obscures parts of the new building (c). To do it right, the Background color shader is applied to the massing model. Now all the pixels of that model allow the background to show through it, which produces the desired result (d).

In order to keep the background pixels unaltered when they are displayed on a surface, the Constant reflection shader should be used. However, it is also possible to assign other reflection shaders, such as Mirror, which allows other objects in the scene to be reflected on the surfaces with the Background shader.

Figure 6.2.3.2: Using Background to blend a new object into a scene.
Streaks

This shader simulates simple chrome-like streaks generated from a combination of the Color defined in its dialog (Figure 6.2.3.3) and gray.

This pattern is derived from the angle of a surface’s normal and the view line. Consequently, orthographic projections of flat surfaces do not generate any streaks, because the normal and the view line are parallel. The streaks texture works best on curved surfaces and with perspective views.

This is a directional texture, and its coordinate system can be positioned with the use of the Texture Map tool.

Mix: This slider controls the mix of the color defined in the Color box and gray. 0% results in gray only, and 100° results in color only and no gray, which causes the streaks to disappear.

Figure 6.2.3.3: The Streaks Options dialog.

Figure 6.2.3.4: Streaks textures generated on 16' spheres.
Surface Evaluation

Also known as Zebra, this shader simulates reflections on a surface that result from an imaginary cylinder that surrounds the surface and contains stripes that run parallel to the cylinder axis. It is intended to assist in evaluating whether complex smooth surfaces, such as nurbs used for automotive design, contain surface irregularities. For example, if a smooth looking surface contains a sharp unintended crease, a regular rendering may not show the imperfection. However, the Surface Evaluation shader will highlight that crease, because the reflected stripes will bend at a sharp angle at the crease. This is illustrated in Figure 6.2.3.5, which shows a car body part (a) without imperfection and (b) with a small crease. The surface evaluation shader will also highlight areas of a surface where the curvature changes from convex to concave and vice versa. In that case, the density of the stripes will increase at the area of transition. This is illustrated in Figure 6.2.3.6, where the same body part contains a circular dent.

Figure 6.2.3.5: A car body part rendered (a) without imperfections and (b) with a small crease.

Figure 6.2.3.6: A car body part rendered (a) without imperfections and (b) with a concave dent.
The **Surface Evaluation Options** dialog, containing the shader parameters, is shown in Figure 6.2.3.7.

The standard parameters **Color** and **Area Sample** are described in detail in section 6.2.2. The parameters unique to the **Surface Evaluation** shader are as follows:

**Axis**: This menu offers three choices for the axis of the cylinder: X, Y and Z. They are the axes of the texture map control attribute associated with the rendered object. For example, if X is selected, the cylinder axis is defined as the X axis of the texture map control assigned to the object. If the object does not use a texture map control attribute, the default texture map control, as defined in the **Project Rendering Options** dialog, is used.

**# Of Stripes**: The number of stripes around the cylinder

**Width**: The width of each stripe, relative to the amount of background shown.

**Fuzz**: This parameter controls the smoothness of the stripe edges. A value of 0% creates crisp edges. 100% fuzz causes blurred edges.

Note that for the reflected stripes to show as smooth as possible, the **Quality** slider in the **Geometry** tab of the **RenderZone Options** dialog should be moved to its highest position. This is described in more detail in section 6.1.7.
Colors based on wrapped procedural textures

**Brick, Paver:** This shader allows the user to choose between several different patterns, to use different colors and parameters. There are set in the **Brick, Paver Options** dialog shown in Figure 6.2.3.8.

**Patterns:** A number of patterns can be selected from from this pop up menu: Herringbone, Pinwheel, Boxed Basketweave, Double Basketweave, Single Basketweave, Diamonds, Hexagon, Running Bond, Stacked Bond, Combination Running/Stacked Bond, Squares, and Octagon.

**Dimensions:** This group of parameters controls the sizes of the pattern, as follows:

**Brick Size:** This parameter controls the brick layout of certain patterns. It is dimmed when it cannot be applied.

**Mortar Thickness:** This parameter controls the amount of space between bricks.

**Mortar Fuzz:** This parameter controls the antialiasing between the edge of a brick and mortar.

**Colors:** The colors shown in the three boxes are used to generate the paver pattern. They can be redefined in the usual way (double click on them to invoke the color wheel).

**Variation:** This group of options controls how details of a pattern vary.

**Color Variation:** This parameter controls the variation in the tone of different bricks.

**Noise Scale:** This parameter controls the noise pattern that mixes the two brick colors.

**Rough Scale:** This parameter controls the amount of the variations on the brick edges.

**Rough Amplitude:** This parameter controls the size range of the variations.

**Noise:** This pop up menu controls the noise pattern that mixes the two brick colors.

Examples of patterns that can be generated with this shader are shown in Figures 6.2.3.9 and 6.2.3.10.
Figure 6.2.3.9: Brick, Paver textures displayed on 8' x 8' squares.

<table>
<thead>
<tr>
<th>Scale</th>
<th>Pattern</th>
<th>Brick Size</th>
<th>Mortar Thickness</th>
<th>Mortar Fuzz</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>Basket Weave, Box</td>
<td>50</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>300</td>
<td>Hexagon</td>
<td>50</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>300</td>
<td>Octagon</td>
<td>50</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>300</td>
<td>Stacked Bond</td>
<td>50</td>
<td>25</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 6.2.3.10: Patterns that can be generated with the Brick, Paver shader.
**Brick, Simple**: This shader generates a simple brick pattern, in which all bricks have the same color. Its dialog and parameters are shown in Figure 6.2.3.11.

**Brick Length, Brick Height**: These parameters represent the horizontal and vertical dimensions of a brick, expressed as percentages of the texture tile.

**Mortar Thickness**: This parameter controls the width of the mortar, expressed as a percentage of the tile. Note that if the tile is not square, the vertical and horizontal mortar lines will have different widths.

**Offset**: This parameter determines how the vertical mortar joints line up. At 50% they are centered relative to the bricks on the row above them. At 0% or 100% they are aligned with the vertical joints on the row above them.

**Fuzz**: This parameter controls the smoothness of the edges in the brick pattern. A value of 0% creates crisp edges. 100% fuzz causes blurred edges.

![Simple Brick Options dialog](image)

**Figure 6.2.3.11**: The *Simple Brick Options* dialog.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Brick Length</td>
<td>90</td>
<td>90</td>
<td>30</td>
<td>200</td>
</tr>
<tr>
<td>Brick Height</td>
<td>40</td>
<td>40</td>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td>Mortar Thickness</td>
<td>10</td>
<td>25</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Offset</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Fuzz</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

![Simple Brick textures](image)

**Figure 6.2.3.12**: *Simple Brick* textures displayed on 4' x 4' squares.
Brick, Textured: This shader generates a more detailed brick texture. The side of each brick (stretcher) has a base color, randomly assigned from two colors. The bricks are painted with the base color and are textured randomly using the second color variation. The front of each brick (header) is painted with two different colors allowing for varying patterns of stretchers and headers (Figure 6.2.3.13).

Bonds: These options specify a brick pattern.

Layout: Three brick styles can be selected from this pop up menu: Stretchers Only, Bands, and Alternate (Figure 6.2.3.14). The Bands and Alternate bond styles generate additional bricks, whose headers are visible. The dimension of the brick ends is determined by the value in the Brick Depth field. The color of the brick headers can be different from the color of their stretchers and is determined by the additional two color fields.

Stretchers Only: This bond style only generates rows of stretchers, that is bricks whose sides are visible (Figure 6.2.3.15(1a)).

Bands: This style consists of alternating rows of stretchers (bricks whose sides are visible) and rows of bricks whose headers are visible (Figure 6.2.3.15(1b)).

Alternate: This layout generates rows of bricks that alternate between their stretchers and their headers being visible (Figure 6.2.3.15(1c)).

Offset: This option sets the horizontal offset between two rows of bricks. At 50%, the vertical joint of one stretcher brick in the Stretchers Only and Bands layout is located in the center of the stretchers above. At 100% or 0%, the vertical joints are aligned. In the Alternate layout, the offset determines the horizontal offset between the header bricks of two different rows.

# Of Stretchers: The number entered in this field determines how many times the stretchers will be repeated. When used with Bands, complete rows are repeated (Figure 6.2.3.15(2b)). When used with Alternate, it affects the number of stretchers arranged horizontally (Figure 6.2.3.15(2c)). This parameter has no effect on the Stretchers Only style (Figure 6.2.3.15(2a)).
**Dimensions**: This group of options determines the sizes of the bricks and mortar joints through the following parameters: **Scale**, **Length**, **Height**, **Depth**, and **Mortar**.

**Colors**: This group of options consists of color boxes, which can be changed by clicking on them to invoke the standard color picker dialog.

**Stretcher 1, Stretcher 2**: These colors are used to paint the stretchers of bricks. The first is the base color; the second is used to apply a random texture on the first.

**Header 1, Header 2**: These colors are used to paint the headers of bricks, whenever they are included in the layout. The first and second colors are used as they are for the stretchers.

**Variation**: These parameters determine the shape of the edges of the bricks.

**Rough Scale, Rough Amplitude**: These slider parameters control the size and the depth of the roughness of the edges between the bricks and the mortar.

**Fuzz**: As for **Brick, Simple**.

**Figure 6.2.3.16**: Textured Brick textures for Stretchers Only layout.
**Checker**: This shader generates a checker board pattern from two colors, as shown in its dialog, Figure 6.2.3.17.

*Figure 6.2.3.17*: The **Checker Options** dialog.

*Figure 6.2.3.18*: Wrapped **Checker** textures.
**Fabric**: This shader simulates a woven pattern. It can be used to render a wood lattice fence, a wood basket, or cloth. The parameters in the **Fabric Options** dialog (Figure 6.2.3.19) control color, dimensions of the threads, and background. They are graphically shown in Figure 6.2.3.20.

- **Thread Color**: The base color of the threads.
- **Shaded Color**: The color with which the thread is interpolated towards the edges of the threads.
- **Outline Color**: The color of the border lines.
- **Background Color**: The color of any area not covered by thread.
- **Line Width**: The width of the outline along threads. It can be set to 0 if no outline is desired.
- **Thread Width**: The width of thread, not including the outlines.
- **Anti-Aliasing**: This sliding bar controls how much the outline is interpolated into the background and into the thread color.
- **Thread Shading**: This sliding bar controls the distance over which the thread color is interpolated with the shaded thread color. Can be set to 0 if no shading is desired. No shading will give the appearance of a flat object.
- **Thread Separation**: This controls the distance over which the thread color is interpolated with the shaded color, as the one thread goes under another thread.
- **Angle**: The value in this field determines the rotation angle of the fabric.
- **Use Noise**: When this option is on, noise is applied to the thread pattern.
- **Noise Amount**: This sliding bar determines how much noise will be applied.

Examples of the **Fabric** shader are shown in Figure 6.2.3.21. Note that this shader and the examples also include **Transparency** and **Bump** parts, which are discussed in sections 6.2.5 and 6.2.6, respectively.
Following are some suggestions for achieving better results:

If the direction of the threads follows the shape of the surface, build the surface and map the texture using **Parametric** as the **Mapping Type**.

If the surface style uses a transparent color, set the **Background Color** to the same color as the edge of the thread. If there is an outline thickness, set the background to the same color of the outline color. If there is no outline, set the background to the same color as the shaded thread. If the threads have no shading, set the background to the same color as the thread. This will prevent the shader from forming a “halo” as the threads fade from opaque to transparent.

**Figure 6.2.3.21**: Examples of Fabric.
**Gradient**: This wrapped shader creates a pattern by interpolating between colors. By default, it starts with two colors, but additional colors (up to a maximum of 50) can be added and positioned relative to the others. This shader can be used for color, transparency, reflection, bump, and background. Its parameters can be set in the **Gradient Options** dialog shown in Figure 6.2.3.23.

**Blend**: This pop up menu contains three choices for interpolating colors into each other. They are illustrated in Figure 6.2.3.22.

- **None**: With this item selected, no interpolation is applied, which causes an abrupt shift from one color to another.
- **Linear**: This item causes a direct interpolation between two colors.
- **Smooth**: With this item, the interpolation curve is flattened near the ends of the line, which causes the ends of the transition to be less noticeable.

**Color Mixer**: This is a horizontal rectangle, found under the **Blend** menu. It can be used to mix colors, to introduce new and delete colors, interactively.

For each color in the rectangle, a little **diamond** is displayed under it. Above one of the colors, a small **triangle** is also displayed, while the diamond is highlighted in **red**, which signifies that the color is **active**. These handles are used to apply operations, as follows:

- Clicking on a diamond makes that color active and displays a triangle above it. When active, the attributes of a color can be edited numerically using options in the dialog.
- Clicking inside the rectangle generates a new color, which is initially the color shown at the click point. This color is also displayed in the **Color** box. Clicking on the box invokes the Color Wheel, which allows you to redefine the color in the usual manner.
- Clicking and dragging a diamond repositions the respective color relative to the other colors. When two diamonds get close to each other, the one on the right will move down, preventing it from obscuring the other. Each additional close marker will be moved farther down.
- The **left** and **right arrow** buttons, found under the mixer can be used to move the active color to the next color. The sequence is circular and after the last color the selection moves to the first.
- Clicking on the **trash button** delete the active color.

---

**Figure 6.2.3.22**: Blend options: (a) **None**, (b) **Linear**, and (c) **Smooth**.

**Figure 6.2.3.23**: The Gradient Options dialog.
**Position:** The value displayed in this alpha field represents the position of the active color. Changing this value either numerically or by using the slide rule repositions the active color. Or, reversely, when moving the active color the value in the field changes.

**Direction:** This pop up menu controls the shape of the gradient, as shown in Figure 6.2.3.24.

- **Straight:** When this item is selected, which is the default, the gradient follows a linear path. The angle of this path is controlled by the **Angle** option.

- **Radial:** With this item selected, the path of the interpolation runs around a circular path, which causes the colors to radiate from the center of the texture sample.

- **Circular:** With this item on, the gradient interpolates from the center of the sample towards the outside. This option forms concentric rings.

**Angle:** The value entered in this field determines the orientation of the **Straight** and **Radial** gradients.

**Variation:** This group of options allows one to add randomness to the pattern.

- **Noise Scale:** This slide rule controls the size of the noise pattern.

- **Noise Amplitude:** This slide rule controls the intensity of the noise pattern.

**Area Sample, Noise Quality, # Of Impulses:** As for the other shaders.

Examples of **Gradient** textures are shown in Figure 6.2.3.25.
**Grid**: This shader generates a two color grid pattern whose tiles can be square or have different horizontal and vertical dimensions. Its dialog is shown in Figure 6.2.3.26.

*Figure 6.2.3.26*: The *Grid Options* dialog.

*Figure 6.2.3.27*: *Grid* textures.

<table>
<thead>
<tr>
<th>Scale</th>
<th>Width</th>
<th>Height</th>
<th>Grid Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>80</td>
<td>80</td>
<td>20</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>400</td>
<td>20</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>80</td>
<td>20</td>
</tr>
</tbody>
</table>
**Paint Splat**: This shader generates a pattern of one or more layers of random dots on a plain background color. Each layer has its own parameters that allow the user to change the color, size and shape of the dots.

**Background Color**: The plain background color on which the dot layers are placed.

**Fuzz**: This parameter controls the smoothness of the edge between the dots and the background.

**# Of Layers**: From this menu the number of dot layers is chosen. From 1 to 5 layers are supported. The dots of Layer 1 are always rendered first, with the other layers placed on top. The parameters of each layer are shown in a separate tab.

**Layer Scale**: This parameter applies an additional scale to the dots. This allows the user to create layered dots of different sizes.

**Density**: When this parameter is increased, more dots will cover the background. A low density parameter will generate less dots.

**Variation**: When this parameter is set to 0%, all dots have the same size. At 100%, the dots will vary randomly from a small size up to the size they would be at 0%.

**Detail**: At 0% the dots will be perfectly circular. Increasing the detail parameter will result in a rougher dot edge.

**Color 1, Color 2**: The color of the dots is a random mixture of these two colors.

The **Paint Splat** shader may also be used very effectively as a decal. In the example below, the bike helmet is assigned the **Stripe** color shader as its main surface style. A decal surface style is created with the **Paint Splat** shader for the color shader and also for the transparency shader. As a transparency shader, the paint splat pattern acts as a stencil, rendering just the dots, but not the background. With the Decal tool, the paint splat decal surface style is assigned as a decal to the helmet surface. The resulting rendering is shown in the Figure 6.2.3.30.
**Polka Dots:** This shader generates a flat dot pattern based on values entered for the size of and distance between the dots, as well as the smoothness of their edges. Its dialog is shown in Figure 6.2.3.31.

**Separation:** This parameter, expressed as a percentage of the tile size, represents the distance between the centers of the dots.

**Radius:** This parameter represents the radius of the polka dots relative to the tile size.

**Fuzz:** This slider parameter controls the smoothness of the edge between the dots and the background color.

Note that radius values greater than 50 (with separation at 100%) result in overlapping dots. Separation values other than 100% result in scaled tiles.
**Roof Tiles:** This shader generates a pattern of roof shingles or roof tiles. The **Type** menu offers eight different styles: **Flat**, **Scalloped**, **Diamond**, **Hexagonal**, **Classic**, **Espana**, **Spanish**, and **Roma**. Examples of all eight patterns are shown in Figure 6.2.3.34.

The dimensions and colors of the tiles are entered in the **Dimensions** and **Colors** option groups. The **Variation** slider determines how much color variation occurs on each tile. The variation is displayed as a pattern of light and dark patches. The **Variation Scale** slider determines the size of the patches. The **Roof Tiles Options** dialog is shown in Figure 6.2.3.33.

---

**Figure 6.2.3.33:** The **Roof Tiles Options** dialog.

---

**Figure 6.2.3.34:** The 8 **Roof Tiles** patterns (a) **Flat**, (b) **Scalloped**, (c) **Diamond**, (d) **Hexagonal**, (e) **Classic**, (f) **Espana**, (g) **Spanish**, and (h) **Roma**.
**Sine Wave**: This shader generates a sine wave pattern of two colors. Its dialog is shown in Figure 6.2.3.35.

**Amplitude**: Height the sine wave reaches.

**Fuzz**: How much the colors along the edges of the sine wave blur together.

**Figure 6.2.3.35**: The Sine Wave Options dialog.

**Figure 6.2.3.36**: Sine Waves textures.
**Stripes**: This shader generates two color striped patterns of variable width and orientation, using the parameters shown in Figure 6.2.3.37.

  **Width**: This parameter represents the width of the stripe, expressed as a percentage of the tile size. Note that, contrary to other shaders, values greater than 100 entered for this parameter do not scale the texture tile, but the width of the stripe is always trimmed by the edge of the tile.

**Fuzz**: This parameter controls the smoothness of the edges between the stripes. Note that if **Width** is set to 100 or greater and **Fuzz** is set to 0, no stripes will appear.

  **Stripe Angle**: This slider parameter determines the direction of the stripes. The default value is 0°, which corresponds to a position of the knob in the middle of the slider bar and generates horizontal stripes. Positions of the knob to the left of the middle correspond to negative angles, and positions to the right correspond to positive angles. The maximum angle possible is 90° (vertical stripes). Values greater than 90 that are entered into the text field are adjusted by an increment of 90°.

**Horizontal, Vertical**: These buttons can be used to directly set the stripes' angle to 0° (horizontal) or 90° (vertical) orientations.
**Wood Boards**: This shader offers advanced parameters to simulate the irregular patterns generated by cut wood. It creates board patterns in different layouts, which are filled with wood surfaces. In the **Wood Boards Options** dialog (Figure 6.2.3.39), the parameters of the shader are in five groups: **Wood, Rings, Grain, Boards**, and **Grooves**.

**Wood**: These options set the size, color, and irregularity of the concentric wood rings (*Gnarl*).

**Rings**: These options set the color, size, and shape of the rings.

**Grain**: These parameters specify the shape of small dots that produce a grainy appearance.

**Boards**: These parameters determine the size and shape of the wood boards.

**Layout**: The item chosen from this menu determines the layout of the wood boards. The available options are **Square**, **Herring Bone**, **Floor Boards**, and **Ladder**. They are illustrated in Figure 6.2.3.40.

**Length, Width**: Sets the length and width of a wood board relative to the overall pattern.

**Strips, Offset**: Sets the number of repeated boards and their alignment, respectively.

**Variation**: Sets the amount of color variation between individual boards.

**Grooves**: These parameters determine the size and color of the grooves between the boards.

**Preset Wood**: Preset parameters for a number of wood types are chosen from this menu.

---

**Figure 6.2.3.39**: The **Wood Boards Options** dialog.

**Figure 6.2.3.40**: Examples of the Wood Boards shader with different board layouts:
(a) **Square**, (b) **Herring Bone**, (c) **Floor Boards**, and (d) **Ladder**.
Colors based on solid procedural textures

**Checker, Solid:** This solid shader generates a texture similar to that generated by **Checker**, except that it produces a 3D grid of cubes rather than a 2D grid of squares. Its dialog is shown in Figure 6.2.3.41 and examples are shown in Figure 6.2.3.42.

![Checker Options dialog](image)

**Figure 6.2.3.41:** The **Checker, Solid Options** dialog.

![Surface Styles](image)  
Scale  
100  
200  
320  
320

**Figure 6.2.3.42:** **Solid Checker** textures generated on cubes and a sphere.
Granite: This shader generates a solid pattern which resembles the granulated structure of granite. It consists of four different colors. Two of the colors, by default dark brown and black, create a patchwork of veins which are overlaid on a grainy background composed of the other two colors. The pattern of the veins is influenced by the style, selected from the Type menu. Five styles are available: Sierra, Bianco, Diamond, Azalea, and Auburn.

Mineral Colors: Color 2 and Color 3 are used by the veins. Color 1 and Color 4 define the grains.

Variation: When this parameter is set to 0, the granite pattern is composed exclusively of the four colors. As this parameter is increased, lighter and darker colors are mixed into the pattern. The higher the Variation parameter, the more the colors deviate from the original four colors.

Noise: When this parameter is increased, the background colors (Color 1 and Color 4) are further altered through a random noise pattern.

Noise Scale: This parameter determines the size of the noise pattern.

Fragments: The following parameters influence the vein fragments.

Softness: At a low value the fragments of the veins are coarse. As this parameter is increased, the fragments become softer and more broken up.

Detail: This parameter affects how detailed the broken up fragments appear. When the Softness parameter is very high, an increased Detail parameter creates very fine vein fragments.

Size: This parameter determines the size of the background grain fragments (Color 1 and Color 4).

Cracks: When this parameter is increased, the cracks between the background grain fragments become larger.
**Marble, Simple**: This shader generates a marble or stone-like solid texture from a color entered by the user. Its dialog is shown in Figure 6.2.3.45, and examples in Figure 6.2.3.46.

**Detail**: This slider parameter controls the detail or complexity of the marble texture.

**Figure 6.2.3.45**: The Simple Marble Options dialog.

**Figure 6.2.3.46**: Simple Marble textures.
**Marble, Textured:** This solid shader generates a veined, two color marble or stone texture which is more detailed than that generated by **Marble, Simple.**

**Vein Contrast:** This slider parameter controls the contrast between the marble veins and the stone background.

**Grain:** This slider controls the granularity of the marble.

**Grain Scale:** The value entered in this numeric field controls the size of the marble grains.

*Figure 6.2.3.47: The **Textured Marble Options** dialog.*

*Figure 6.2.3.48: **Textured Marble** textures.*
**Mist**: This solid shader generates two color textures that depict small clouds or mist. Its dialog is shown in Figure 6.2.3.49.

**Detail**: This slider controls the detail or complexity of the mist forms. Examples of mist textures are shown in Figure 6.2.3.50.

![Mist Options dialog](image)

![Mist textures](image)

**Polka Dots, Solid**: This solid shader generates polka dot patterns similar to those generated by its wrapped counterpart (*Polka Dots*), except that the dots generated by this shader appear to be carved out of a block of material composed of a three dimensional grid of spheres. Its dialog is shown in Figure 6.2.3.51. For the examples in Figure 6.2.3.52, the same parameters as Figure 6.2.3.32 were used. However, rotations have also been applied to the texture in (c) and (d).

![Polka Dots Options dialog](image)

![Polka Dots textures](image)

**Polka Dots, Solid** textures. In all cases, the origin of the texture is at the center of the cube. Examples (a) and (b) have no rotation applied to the texture, (c) has a 45° rotation about the X and Z axes, and (d) has a 45° rotation about the Y axis applied.
**Paving**: This solid shader creates a random layout of paving stones, separated by mortar. The stone surface and mortar are defined by two colors each, which are randomly mixed to create the effect of natural stone.

**Stones**: This group of parameters controls the size and shape of the stones.

- **Size**: This parameter determines the size of the stones relative to the overall shader pattern.

- **Shape Variation**: This parameter controls the random shape and size of the stones. The higher the value, the more random the stones become. At low values, the stones approach a square grid.

- **Smoothness**: This parameter controls the shape of the edges of the stones. Low values result in straight and high values in round edges.

- **Color 1, Color 2**: These two colors are blended to create the stone color. **Color 1** becomes the base color and **Color 2** is superimposed in the form of small speckles.

- **Color Variation Scale**: The two colors are mixed to form a random pattern. This parameter determines the scale of this pattern.

- **Grain Variation**: The pattern of the two colors can be further randomized to resemble small dots of grain. How much grain is produced is determined by this parameter.

- **Grain Variation Scale**: The size of the extra grain is set by this parameter.

**Mortar**: This group of options controls the shape of the mortar.

- **Size**: This parameter determines the thickness of the mortar joints.

- **Center Color, Edge Color**: These are the colors of the joints in the middle and at the edge.

- **Color Variation**: This parameter determines the randomness, when the edge and center color are blended together.

- **Color Variation Scale**: When the edge and center colors are blended, a random pattern is generated. This parameter determines the size of the pattern.

- **Irregularity**: This parameter further specifies the shape of the joints. High values produce rough and broken up joints while low values result in straight and smooth joints.

Examples of **Paving** textures are shown in Figure 6.2.3.54.

![Paving Options dialog](image)
Figure 6.2.3.54: Paving textures and their parameters.
**Wood, Simple** (Figure 6.2.3.55): This shader generates a simple wood pattern. It is based on a tree trunk centered around an axis, with concentric rings of light and dark wood colors. If a Texture Map Control is assigned to an object or face using this shader, the z axis of the texture coordinate system is used as the wood axis. If no Texture Map Control exists, the world z axis is used.

![Simple Wood textures](image)

**Wood, Accurate**: Similar to **Wood, Simple**, this shader generates a 3-dimensional pattern of wood rings. Additional parameters allow the wood rings to be irregular in shape. Its parameters are the same as the **Wood Boards** shader, except that there are no options for boards and grooves. The **Accurate Wood Options** dialog is shown in Figure 6.2.3.57.

![Accurate Wood options](image)
Color image maps

Selecting the Color Map item from the Color menu invokes the Color Map Options dialog shown in Figure 6.2.3.59. The upper part is identical to the dialog discussed in section 6.1.11 (Figure 6.1.11.1). The lower part includes additional options.

Repetitions: The parameters in this category determine how many times a color map is repeated on an object, how it is centered relative to the origin of the color map coordinate system, and which color is used beyond repeated textures.

Infinite: When this option is selected, the image is repeated infinitely (or as many times as is necessary to cover a surface) in the respective direction. If it is not selected, the number of repetitions can be typed in the Times field under it.

Center: This option is available only when Infinite is off and the Times option is on for the respective orientation (horizontal or vertical). When off (default), the color map texture is generated by starting at the origin and repeating the tiles in the positive horizontal and vertical directions only. When on, the tiles are repeated in both the positive and negative directions.

Recall that the arrangement of the tiles is also affected by the Center options in the Texture Map Controls dialog, which determine whether the center of a tile or the lower left corner of the tile is placed on the texture’s origin when the texture is generated. The effect of this parameter is illustrated in Figure 6.2.3.60, where in (a) the Texture Map Controls dialog’s Center option is on for both the Horizontal and Vertical Tiling. In (b) it is off for both orientations.

When the Center option of the Color Map Options dialog is on, and an even number is entered in the Times field, then half of the tiles will be arranged in the positive and half in the negative direction. If an odd number is entered in the Times field, then one more tile will be arranged in the positive direction than in the negative. For example, if 5 is entered in the Times field, then 3 tiles will be arranged in the positive direction and 2 tiles in the negative.
To perfectly center an image map texture, the **Center** parameter in the **Color Map Options** dialog should be correlated with the **Center** parameter in the **Texture Map Controls** dialog. To perfectly center an even number of tiles, the **Texture Map Controls** dialog **Center** option should be off. To center an odd number of tiles, it should be on. These conditions apply independently for the horizontal and vertical orientations. Examples are shown in Figure 6.2.3.61.

![Texture Map Controls](image)

**Texture Map Control:**
- **Center Horizontal:** No, No, No, Yes
- **Center Vertical:** No, No, No, Yes

**Image Map Options:**
- **Center Horizontal:** No, Yes, No, Yes
- **Center Vertical:** No, Yes, No, Yes

**Times:**
- **Horizontal:** 2, 2, 3, 3
- **Vertical:** 2, 2, 3, 3

*Figure 6.2.3.61:* The effects of the **Center** options for odd and even repetitions.

In all cases, the texture origin is at the center of the surface.

When the **Times** option is used, whether the produced texture will cover a surface completely or partially depends on the size of the texture relative to the size of the surface. The size of the texture initially depends on the values entered in the **Size** parameters of the **Texture Map Controls** dialog. The **Scale** parameter, found in the **Color Map Options** dialog, can also be used to scale the tiles up or down. The default value for this parameter is 100%, which does not resize the tiles.

Whenever the generated texture does not cover a surface completely, two options available in the **Color Map Options** dialog determine how the portion of the surface that is not covered by the texture will be treated.

- **Repeat Color At Image Edges:** When this option is selected, the area of a surface which is not covered by a texture is colored according to the color of the pixels at the edge of the image.

- **Plain Color Beyond Image Edges:** When this option is selected, the area of a surface which is not covered by a texture is filled with the plain color shown in the color box to the right of this option. This color can be changed by clicking in the color box, which invokes the Color Picker dialog.

**Memory Usage:** The exact size of the texture in memory and the amount of memory consumed is shown in this field. Recall that the resolution of a texture map can be rounded or limited to a certain size depending on the option selected in the **Texture Options** dialog which is invoked from the **Preferences** dialog.
**Sampling**: Selecting one of these mutually exclusive options (MIP Map or Summation Table) applies the respective sampling method, when rendering an image map. Note that the same Sampling options are also available in the Transparency Map Options and the Bump Map Options dialogs.

When rendering in Z-Buffer or Raytrace with the Antialiasing option, the area in a texture covered by a pixel on the screen is used to determine the average color value of the pixel. This averaging can be done using either MIP mapping or summation tables. Each method has advantages and disadvantages.

MIP mapping uses less memory, but textures rendered with this technique tend to look more blurred. This is especially true when one direction of a texture occupies a smaller area in the image than the other direction. This can be seen on the top and side of the cube in Figure 6.2.3.62(a). Summation table sampling is the reverse; while it is more accurate, it requires more memory (Figure 6.2.3.62(b)).

The following formulas can be used to determine exactly how much memory is used by each of the sampling methods:

Color/transparency maps, no alpha channel:
- MIP Mapping:
  \[ \text{bytes required} = \text{texture width} \times \text{texture height} \times 3 \times 1.33 \]
- Summation Table:
  \[ \text{bytes required} = \text{texture width} \times \text{texture height} \times 3 \times 4 \]

Transparency maps with alpha channel and bump maps:
- MIP Mapping:
  \[ \text{bytes required} = \text{texture width} \times \text{texture height} \times 1.33 \]
- Summation Table:
  \[ \text{bytes required} = \text{texture width} \times \text{texture height} \times 4 \]

**Smoothness**: The value entered in this field determines how crisp or blurred a texture appears when rendered in the z-buffer or raytrace renderings with antialiasing. 0% smoothness creates crisp textures. Each rendered pixel on the screen is represented by a single pixel on the texture map. This is also called point sampling. 100% smoothness assigns an average color for an area in the texture to a rendered pixel in the image. For values less than 100%, the area from which the average color is derived is decreased, resulting in a crispier texture appearance. For values larger than 100%, this area is increased, resulting in blurred textures. Values larger than 100% also cause slightly longer rendering times. Note that if the Antialiasing option in the RenderZone Options dialog is not selected, textures always render with 0% smoothness.
6.2.4 Reflection/transmission shaders

Shaders in this class determine how light is reflected off a surface. The desired shader is selected from the Reflections pop up menu, shown in Figure 6.2.4.1. This menu contains four groups of items:

• The first group contains a single shader, Matte, which is the default and produces no reflections and no glossy highlights.

• The second group contains seven shaders that produce no reflections but render surfaces that depict shining or glowing materials, such as metal and plastic.

• The third group contains seven shaders that produce true reflections, and transmissions calculated by a ray tracing procedure. When these shaders are applied to objects in a scene, and the scene is rendered using one of the z-buffer methods, then the main image is rendered with the z-buffer algorithm, but the reflections are calculated and rendered through a supplementary ray tracing procedure. That is, a mixed rendering method is used.

• The fourth group contains a single shader, Environment, which produces simulated reflections through the use of an environment cube or sphere, as discussed in section 6.1.8.

All shaders in this class use some combination of the following set of parameters, which are set through a slider bar.

**Ambient Reflection**: This parameter indicates how much of the ambient light intensity is reflected off a surface. Recall that ambient light is distributed uniformly in a scene. Therefore, if the ambient factor is set to a high value, the contrast between areas which are directly illuminated and areas which are in shade decreases. A low ambient factor results in a higher contrast between these areas. In a scene which is rendered based on a radiosity solution, this parameter is ignored. See Chapter 7 for details about radiosity and ambient light.

**Diffuse Reflection**: When light illuminates a surface, a certain percentage is reflected uniformly in all directions. This percentage is set by this parameter. Rough surfaces usually have a high diffuse reflection value, whereas smooth and glossy surfaces have low diffuse reflection values. Note that a high diffuse reflection parameter helps emphasize the color of a surface.

**Specular Reflection**: This parameter determines how much of the light illuminating a surface is reflected at the incoming angle. As it reflects the incoming light directly into the viewer’s eye, specular reflection generates highlights or “hotspots” on curved surfaces. Surfaces with a high specular reflection value, such as polished materials, appear glossy.
A shader that uses the **Specular Reflection** parameter also uses an option called **Calculate Specular Final Gather**. This option is used in a RenderZone rendering that involves the **Final Gather** global illumination method. It is discussed in more detail in section 6.1.6.

**Roughness**: Curved surfaces with specular reflection generate highlights. The size of these hotspots can be controlled with this parameter. If roughness is set to a high value, hotspots become larger, which makes the surface appear rougher, whereas a small roughness value generates small hotspots, which gives the impression of glossy materials.

**Glow**: Shaders which carry a glow parameter cause surfaces to appear brighter than those only using ambient, diffuse, or specular reflection parameters. If ambient, diffuse and specular components are set to 0% and glow is set to 100%, the surface shows the original surface color at every pixel. Other values generate surface illumination, which employs a combination of shaded reflection and surface glow.

**Reflectivity**: Shaders which have been assigned a reflective shader generate surfaces which reflect the surrounding environment. A reflection value of 100% causes a surface to receive all of its color from the surrounding environment, instead of its own color. True reflection effects are always generated using ray tracing with all the rendering methods that support reflections. In general, reflections require additional rendering time. However, they contribute significantly to generating photorealistic images.

**Transmission**: Shaders which carry a transmission parameter generate transparent surfaces, which are calculated by a ray tracing procedure. Transmission causes a ray to pass through a surface, rather than being reflected off a surface. Transmitted rays are bent according to the refraction parameter. Note that transparency shaders also produce transparent surfaces, but no refraction effects. This is discussed in more detail in subsection 6.2.5. Reflection shaders which include transmission and refraction parameters produce more accurate transparencies. When used in combination with the **Alpha Channel** background shader, transparent pixels generated from these reflection shaders always produce a fully opaque alpha value. Only pixels whose transparency was generated by a transparency shader have attenuated alpha values.

**Refraction**: Rays which are transmitted through a surface are bent according to the index of refraction of that surface. This generates effects such as a magnifying glass and a “broken stick in water”. Each transparent material in the real world has a unique index of refraction. A value of 1.0 indicates no refraction. Water’s index of refraction is about 1.5. No real material has a refraction value greater than 2.5. In the shader dialogs which include the **Refraction** parameter, a pop up menu allows you to select refraction values that correspond to commonly used materials.

All the reflection/transmission shaders use some combination of these parameters, as summarized in the following table (Figure 6.2.4.2). In addition to these general reflection/transmission parameters, some shaders in this class have specialized parameters which are particular to the material which is simulated. These are discussed in detail with the respective shaders.
Most of the reflection/transmission parameters are set through a slider bar and are by default applied uniformly across a surface. For example, the Ambient Reflection on a face is the same at the center and the corners of the face. However, it is sometimes also desirable to vary the amount of each parameter across a surface. This can be achieved by selecting a shader other than None from the Map pop up menu (shown in Figure 6.2.4.3), which is associated with each parameter, except for Refraction and Roughness.

The selected shader generates a gray scale pattern. Where the pattern is white, the parameter is used at its original value, as indicated by the value in the respective numeric field. Where the pattern is black, the value for the parameter is set to 0%; that is, it has no effect. Gray values between black and white generate intermediate values for the parameter, based on the amount of gray. The shaders available in the Map pop up menus are the same as the shaders used by the Color, Transparency, and Bump shaders. When clicking in the Options... button next to the pop up menu, the parameters of each shader can be edited, as with the other shaders.
Non-reflective shaders

Matte: This shader generates surfaces with a dull, matte appearance. It is suitable for non-glossy materials, such as brick or fabric (Figure 6.2.4.4).

Brushed: This shader simulates a material with linear or circular microscopic grooves. For example, a metallic surface which was brushed would exhibit such a reflective property (Figure 6.2.4.5).

Linear/Circular: Selecting from these options determines whether the grooves run in a linear or circular fashion.

Contrast: This slider bar at the bottom of the dialog controls the strength of the effect.

The orientation of the linear or circular grooves is controlled by the texture map control associated with the object. Linear grooves follow the horizontal direction of the texture map control. Circular grooves are concentric rings around the center of a single texture file. If an object is covered by multiple texture tiles, the circular pattern will also occur multiple times. Examples of objects rendered using the Brushed shader are shown in Figure 6.2.4.6.
**Catcher**: This reflection shader, like **Constant**, applies no shading to a surface. That is, the original surface color is maintained. However, **Catcher** allows for surfaces to show shadows and raytraced reflections. Its dialog is shown in Figure 6.2.4.7.

**Shadow Intensity**: This parameter determines how strong the shadows are. At 0% no shadows are visible. At 100% the shadows are at maximum contrast.

**Reflectivity**: This parameter is the same as in the other reflection shaders. It sets the amount of mirrored reflections on the surface.

The **Catcher** shader is most useful when blending rendered models with background photographs. How this can be done is illustrated in Figures 6.2.4.8 through 6.2.4.11.

Figure 6.2.4.8 shows the wireframe display of a car with a photographic background of a showroom lobby. The perspective view in this image has been adjusted, so that the 3D view of the model and the position and angle of the camera, with which the photograph was taken, match. This can be achieved with the Match Perspective View tool, described in section 2.2.8 of the **form•Z** User’s Manual. Figure 6.2.4.9 shows the same model, rendered with **RenderZone**. While reflections and shadows can be observed on the body of the car, the floor in the photograph on which the car is placed does not show any reflection or shadows of the modeled car.

In Figure 6.2.4.10, a simple rectangular polygon was added below the car. The surface style assigned to this polygon uses **Background** for color, **Catcher** for reflection, **None** for transparency, and **None** for bump. As described in section 6.2.3 of the User’s Manual, the **Background** color shader assigns the color of the background to a surface, in this case the photographic background. The **Catcher** reflection shader prevents the polygon from being shaded by the active light sources. As a result, it will blend into the background. In addition, the **Catcher** shader will generate shadows and reflections on the polygon. The rendered image, shown in Figure 6.2.4.11 now gives the impression that the modeled car is part of the photograph, since it casts shadows and reflections on the photograph’s floor.
Figure 6.2.4.8: Wireframe display of a car model on a photographic background.

Figure 6.2.4.9: RenderZone display of the same scene.

Figure 6.2.4.10: Wireframe display, after adding a floor polygon.

Figure 6.2.4.11: RenderZone display with the floor polygon using the Background Color and Catcher reflection shaders.
Chrome: This shader, whose dialog is shown in Figure 6.2.4.12, generates surfaces with a chrome-like effect. It does not generate true reflections, and is a variation of the Streaks shader found in the Color menu of shaders (see section 6.2.3).

Chrome Factor: This parameter determines the contrast between the chrome streaks.

Constant: When this shader is used, the effect of all light sources is ignored. The object is rendered with the original, unshaded surface color.

Metal, Simple: This shader generates a specular metallic appearance, with no reflections. It is suitable for most metallic materials such as steel or brass. Its dialog is seen in Figure 6.2.4.13.

Plastic: This shader generates plastic-like highlights. It is suitable for shiny or highly polished materials such as plastic or varnished surfaces. Its dialog is in Figure 6.2.4.14.

Specular Color: This is the color of hotspots generated by specular reflection. The center of a hotspot shows the pure specular color. At the perimeter of a hotspot, the specular color blends with the surface color.
**Woven**: This reflection shader simulates a material with crossing microscopic grooves, such as a woven material. Its dialog is shown in Figure 6.2.4.15.

**Contrast**: This parameter controls the strength of the reflective effect.

**Bias**: This parameter favors one or the other direction of the crossing grooves. At 50% each direction is given the same weight.

As with the **Brushed** reflection shader, the direction of the grooves, as they occur on a rendered surface, is determined by the texture map control associated with an object or face. An object rendered with the Woven shader is shown in Figure 6.2.4.16. Note that this shader does not render any grooves on the surface, but merely creates reflections as if there were such grooves. In order to give the impression of a cloth-like surface, an appropriate bump shader should be used.

**Figure 6.2.4.15**: The **Woven Options** dialog.

**Figure 6.2.4.16**: Three flags rendered with the **Woven** shader.
(a) 50% bias, (b) 100% bias, and (c) 0% bias
Ray traced reflections

The eight reflection shaders in this group use ray tracing to calculate reflection and transmission properties. Glossy, Metal, Accurate, Mirror and Multilayer Paint are suitable for opaque, reflective materials, whereas Frosty Glass, Accurate and Glass, Simple are appropriate for transparent and reflective surfaces. The Generic shader does not simulate a material in particular, but offers all reflection parameters in one dialog. The Glass, Accurate and Metal, Accurate shaders both calculate the reflection of light using the Fresnel filtering method. Fresnel filtering specifies how the specularly reflected fraction of incident light varies with the angle of incidence on a given surface material. This fraction attains a maximum of 1 at surface-grazing incidence, and generally falls with the angle of incidence approaching the surface normal. The Fresnel filtering method models specular reflections in a physically accurate manner, but requires additional computation time.

**Frosty**: This shader produces blurry transparency. It can be used for frosted windows, glasses, or thin materials like curtains that have a blurry transparency. The effect of this shader is subtle, and is very dependent on what is visible through the Frosty object. Instead of blurring the reflections, it blurs the transmitted light. Note that this shader is computationally intensive and significantly slow. It should thus be used cautiously.

In the real world, a surface cannot reflect more light than it receives. For example, it cannot reflect light diffusely 100% and reflect it in a mirror direction 100%. The light being reflected in all the various ways from the surface cannot exceed the sum of 100%. This is a physical law observed by real materials and is known as the conservation of energy. Frosty observes this principle.

Because Frosty is energy conserving, the factors Ambient, Diffuse, Reflectivity, and Transmission cannot all contribute 100% to the final color output. Instead, internally, certain factors are weighted more heavily than others. Frosty is very dependent on its Transmission factor, therefore, when it is set to 100%, the other factors contribute much less to the final result. So, even if you have Reflectivity set to 100% and Transmission is very high (such as 80%), internally, Reflectivity will be scaled down so that Frosty honors the Transmission factor first and then the Reflectivity factor, and will still be energy conserving.

Frosty also exhibits glass-like behavior in the way it reflects light (when Reflectivity is not 0%). When viewing a Frosty surface straight on, it is more transmissive (more transparent). When viewing a surface at “grazing” angles where the surface normal is closer to 90 degrees from the viewing direction, the surface appears more reflective.

The parameters of this shader are set in the Frosty Options dialog, shown in Figure 6.2.4.17. All parameters are the same as in other reflection shaders except for the following:
**Blur Spread Angle**: This value controls how blurry the transmission can be. If there were no blurriness in the transmission (like in glass), the transmitted light rays would all reflect at the same angle (accounting for the Index of Refraction). The blurry transparency effect is achieved by offsetting multiple light rays at this angle from the original transmitted light ray. For a perfect glass-like transmission, the offset angle would be 0. Thus, smaller values yield more glass-like transparency and higher values yield more blurry transparency.

**Blur Quality**: This factor controls how smooth (lack of noise) the blurriness looks. For more speckled surfaces, a lower quality ought to be used. For less noisy surfaces, a higher quality should be used. This factor ranges from 1 to 10. The highest setting sends out 100 rays to sample what’s being blurred which results in better blurring, but also adds significantly to the rendering time.

**Adaptive Sampling**: Turn this on to further refine difficult areas of noise. Those difficult areas of grainy noise may result because of surfaces that are far away from what they’re reflecting, or because of a large Blur Spread Angle.

This option will concentrate sample rays in areas of higher contrast. Also, at very high Blur Quality settings, turn on this option, as this may yield slightly better performance than without.

**Reuse Sample Directions**: This causes Frosty to reuse the sampling directions it computes at one location for all other locations. This saves a little time, but does not produce accurate results. This produces several “bands” in the transmitted light which may or may not be noticeable.

Performance Tips: If you have a low Transmission setting in your Frosty surface, you can most likely get away with less quality, so turn down the Blur Quality to improve performance. Likewise turn on Reuse Sample Directions to improve performance, if the artifacts aren’t noticeable.
Examples

One possible use of the Frosty shader is for frosted windows. Figure 6.2.4.18 shows an example of this, with two different settings for Blur Spread Angle.

Note that how blurry something appears depends not only on the Blur Spread Angle, but also on how far away it is from the Frosty surface. Figure 6.2.4.19 shows an example of increased blurriness the farther away an object gets.

We can get a variety of different frosty effects by combining different amounts of Transmission, Blur Spread Angle, Reflectivity, etc. See Figure 6.2.4.20 for an example.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value a</th>
<th>Value b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Reflection</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Diffuse Reflection</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Specular Reflection</td>
<td>85</td>
<td>85</td>
</tr>
<tr>
<td>Reflectivity</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Transmission</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Roughness</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Refraction</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>Blur Spread Angle</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>Blur Quality</td>
<td>3</td>
<td>ON</td>
</tr>
<tr>
<td>Adaptive Sampling</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>Reuse Sample Directions</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

Figure 6.2.4.18: The effect of the blur angle on the blurriness of a window.
**Figure 6.2.4.19:**
The effect of distance on the blurriness.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Reflection</td>
<td>0</td>
</tr>
<tr>
<td>Diffuse Reflection</td>
<td>100</td>
</tr>
<tr>
<td>Specular Reflection</td>
<td>85</td>
</tr>
<tr>
<td>Reflectivity</td>
<td>5</td>
</tr>
<tr>
<td>Transmission</td>
<td>80</td>
</tr>
<tr>
<td>Roughness</td>
<td>5</td>
</tr>
<tr>
<td>Refraction</td>
<td>1.5</td>
</tr>
</tbody>
</table>

**Figure 6.2.4.20:**
Different effects achieved by the **Frosty** shader.

<table>
<thead>
<tr>
<th>Property</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Reflection</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Diffuse Reflection</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Specular Reflection</td>
<td>85</td>
<td>100</td>
<td>0</td>
<td>85</td>
<td>100</td>
</tr>
<tr>
<td>Reflectivity</td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>50</td>
</tr>
<tr>
<td>Transmission</td>
<td>100</td>
<td>50</td>
<td>69</td>
<td>100</td>
<td>95</td>
</tr>
<tr>
<td>Roughness</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Refraction</td>
<td>1.1</td>
<td>1.0</td>
<td>1.3</td>
<td>1.0</td>
<td>1.5</td>
</tr>
<tr>
<td>Blur Spread Angle</td>
<td>15</td>
<td>20</td>
<td>45</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Blur Quality</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Adaptive Sampling</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>Reuse Sample Direct</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>
**Generic**: This reflection shader combines all the reflection parameters: Ambient, Diffuse, Specular, Reflectivity, Transmission, and Glow. It is not intended to simulate any material in particular. The calculation of the illumination of a pixel simply adds the shading contribution of the six reflection parameters. The **Generic Options** dialog is shown in Figure 6.2.4.21.

**Glass, Accurate**: This shader supports an accurate simulation of glass-like materials with reflection and transmission properties, which are generated using ray tracing and Fresnel filtering. Its dialog is shown in Figure 6.2.4.22.

**Glass, Simple**: This shader produces an approximated simulation of glass-like materials with reflection and transmission properties which are generated using ray tracing but no Fresnel filtering. Its dialog is shown in Figure 6.2.4.23.
**Glossy**: This shader gives surfaces a blurry, mirror-like reflection, as for a glossy kitchen floor. Instead of a perfect mirror reflection, the rays of light are slightly, randomly redirected when hitting the surface. This creates the blurry effect, simulating small imperfections on the surface. The parameters of this shader are set in the **Glossy Options** dialog, shown in Figure 6.2.4.24.

All parameters are the same as in other reflection shaders except for the following two:

**Blur Spread Angle**: This value controls how blurry the reflection can be, which is based on the reflection angle offset. For a perfect reflection, the offset from the reflection ray would be 0, so this parameter controls how far off the original reflection the reflected ray can get. Thus, smaller values yield more mirror-like reflections and higher values yield more blurriness.

**Blur Quality**: This factor controls how smooth the blurriness looks. For more speckled surfaces, a lower quality ought to be used. For smoother surfaces, a higher quality should be used. This factor ranges from 1 to 10. The highest setting sends out 100 rays to sample what is being blurred, which results in better blurring, but also adds significantly to the rendering time.

**Adaptive Sampling**: Turn this on to further refine difficult areas of noise, which may result because of surfaces that are far away from what they are reflecting, or because of a large **Blur Spread Angle**. This option will concentrate sample rays in areas of higher contrast. Also, at very high **Blur Quality** settings, turn on this option, to yield slightly better performance.

**Reuse Sample Directions**: This causes **Glossy** to reuse the sampling directions it computes at one location for all other locations, which saves a little time, but does not produce accurate results. It produces several “bands” of reflections that may or may not be noticeable.

Performance Tips: If you have a low **Reflectivity** setting in your **Glossy** surface, then you can most likely get away with less quality, so turn down **Blur Quality** to improve performance. Likewise turn on **Reuse Sample Directions** to improve performance, if the artifacts are not noticeable.
**Metal, Accurate:** This shader, whose dialog is in Figure 6.2.4.25, accurately simulates metallic surfaces with reflections calculated using ray tracing and Fresnel filtering.

**Refraction, Absorption:** These two parameters are unique to this shader. Each allows you to enter values for **Red**, **Green**, and **Blue**, which determine the color of the area between the center of a specular hotspot and its perimeter, calculated by Fresnel filtering. The refraction values for some common metals can be obtained by selecting the respective material from the **Refraction/Absorption** pop up menu. Values for refraction and absorption for specific materials can be found in handbooks of material constants.

**Mirror:** This shader produces reflective surfaces using ray tracing but no Fresnel filtering. It is appropriate for mirror-like surface finishes. Its dialog is shown in Figure 6.2.4.26.
**Multilayer Paint**: This reflection shader simulates a material that is coated by three layers of paint: a base paint, a metallic flakes coat, and a glossy transparent finish. The base layer uses the color generated by the color shader and adds the basic ambient and diffuse reflection properties. The metallic flakes coat adds colored particles, oriented in different directions, which create mirrored reflections. The finish layer adds the common specular and reflectivity parameters.

This shader’s parameters are set in the **Multilayer Paint Options** dialog (Figure 6.2.4.27), which contains a number of groups of options. The **Base** and **Finish** groups are the standard reflection parameters. **Metallic Flakes** contains parameters specific to this shader. The **Transmission Factor** in the **Finish** group determines the clarity of the finish. Note that this does not create glass-like transmissions, but simply determines how much of the base and metallic flakes layers are visible. For examples, see Figure 6.2.4.28.

---

**Figure 6.2.4.27**: The Multilayer Paint Options dialog.

**Metallic Flakes**: These parameters affect the display of the metallic flakes.

- **Scale**: This parameter determines the size of the flakes as a percentage of the **Solid Texture Size** of the texture map control with which the object is rendered. Recall that this is defaulted in the **Texture Map Options** dialog or can be set with the Texture Map tool.

- **Color**: This parameter determines the color of the flakes

- **Coverage**: This determines how much of the base is covered by the metallic flakes.

- **Roughness, Amplitude, Detail, Sharpness**: These parameters determine the strength of the effect created by the flakes.

- **Reflectivity**: This parameter determines how reflective the flakes are. Like in the **Mirror** shader, these reflections are created through secondary raytracing.
Figure 6.2.4.28: Objects rendered with the Multilayer Paint shader. Each car is shown with its respective Specular Reflection Factor and Metallic Flakes color, both chosen in the Multilayer Paint Options dialog.
Simulated reflections

**Environment**: This shader, whose dialog is shown in Figure 6.2.4.29, simulates reflective surfaces by using environment mapping rather than ray tracing. As discussed in more detail in section 6.1.8, images of the surrounding scene are generated and attached to the six planes of an environment cube or a single image of an environment sphere. These images are then reflected back onto the surfaces to which this shader has been assigned. This method is faster than ray tracing, but less accurate and requires additional memory.

The **Environment** shader can display the pattern of four different environments: **global mapped**, **global rendered**, **local mapped**, and **local rendered**. As is also discussed in subsection 6.1.8, the global mapped and rendered environment is defined in the **RenderZone Options** dialog, in the **Scene** tab, and in the **Environment Reflection Options** dialog, shown in Figure 6.2.4.29.

**Global**: When this option is selected, the global mapped or rendered environment is reflected in a surface, depending on the selection in the **Type** menu below the **Global** radio button.

**Local**: When this option is selected, the environment is defined in the **Environment Reflection Options** dialog itself. This allows for multiple environments to be used in a scene, as several surface styles can each define their own local environment.

The local environment can also be rendered or mapped, depending on the selection in the **Type** menu below the radio button. When **Rendered** is chosen from the **Type** menu, the environment is defined by rendering six images at the resolution specified in the **Rendered** section below. These images show all the objects in the scene seen from a center point looking along the orthogonal positive and negative X, Y and Z directions. The center point is determined by averaging the location of all the objects to which a particular **Environment** shader has been assigned. When **Mapped** is selected from the **Type** menu, the pattern reflected in the surface is determined by a shader. The pattern can be projected in a cubic or spherical fashion, depending on the selection in the **Mapped** menu. The shader can be selected from the **Shader** menu.
One way to use local environments would be to create more accurate reflections when several reflective objects are placed in a scene, which are spatially separated. The example in Figure 6.2.4.30 (a) shows such a scene. There are three reflective spheres. They all use a surface style which has the Environment shader set to use the global rendered environment. They all show the ground plane as the reflected pattern. The same scene is shown in Figure 6.2.4.30 (b), but each sphere uses a separate surface style, each with the Environment shader set to use the local rendered environment. Notice that each sphere shows the ground but also the adjacent spheres as reflections.

*Figure 6.2.4.30:* A scene where each object uses the (a) global rendering environment and (b) its own rendered environment.
Applying reflection and transmission effects with textures

Recall that all the reflection and transmission shaders use some combination of a set of parameters, most of which are set through a slider bar which applies these parameters uniformly across a surface. Whenever it is desirable, a shader other than None can be selected from the Map pop up menu, which is associated with each parameter, except for Refraction and Roughness.

The selected shader generates a gray scale pattern. Where the pattern is white, the parameter is used at its original value, as indicated by the value in the respective numeric field. Where the pattern is black, the value for the parameter is set to 0%; that is, it has no effect. Gray values between black and white generate intermediate values for the parameter, based on the amount of gray.

One frequently used application of reflection maps is a glow map. The example illustrated in Figure 6.2.4.31 shows how a glow map can be used to give the impression of equally bright areas on a surface, regardless of the shading generated by incident light.

(a) For the Color shader, an Color Map is loaded which displays the word “GLOW” on a blue background. The Reflection shader is set to Matte.

(b) Clicking on the Options... button next to the Matte menu item, invokes the Matte Options dialog. In this dialog the Glow: Factor is set to 100%.

(c) From the Map menu, the Glow Map shader is selected. After clicking on the Options... button next to it, the same texture map as used for the Color shader is loaded. However, Use Alpha Channel is also selected. The alpha channel of the texture has a gray scale image of the map, with a black background and white highlights for the word "GLOW". This will apply the glow factor to the green pixels but not to the blue pixels of the color map.

(d) To avoid that, ambient and diffuse reflection parameters are applied to the glowing area, the Ambient Map and Diffuse Map shaders are selected from the Ambient Reflection: Map and Diffuse Reflection: Map menus, respectively. In their Options dialogs, Use Alpha Channel and Invert are selected.

(e) By inverting the gray scale values of the alpha channel, the ambient and diffuse parameters are applied only to the blue pixels, but not to the green pixels of the color map, producing the desirable results.
Figure 6.2.4.31: Rendering with a glow map.
6.2.5 Transparency shaders

Transparency shaders are selected from the Transparency pop up menu, shown in Figure 6.2.5.1. This menu contains six groups.

- The first group contains the default None, which produces no transparency.
- The second group contains four shaders, which, contrary to the transparencies in groups four and five do not have a color equivalent. Center-Edge and Neon produce graduated transparencies based on the angle between the surface normal and the view direction. Plain Color and Simple produce uniform transparencies.
- The third group contains a single shader, Eroded, which uses a solid shader and produces a pattern of irregular transparent and opaque areas that resemble rusted surfaces. It also does not have a color equivalent.
- The fourth group contains wrapped shaders whose pattern can also be generated by a color shader.
- The fifth group contains solid shaders whose pattern can also be generated by a color shader.
- The sixth group contains the single Transparency Map shader, which produces transparent patterns from image files.

While the transparency attributes supported by these shaders resemble the transmission attributes supported by the reflection shaders, the two are also significantly different.

Transparency is generated by two different techniques. The Plain Color and the Transparency Map shaders filter the color of a surface by using another color as a filter. That is, the values of the RGB (red, green, blue) components of the color of a surface are augmented by the inverse RGB values of the filter color multiplied by the background color. For example, assume that we use a filter color of 40% red, 40% green and 40% blue, which is a medium grey. A surface filtered through this color will add 60% of each of the RGB components of the background color to the surface color. In general, this method produces transparent surfaces which are slightly brighter than their opaque counterparts, since the background color is added to the surface color. The other transparency shaders use a simple transparency factor. It determines how much of the background color will be mixed with the surface color of an object. For example a transparency factor of 20% produces a final surface color with 20% of the background color and 80% of the original surface color.
Both methods produce transparent surfaces, but they do not model the change in view direction caused by refraction. To incorporate the latter effect, the Frosty, Generic, Glass, Simple or Glass, Accurate reflection shader should be used. That is, to render transparent objects in a physically accurate manner, one of these four reflection shaders, which contain a transmission parameter, must be used.

Transparency shaders are also useful as “masks.” That is, you can use either a procedural or a precaptured pattern which represents a high contrast shape, typically in black and white. As a surface is filtered with this pattern, the portion that corresponds to the white will be fully visible, while the portion that corresponds to the black is fully transparent and thus completely invisible. This is actually the technique used for rendering realistic trees, as discussed in more detail in section 6.2.9. In general, this method allows you to generate regular or irregular transparent patterns, which would be hard to create using only modeling techniques.

As with the other shaders, a dialog is associated with each of the transparency shaders, which is invoked by selecting the Options... button when the name of the respective shader is displayed in the Transparency menu. To set a specific variation of the shader, these dialogs use scale, texture size, and slider parameters, which are similar to those used by the Color shaders. They all have a preview window, which produces a rendering using the parameters currently set in the dialog. Transparency shaders in group three, four and five display a white surface rendered on a black background. Shaders in the second group, except for the Plain Color shader, render the preview object selected in the Surface Style Parameters dialog.
Independently set transparencies

**None**: When this item is selected, no transparency shader is applied, and the surface to which the surface style is assigned is rendered opaque.

**Simple**: When this shader is selected, a uniform transparency is applied to the surfaces of an object. The level of transparency desired is entered in the Simple Transparency Options dialog shown in Figure 6.2.5.2.

**Transparency**: This slider parameter controls the transparency level, where 0% produces completely opaque objects, and 100% produces completely transparent (invisible) objects. An example is shown in Figure 6.2.5.3.

**Plain Color**: When this item is selected, a uniform transparency is applied to each surface, as with the Simple transparencies. For this shader, the level of transparency is determined by a color assigned to a surface and used as a transparency filter. A new color can be selected by clicking in the Color box to invoke the Color Picker dialog.

The example in Figure 6.2.5.5 illustrates the assignment of different shades of grays as transparency shaders to a scene of primitive geometric shapes. The respective surface styles are displayed in the palette shown in Figure 6.2.5.4. If it is not important to render transparent surfaces with refractions, renderings can be created much more quickly using the Simple or Plain Color transparency shaders, as in the examples on this page, instead of one of the Reflection shaders with a transmission parameter.
Center-Edge: This shader generates a variable transparency based on the angle between the view direction and the normal direction of a surface. If the view direction is parallel to the normal direction of a surface (i.e. the line of sight is perpendicular to the surface), the transparency assigned to this area is determined by the Center Transparency parameter. If the view direction is perpendicular to the surface normal, the transparency is determined by the Edge Transparency parameter.

Figure 6.2.5.7: Center-Edge transparency shaders applied to spheres.

Falloff: One of three options may be selected from this pop up menu to determine how the areas between the Center and Edge Transparency levels will be interpolated. If set to Linear, the transparency changes in a linear fashion from the center transparency to the edge transparency. If set to Square or Cubic, the transparency falls off slower. If the center transparency is set higher than the edge transparency, this shader can provide a simple simulation of glass-like surfaces without the overhead of computing the less transparent areas of glass at the perimeter of curved objects through ray tracing. If the edge transparency parameter is set to a higher value than the center transparency, the edges of curved objects can become “fuzzy.”
**Neon:** This shader can generate glowing effects. It works best with curved objects, such as spheres and tubes. The transparency of a surface is determined by the angle between the surface normal at a given pixel and the view direction. If this angle is 180°, the surface is perpendicular to the view direction, and is rendered with maximum opacity. If this angle is 90°, the surface is parallel to the view direction, and is rendered with maximum transparency. Angles between these two values generate an intermediate level of transparency.

Neon-like glowing effects can be generated with this shader when combined with the **Constant** reflection shader and a bright, plain color. For example, if applied to a sphere, the object appears opaque at its center and gradually becomes more transparent toward its perimeter. The **Constant** reflection shader helps to give the sphere a glowing brightness, since no shading is performed.

**Intensity:** This parameter determines the level of transparency when the view direction is perpendicular to the surface (the angle between the surface normal and the view is 180°). 0% generates maximum transparency, which results in invisible objects. 100% generates completely opaque surfaces at angles of 180°.

**Falloff:** This parameter determines how quickly the transparency diminishes from angles of 180° toward 90°. **Linear** falloff gives an even distribution. With **Square** and **Cubic** falloff, the transparency diminishes more quickly.

*Figure 6.2.5.8: The Neon Options dialog.*

*Figure 6.2.5.9: Neon transparency shaders applied to spheres.*
**Eroded:** This solid texture shader works as a stencil that creates the illusion of erosion on a surface. The size of the texture is controlled by a *Scale* parameter, as shown in the dialog of Figure 6.2.5.10.

![Eroded Options dialog](image)

**Coverage:** This slider parameter controls the coverage of the eroded area. A value of 100% means that the entire surface has been eroded (i.e. it is fully transparent). A value of 0% generates no erosion. Values in between determine how much of a surface is covered by eroded areas.

**Fuzz:** This slider parameter is used to set the fuzziness of the edges of the eroded area.

![Eroded shaders applied to cubes](image)

**Figure 6.2.5.11:** *Eroded* transparency shaders applied to cubes.
Transparency shaders matching the color textures

The fourth and fifth groups in the **Transparency** shader menu contain shaders which are the same as the wrapped and solid **Color** shaders.

Clicking on the **Options...** button invokes a dialog that contains the transparency controls for the shader currently selected in the **Transparency** menu. These dialogs are identical to the dialogs invoked by their corresponding color shaders, except that the **Color** boxes are replaced with slider parameters, which control the level of transparency. In addition, they all have one additional button, **Match Color Shader**.

Two dialogs, the **Grid Transparency Options** and the **Simple Marble Transparency Options** dialogs, are shown in Figures 6.2.5.12 and 6.2.5.13.

**Match Color Shader**: This button command is available in all the dialogs that support transparency shaders which have a corresponding color shader. Clicking on it sets all the parameters to those set in the corresponding **Color** shader dialog, except for the **Transparency** parameters.

![Figure 6.2.5.12: The Grid Transparency Options dialog.](image1)

![Figure 6.2.5.13: The Simple Marble Transparency Options dialog.](image2)

![Figure 6.2.5.14: The Grid Transparency shaders applied to three cubes and a sphere.](image3)
Transparencies from images

Transparency Map: This wrapped texture uses an image file as a transparency filter. When the Options... button is clicked, the Transparency Map Options dialog shown in Figure 6.2.5.15 is invoked. This dialog is very similar to the Color Map Options dialog (Figure 6.2.3.59), but contains three additional options.

**Use Alpha Channel:** When this option is selected, the transparency is determined by the gray value of the alpha channel of the image. The alpha channel is available when images are captured using a 32 bit representation. It can be thought of as a second layer of the image representation, which carries information that complements the image. That is, while three of the bytes of a 4 byte representation are used for the color information of an image, the fourth byte is used for the alpha channel, each pixel of which can take 256 different values. The alpha channel is typically used to store shades of gray which are used to determine transparency levels or bump amplitudes (see next section).

**Invert:** When this option is selected, the gray scale values of the alpha channel are inverted.

**Match Color Shader:** This button is available only when the Color Map shader has been used to assign a precaptured image as Color to an object. It is dimmed and inactive otherwise. Clicking on it copies all the values used in the Color Map Options dialog (including the selection of an image map) into the Transparency Map Options dialog. This feature facilitates the correlated positioning of texture maps when the same image is used for Color and for Transparency.
**Make Only Black Areas Transparent:** When this option is off (default), surfaces will be made transparent according to the percentages of red, green, and blue they contain for each of their pixels. 100% for all the RGB colors, which is white, makes the surface completely opaque, and 0%, which is black, makes the surface completely transparent and thus invisible.

When this option is on, all areas of a surface which correspond to a black pixel in a texture map become 100% transparent, whereas pixels of any other color become 100% opaque. This option is very useful in creating texture maps with a transparent background. For example, simulated trees can be rendered this way, as discussed in section 6.2.9. Examples of how a simple gray pattern affects the transparency of an object are shown in Figure 6.2.5.16.

![Figure 6.2.5.16: Applying texture maps as transparency filters: Black (1) background and (2) pattern.](image)

- (a) The 2D pattern.
- (b) The pattern applied as color and no transparencies.
- (c) Applied as a transparency filter with **Make Only Black Areas Transparent** off.
- (d) Applied as a transparency filter with **Make Only Black Areas Transparent** on.
6.2.6 Bump shaders

A **bump** is a three dimensional effect which is added to a surface through rendering. While most of the bumps can also be created more permanently through modeling, generating them through rendering is more efficient, primarily due to significant memory savings.

Bump shaders are selected from the **Bump** pop up menu, shown in Figure 6.2.6.1, which contains six groups of shaders.

- The first group contains the default **None**, which produces no bumps.
- The second group contains six bump shaders, which are based on wrapped (flat) textures. These shaders do not correspond to other color shaders.
- The third group contains three solid texture based shaders, which have no corresponding color shaders.
- The fourth group contains wrapped shaders whose pattern can also be generated by a color shader.
- The fifth group contains solid shaders whose pattern can also be generated by a color shader.
- The sixth group contains the single **Bump Map** shader, which produces bumps from image files.

The following parameters are used by most of the bump shaders:

**Scale:** This parameter determines the size of the texture unit and works as for the **Color** shaders. A value of 100 preserves the unit size defined in the **Texture Map Options** dialog. Values less and greater than 100 scale the unit size down and up, respectively.

**Amplitude:** This slider parameter controls the three dimensionality of the bumps. Positive values produce raised bumps, while negative values produce indented bumps.

**Blend:** This slider parameter controls the transition from a bump to the surface it is on. Small values produce sharper transitions, and higher values produce smoother transitions.
Bumps based on wrapped procedural textures

**Cells**: This bump shader creates a pattern of bumpy cells. Initially the cells are aligned in a square grid. Through a number of randomizing parameters, the cell structure can be made irregular, resembling the pattern of leather and skin-like surfaces. The parameters are **Rough**, **Curve**, and **Fold**. Rough superimposes a dense pattern of random bumps over the cells. Curve adds waviness to the cell edges and Fold adds a large bump pattern. The **Cells Options** dialog is shown in Figure 6.2.6.2.

**Cell Amplitude**: This parameter determines the height of the cell.

**Irregularity**: Increasing this parameter disturbs the grid-like layout of the cells.

**Smooth Min, Smooth Max**: These parameters determine the size of the smooth edges of the cells at their bottom and top seams, respectively.

**Rough/Curve/Fold**: Each of these categories contains three parameters which determine the magnitude and shape of the respective disturbance: **Amplitude** is the height/magnitude of the random disturbance, **Detail** is the amount of detail added to the disturbance, and **Frequency** is the size of the random disturbances. Examples of **Cell** patterns and the parameters to generate them are shown in Figure 6.2.6.3.

![Figure 6.2.6.2: The Cells Options dialog.](image)

![Figure 6.2.6.3: Bump patterns generated with the Cells shader.](image)
**Dimple**: This wrapped texture shader generates a regular dimple pattern, which can be thought of as a grid of spheres protruding from a surface. Its parameters, shown in Figure 6.2.6.4, resemble those of the **Polka Dots**.

**Separation**: This parameter represents the distance between the centers of adjacent spheres, expressed as a percentage of the texture tile size.

**Radius**: This parameter represents the radius of the dimple sphere, as a percentage of the texture tile size.

**Center Depth**: This is the distance of the center of the sphere from the surface, expressed as a percentage of the tile size. Positive values generate raised dimples. Negative values generate indented dimples.

Examples of dimple bumps are shown in Figure 6.2.6.5, where the bumps have been applied on a plain color, and in Figure 6.2.6.6, where the dimple bumps have been coupled with polka dot and grid textures applied as **Color** shaders.

![Figure 6.2.6.4: The Dimple Options dialog.](image)

![Figure 6.2.6.5: Dimple bumps generated on plain color 16' x 16' cubes. Texture tile size: (a), (b), and (d) 4' x 4'; (c) 2' x 2'.](image)

![Figure 6.2.6.6: Dimple bumps generated on 16' x 16' with (a), (b) wrapped polka dots, and (c), (d) wrapped grid textures applied as Color shaders. All tile sizes are 4' x 4' except in (d) where the grid texture size is 8' x 8'.](image)
**Knurl**: This wrapped texture shader generates a regular knurled pattern consisting of pyramid-shaped bumps. Its dialog and parameters are shown in Figure 6.2.6.7. Examples are shown in Figure 6.2.6.8.

![Knurl Options dialog](image)

**Figure 6.2.6.7**: The **Knurl Options** dialog.

<table>
<thead>
<tr>
<th>Scale</th>
<th>Blend</th>
<th>Amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>10%</td>
<td>100%</td>
</tr>
</tbody>
</table>

**Figure 6.2.6.8**: Wrapped **Knurl** bumps generated with tile size 4' x 4'.

**Rough**: This wrapped texture shader generates an irregular, rough surface. Its dialog and parameters are shown in Figure 6.2.6.9. Examples are shown in Figure 6.2.6.10.

**Detail**: This slider parameter controls the complexity of the bumps.

**Sharpness**: This slider controls the sharpness of the bumps. A low value produces abrupt changes between the peaks and valleys of the bumps, while a high value generates smooth transitions.

![Rough Options dialog](image)

**Figure 6.2.6.9**: The **Rough Options** dialog.

<table>
<thead>
<tr>
<th>Scale</th>
<th>Amplitude</th>
<th>Detail</th>
<th>Sharpness</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>10%</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 6.2.6.10**: Wrapped **Rough** bumps generated with tile size 1' x 1'.

![Surface Styles](image)
Shingles: This wrapped shader generates a bump pattern that simulates roof and wall shingles. The pattern is formed from individual tiles of a certain shape, which are separated by grooves. The parameters of this shader are set in the Shingles Options dialog shown in Figure 6.2.6.11.

Type: Four shingle patterns are available from this menu: Rectangular, Double Castellated, Tripple Sawtooth, and Tripple Curved Slot Tooth. These patterns are shown in Figure 6.2.6.12.

Tiles: This group of options controls the shape and size of the shingles pattern.

Width, Height, Thickness: These parameters define the size of a shingle, relative to the overall pattern size. Note that the tile is assumed to be tilted. Therefore, Thickness determines the size of the shingle’s bottom edge.

Angle: This parameter determines how much a single is tilted off the surface. The larger the angle, the more pronounce the bottom edge of the shingle will be rendered.

Vertical Offset Variation: This value controls how shingles are lined up horizontally. At 0% all shingles are aligned straight. At higher values the shingles are randomly offset vertically.

Overlap Area: This value determines how much one shingle overlaps the next, below it.

Row Offset: This parameter determines the horizontal offset between two rows.

Side Chamfer, Bottom Chamfer: These parameters define the amount of chamfer (beveling) at the side and bottom edges of the shingle.

Grain: This group of options controls the shape and size of the grain of the shingles.

Amplitude: This parameter determines the height of the wood grain of the shingles.

Horizontal Scale, Vertical Scale: These two parameters define the size of the grain.

Threshold: This parameter defines the sharpness of the grain. At a low value, the grain appears smoother. At a high value, the grain becomes sharper.
Tread Plate: This wrapped texture shader generates a wrapped tread plate pattern, which consists of cylinders with rounded ends that protrude above the surface. Its dialog and parameters are shown in Figure 6.2.6.13. Examples are shown in Figure 6.2.6.14.

**Figure 6.2.6.12:** Shingles bump textures applied to 2' x 2' tiles.

**Figure 6.2.6.13:** The Tread Plate Options dialog.

**Figure 6.2.6.14:** Tread Plate bump textures applied to 16' x 16' cubes. Cylindrical mapping with number of Tiles: horizontal 8 and vertical 2.
Bumps based on solid procedural textures

**Casting**: This shader generates a cast metal-like bump pattern. It is composed of surface bumps and superimposed dented bumps. It can be used for any surface requiring irregular roughness. Its options are shown in Figure 6.2.6.15, and examples in Figure 6.2.6.16.

**Casting Amplitude, Dented Amplitude**: These amplitude sliders control the height of the surface bumps and the height of the superimposed dented bumps, respectively.

**Dented Scale**: This scale factor may be used to resize the dented bumps.

**Dented Threshold**: This slider controls the ratio of dented bumps relative to raised bumps. A value of 0% covers the surface entirely with surface bumps, whereas a value of 100% covers the surface entirely with dented bumps.

**Detail**: This slider controls the complexity of the dented bumps.

**Cells, Solid**: This shader is the same as the **Cells** shader, except that the pattern is based on a 3 dimensional grid of cubes. See Figure 6.2.6.2.

**Rough, Solid**: This shader is the solid counterpart of the **Rough** shader, and uses the same dialog and parameters, except for the **Area Sample** option. Examples are shown in Figure 6.2.6.17.
Bumps matching the color textures

The fourth and fifth groups of shaders in the Bump shaders menu contain shaders which are the same as the wrapped and solid Color shaders, respectively. This allows you to add bumps that match the pattern generated by the color shaders.

Clicking on the Options... button invokes the dialog that contains the bump controls for the shader currently selected in the Bump menu. These dialogs are identical to the dialogs invoked by their corresponding color shaders, except that the Color boxes are replaced with slider parameters which control the amplitude of the bumps. In addition, they all have one extra button. Two dialogs, the Simple Brick Bump Options and the Simple Marble Bump Options dialogs are shown in Figures 6.2.6.18 and 6.2.6.19.

Match Color Shader: This button command is available in all the dialogs that support bump shaders which have a corresponding color shader. Clicking on it sets all the parameters to those set in the corresponding color shader dialog, except for the amplitude parameters.

Even though these bump shaders are intended to complement the color shaders, this is not a necessary condition, and they can also be used without a matching color shader. For example, a plain color may be used as color, and then a bump brick texture may be applied to it resulting in the effect produced when brick walls are painted.
Bumps from images

Bumps can be generated from image files. The bumps are determined by the relative change of color brightness in the image. That is, areas in an image which change from dark to light will cause a raised bump at the light portion, and an indented bump at the dark portion.

**Bump Map:** This wrapped texture generates the bumps of a surface from an image file.

Clicking on the **Options...** button invokes the **Bump Map Options** dialog, shown in Figure 6.2.6.20.

**Amplitude:** This slider parameter is used to control the height and depth of the bumps. Examples are shown in Figures 6.2.6.21 and 6.2.6.22.

**Figure 6.2.6.20:** The **Bump Map Options** dialog.

**Figure 6.2.6.21:** Using a precaptured leather image as **Color** and/or **Bump** texture. Cubic mapping on 16' x 16' cube with texture tile size 24' x 24'.

**Figure 6.2.6.22:** Using prerendered texture on a 16' x 16' cube, as (a), (c) **Color** only; (b), (d) **Color** and **Bump**. Cubic mapping on a 16' x 16' cube, tile size 16' x 16', texture origin at (a), (b) center of cube and (c), (d) corner point.
6.2.7 Copying surface styles and invoking predefined surface styles

Two button commands available at the lower right end of the Surface Style Parameters dialog allow you to copy the parameters of one surface style into another, and to pick the parameters from a prestored surface style.

**Copy From...**: When this button is selected, a dialog is invoked which displays the palette with the list of surface styles (Figure 6.2.7.1). Any surface style displayed can be selected by clicking on it. When OK is selected to close the dialog, the parameters of the highlighted surface style are copied into the surface style which is active in the Surface Styles palette.

Note that this operation is different from the Copy... operation available in the Surface Styles dialog. While the latter duplicates the active surface style and creates a new item in the Surface Styles palette, this Copy From... operation copies the parameters of a selected surface style into the surface style active in the Surface Styles palette, which is an existing surface style.

**Predefined...**: Clicking on this button invokes the Predefined Material dialog, shown in Figure 6.2.7.2.

A *predefined material* is a surface style that has already been defined, has been assigned a name, has been grouped into a category of materials, and has been saved into a file from which it can be retrieved. These predefined surface styles represent commonly used materials. Included with form•Z RenderZone Plus is a folder called form•Z Materials (located in the form•Z application folder), which contains a number of material files. Each file contains several predefined surface styles, which become available to the user when invoking the Predefined Material dialog.

The Predefined Material dialog consists of a Material display window at its top portion, a Surface Style preview window at its left portion, and a number of button commands that allow you to select the material file whose content will be displayed, to start a new material category, to add new or delete existing materials, and to save a material category to a different file. Using these commands, you can build your own customized material categories.
**Category**: This pop up menu is constructed from the names of the files in the form•Z Materials folder. When this folder contains only the files shipped with the program, this menu will appear as shown in Figure 6.2.7.3. Any new material category that you may establish will be appended to this menu. Selecting one of its items displays its content in the Material window.

**New...**: This button creates a new material file, which is a new category. The **New Category Name** dialog (Figure 6.2.7.4) is invoked where the name of the new material file is entered. This file is initially empty, but material definitions (surface styles) can be saved into it as soon as it is created.

**Sort**: Clicking on this button sorts the materials in the active category alphabetically. The active category is that whose name is selected in the Category menu.

**Save A Copy As...**: This button invokes the standard Save dialog, where a file name and directory for a copy of the current material category can be selected. After the Save dialog is closed, the Material File Version dialog (Figure 6.2.7.5) is invoked. From the pop up menu you can select the version to which the material category will be saved.

**Selected Material**: This non-editable text field displays the name of the material currently selected in the Material window.

**Delete...**: This button deletes the highlighted material from the active category. Since this operation cannot be undone, the alert shown in Figure 6.2.7.6 is posted to verify the intentions of the user.

**Rename...**: This button invokes the **New Material Name** dialog (Figure 6.2.7.7) in which a new name for the material highlighted in the Material display window may be entered.

**Save As Material...**: This button invokes the **New Material Name** dialog (Figure 6.2.7.7) where a name is entered. When the dialog is closed, a new material is created with the parameters of the surface style shown in the Surface Style preview window and stored into the file of the active material category.

**Apply Selected Material**: Clicking on this button closes the Predefined Material dialog and copies the parameters of the highlighted material into the surface style edited in the Surface Style Parameters dialog (the surface style active in the Surface Styles palette).

Note that changes made to a material category, such as deleting a material or sorting, cannot be undone. All changes to a category are saved into the corresponding material file when the Predefined Material dialog is closed, even if the **Cancel** button is pressed.
6.2.8 LWA predefined materials

As discussed in the previous section, the Predefined Materials dialog is invoked from the Predefined... button found in the Surface Styles dialog. A variety of predefined materials can be selected from the Predefined Materials dialog.

A predefined material file may be either a form•Z material, identified by the .zmt extension, or a LightWorks Archive (LWA) file, which has the .lwa file extension. A LWA file may contain rendering materials which are equivalent to a form•Z surface style. It may also contain other rendering related entities, such as lights or geometry. However, in the context of the Predefined Materials dialog, only rendering materials are extracted from these files. While a form•Z material file can only be created by form•Z, LWA files may have been issued by a number of different sources. LightWorks Design has created a number of LWA files, which contain a large number of sample materials. These sample files can be downloaded from the www.lightworks-user.com website. LWA files have also been issued from other sources, such as paint suppliers or flooring manufacturers. This allows industries to showcase their products as rendering materials, which enables users of form•Z to preview these products in a realistic form•Z rendering. For example, a form•Z user may pick a rendering material from a LWA file which specifically describes a particular surface by its color pattern and reflective properties. In a subsequent rendering, an object displayed with that material will appear very close to that material.

When importing a LWA material, it needs to be translated to a form•Z surface style. To do this, the shaders that make up the LWA material are matched as closely as possible to the shaders that a form•Z surface style offers. For example, a LWA material may use the LightWorks “brick” shader. When used as a form•Z surface style, the form•Z Simple Brick shader is used in the surface style, and the parameters in that shader are taken from the equivalent parameters in the LWA shader. Note that, although in general a LWA shader and the equivalent form•Z shader can be matched one to one, there may be instances where no exact match of parameters can be found. In this case, form•Z will attempt to provide a solution that is as close as possible to the intended rendered appearance of the LWA material. There may also be a situation where a LWA material contains a shader that does not have an equivalent shader in form•Z. In this case, form•Z will substitute a default shader, such as the Plain color shader and an appropriate warning message will be issued to the user.
Legal limits of LWA files

A LWA file may contain certain privacy and copy protection settings. A copy protected LWA file requires that a user verifies the validity of the LWA file through a password scheme. This is currently not supported by form-Z. In other words, only those LWA that can be freely distributed are handled as material files in the Predefined Materials dialog. In addition to copy protection, a LWA file may contain materials which have shaders where one or more parameters are protected. For example, a material may have a reflection shader, where the diffuse reflection parameter is secret. That is, the form-Z user may not know the content of the parameter. When a shader options dialog with such a parameter is opened, the content of the parameter is hidden from the user. LWA files which contain materials with protected shader parameters are supported by form-Z. When such a material is chosen in the Predefined Materials dialog and copied to the current surface style, the protection of the shader parameters is maintained. When the Shader Options dialog is invoked, secret parameters are displayed with their protection intact.

In Figure 6.2.8.1 below, the Plastic Options dialog is shown with the Specular Reflection parameter in secret mode. The content of the parameter is completely removed from the dialog and the word “Secret” is shown in its place.

In addition to tagging individual shader parameters as secret, a LWA material may also lock one or more of the four shaders. In this case, the user cannot change the shader from one type to another. If such a material is chosen from the Predefined Materials dialog, the locked shaders appear dimmed in the Surface Style Parameters dialog, as shown in Figure 6.2.8.2. The user may still open the respective shader options dialog, but cannot access the Shader menu to select a different type.
Textures in LWA materials

A material in a LWA file may use shaders which reference texture map files, such as the Color Map color shader. The texture map file is searched for in the following places and in the order shown:

1. In the folder or nested folders where the LWA file is located.
2. In the Textures folder of the form•Z LightWorks folder of the form•Z application directory.
3. In any of the file search paths, as currently shown in the Preferences dialog.

A shader of a LWA material may also contain a texture which is imbedded directly in the LWA file. When the material is chosen from the Predefined Materials dialog, the texture is extracted and written as a TIFF file to the folder where the LWA file is located. The name of the texture is constructed from the material name, as shown in the Predefined Materials dialog, plus a string which identifies it as a color, reflection, transparency or bump map. For example, the color texture from a material called “marble_style1” would be named “marble_style1_col.tif”. If the texture map of the shader is tagged as a secret parameter, the image is extracted to a private image format that carries the .fzi extension. This image file cannot be opened by the user for viewing or editing. It can only be used as the image shader in a surface style.

LWA files in the Predefined Materials dialog

The Category menu shows all the predefined materials that form•Z finds in the search paths, which have the Predefined Materials option selected. Recall that search paths are edited in the Preferences dialog, invoked from the Edit menu of the main menu bar. If the currently chosen material file is a form•Z material, the Sort, Save A Copy As..., Delete and Rename actions are available as described earlier in this section. If the current material in the Category menu is a LWA file, these actions are disabled, since a LWA file cannot be modified by the user. A preview of all the materials in the LWA file is shown in the graphic area on top of the dialog, however, the preview image stored in the LWA file is shown instead of the standard rendered preview of a form•Z surface style (Figure 6.2.8.3). Note that it is not guaranteed that the LWA preview is identical to a subsequent rendering of a form•Z object using that material, since the content of the preview image depends entirely on the author of the LWA file.

In the Category section of the Predefined Materials dialog, a text area displays information about the currently selected material category. In the case of a LWA file, this information may list restrictions of the file, such as that it is copy protected. In this case, the LWA will still show in the Category menu but, when selected, it will show no materials in the preview area and will display the copy protected message in the text info area (Figure 6.2.8.4).

Apply Significant Shaders Only: This option is displayed only when a LWA file is selected in the Category menu and is on by default. When selected, and the chosen LWA material is copied to the current surface style, only those shaders of the LWA material that are tagged as significant are copied. A significant shader is one that gives the material its essential characteristics. For example, a LWA material that represents a paint color may have the Color and Reflection shaders tagged as significant, but not the Bump shader. When the user applies this material to the current surface style, the Bump shader is not altered. If this option is off, all shaders of the LWA material are copied.
Once a LWA material is applied to a surface style, the **Surface Style Parameters** dialog shows a button named **LWA Info**. When pressed, the full path and name of the original LWA material is displayed as shown in Figure 6.2.8.5.

**Figure 6.2.8.3:** The **Predefined Materials** dialog with a LWA file selected.

**Figure 6.2.8.4:** The **Predefined Materials** dialog with a password protected LWA file selected.

**Figure 6.2.8.5:** The **LWA Info** dialog.

**Saving form•Z files with shaders that have read only or secret parameters**

After a LWA material has been included in the Surface Styles palette of a *form•Z* project, the secret status of shader parameters is maintained throughout the life of the *form•Z* project. Any export of surface style information, be it as a backsave to a previous version of *form•Z*, where the privacy of shader parameters does not exist, or to a generic rendering file format, such as Wavefront (obj), will cause the protected shader parameters to be exported with default values rather than the actual values.
6.2.9 Rendering trees

form•Z RenderZone Plus can complement your rendering with images of trees and plant materials, which are not models in the scene, but are produced with texture mapping techniques. These effects are created using precaptured images of trees, which may have been produced photographically or by hand painting them, as follows:

- Generate two single segment extrusion objects, positioned perpendicular to each other, as shown in Figure 6.2.9.1. Normally, these surface objects should be square.
- Define a surface style where the same tree image map is assigned both as a Color shader and as a Transparency shader. The assignment is executed through the Color Map Options (Figure 6.2.9.2) and Transparency Map Options dialogs, respectively. Recall that these dialogs are identical except that the latter has one additional check box labeled Make Only Black Areas Transparent. Assuming that your tree image has been captured with a black background (as are those included with the RenderZone Plus package), turn this option on. You may also select Constant as a Reflection shader. This is not absolutely necessary but will help the image preserve its original colors.
- Use the Texture Map tool to set the proper mapping parameters (see next section), as follows:
  - Use the Flat mapping method.
  - Rotate the texture axes so that the XY plane sits on the surface object (with X horizontal and Y vertical).
  - Position the texture’s origin on the lower left point of your surface object.
  - In both Tiles fields, for number of tiles, enter 1.

Repeat the texture mapping process for both surface objects, and make certain that the tree image is mapped on both in consistent directions. If in doubt, use the RenderZone display command to preview them while still in the Texture Map Controls dialog.

To double check that your tree images have been correctly mapped onto your surface objects, you may render the image using the Z-Buffer mode, with the Transparencies option off. Your rendering should be as shown in Figure 6.2.9.3(a).
Next, you can render your final image. Use either z-buffer or raytrace to render the image and turn the **Transparencies** option on. Both rendering methods produce satisfactory tree images. A final rendering of the tree is shown in Figure 6.2.9.3(b).

Texture mapped trees can also be generated using a single surface object onto which the tree image is mapped, but using two perpendicular surfaces makes the image more three dimensional. Note that when tree images are mapped onto rectangular surfaces and opaque shadows are turned on, the result is not satisfactory, since rectangular shadows will be cast. If proper shadows are required, one of two methods may be employed to produce satisfactory results.

An irregular surface object can be drawn that roughly corresponds to the shape of the tree. This can be done by importing a tree image as an underlay, and tracing over it in the usual manner. This object can then be copied so that two surface objects can be positioned perpendicular to each other. Such simulated tree models will produce satisfactory results when shadows are also rendered. Alternatively, the **Transparent** shadows options, set in the **RenderZone Options** dialog can be used in conjunction with raytraced shadows. In this case, it is not necessary to create an outline shape following the tree contour. However, rendering times will increase since additional computations are required to determine which pixels are in shadow.

The tree used in the example above is a birch tree whose image was derived photographically. **form•Z RenderZone Plus** is shipped with the eight precaptured tree images shown in Figure 6.2.9.4. Additional tree images can be produced by the user.

---

**Figure 6.2.9.3:** Tree images mapped onto two surface objects and rendered (a) without and (b) with transparencies.

**Figure 6.2.9.4:** Precaptured tree images shipped with **form•Z RenderZone Plus**.
6.3 Texture map control

All form•Z objects are assigned the active surface at the time they are generated. When rendered, the size of the textures in a surface style is determined by a default texture map control setting. Its parameters are displayed in the Default Texture Map Control tab of the Project Rendering Options dialog as shown in Figure 6.3.1.1. This dialog also contains other project rendering options, which is described in further detail in section 3.6.6 of the form•Z User’s Manual. Recall, that the content of this dialog is context sensitive. When invoked from the Display menu, it shows all options. When invoked from a particular rendering options dialog, only those options which apply to the rendering mode are available. The Default Texture Map Control tab is shown only when the Project Rendering Options dialog is invoked from the RenderZone Options or Interactive Shaded Options dialog or directly from the Display menu.

Certain texture mapping methods work well for some shapes, but not for others. Because of this, it is frequently desirable to individually map textures on objects. This is done by assigning a custom Texture Map attribute to individual objects with the Texture Map tool and the Texture Map Controls dialog, that it invokes. In other words, unless an object has been specifically assigned a Texture Map attribute through this tool, when rendered, the size, mapping type and orientation of a texture on an object is determined by the project’s default texture map control settings.
6.3.1 The Default Texture Map Control tab

The options in the **Default Texture Map Control** tab of the **Project Rendering Options** dialog are the same as those in the **Texture Map Options** dialog, described in section 6.3.2. They are listed briefly here and are explained in more detail in that section.

**Origin**: The values entered in the *X*, *Y* and *Z* fields represent the position of the origin of the texture coordinate system. Unlike in the **Texture Map Options** dialog, there is no **Center** option.

**Rotation**: The values entered in the *X*, *Y* and *Z* fields are in degrees and represent the angles the texture coordinate system will be rotated relative to its axes. These rotations are always executed in *Z*, *Y*, *X* order.

**Reset**: When this button is pressed, all three angles are reset to 0 degrees.

The fields in the **Wrapped Textures** section determine the mapping type and size of wrapped (2D) textures. Since a wrapped texture is two dimensional in nature, but is rendered on a three dimensional object, it must be projected (mapped) from 2D to 3D. This is done via a mapping type and a size specification.

**Mapping Type**: One of four basic mapping types can be selected from this menu: **Flat**, **Cubic**, **Cylindrical** and **Spherical**. The effects of these mapping types are explained in more detail in section 6.3.4.

**Flip**: When this option is selected, the vertical and horizontal directions of the 2D texture are reversed.

**Horizontal/Vertical Tiling Size**: The size of the each dimension of the texture is specified in these fields. For **Flat** and **Cubic** mapping the size is expressed as a linear distance. For **Cylindrical** mapping, the horizontal direction is an angular value and the vertical size is linear. For **Spherical** mapping both values are angular. Section 6.3.3 provides more detail and examples about the mapping size fields. Unlike in the **Texture Map Options** dialog, there are no **Tiles** fields.
**Center**: When this option is selected, the texture tile is centered in the horizontal and/or vertical direction, relative to its origin. When off, the lower left point of the texture is placed on the origin of the texture coordinate system.

**Mirror**: With this option, the texture is mirrored in the horizontal and/or vertical direction.

**Parametric Mapping For Parametrics**: When this option is on and the object rendered is a parametric object, the **Parametric** mapping type with the **Tiling** parameters below this option are applied. Parametric objects which are affected by this option are: analytic primitives (cylinder, cone, sphere, torus), nurbs objects and c-meshes. When this option is off, the mapping type selected in the **Mapping Type** menu above is applied to a parametric object.

**Lock Size To**: When **None** is chosen from this menu, the **Horizontal/Vertical** tile sizes can be set independently. If **Square** is selected, typing a new value for the horizontal size automatically sets the vertical value to be the same, and vice versa. When **Current Proportions** is selected, the ratio of the horizontal to vertical size is maintained when entering a new value.

**Solid Textures Size**: The value entered in this field determines at what size solid (3D) textures are rendered.
6.3.2 The Texture Map tool

When the default texture mapping parameters are not satisfactory, a different mapping method can be assigned, and the mapping parameters can be manipulated using the Texture Map tool.

**Texture Map**

This tool is used to map both procedural and pre-captured textures onto the surfaces of objects. It can be used with either the prepick or postpick method.

When using the postpick method, with the Texture Map tool active, click on the object whose texture you wish to map. The postpick method always picks at the Object level, regardless of the topological level currently selected. To use the prepick method, with the Pick tool active and the topological level set to Object, preselect any number of objects. Then activate the Texture Map tool and click in the graphics window.

Associated with the Texture Map tool are two dialogs, which are to a large extent similar:

- The **Texture Map Options** dialog is invoked directly from the tool, by double clicking on it (Figure 6.3.2.1).
- The **Texture Map Controls** dialog is invoked automatically when the Texture Map tool is applied and the **Edit** option in the **Texture Map Options** dialog is on (Figure 6.3.2.2).

Note that the Texture Map Options dialog is virtually identical to the Default Texture Map Control tab of the Project Rendering Options dialog. Instead of containing the default parameters for objects without a texture map attribute, this dialog contains the attribute setting which will be assigned to an object the next time the Texture Map tool is used. It also has a few extra options which are described below.

**Adjust To New**: When this option is on, the settings in the **Texture Map Options** dialog are applied to the object selected with the Texture Map tool, regardless of whether the operation enters the edit mode and invokes the **Texture Map Controls** dialog (see next option). When off, which is the default, the settings in the **Texture Map Options** dialog have no effect on objects which already have their own texture map parameters.
**Edit:** When this option is on it controls whether the **Texture Map Controls** dialog will be invoked when executing the Texture Map operation. It is invoked when this option is on (default). When this option is off, the settings selected in the **Texture Map Options** dialog are applied to the object, if the **Adjust To New** option is also selected.

Texture mapping involves the following processes, each of which corresponds to a portion of the dialog that contains the respective parameters and controls:

- The texture is moved and/or rotated to position it and orient it relative to an object or a face.
- The size of the tile or texture module is defined.
- One of the available mapping methods is selected.
- Groups of faces may be defined and the mapping process repeated for each of these groups.

These processes are discussed in detail in the next subsections.

![Texture Map Controls dialog](image)

*Figure 6.3.2.2: The **Texture Map Controls** dialog.*
6.3.3 Previewing and positioning textures

Most textures, particularly wrapped textures, have directions. For example, a brick pattern represents the way bricks are laid out in rows, from the ground up. Positioning a brick texture sideways makes little sense. Detailed mapping of textures is frequently required in order to position the textures in a manner consistent with the image they represent. As the position of the texture is manipulated, it can be viewed in the preview window located in the upper left corner of the Texture Map Controls dialog (Figure 6.3.3.1). This preview window works in a fashion similar to that of other preview windows in form•Z.

On invoking the Texture Map Controls dialog, the preview window displays the object on which textures are to be mapped and the current position of the texture coordinate system. Depending on the mapping type currently selected (see section 6.3.4), a flat plane, a cube, a cylinder, or a sphere representing the mapping surface is also displayed. These shapes bound the object and have their center at the origin of the texture’s coordinate system.

As Figure 6.3.3.2 illustrates, these bounding shapes can be significantly different in size, depending on where the texture’s origin is placed. They are the smallest when the Object Center option is on and the texture’s origin is placed at the center of the object.

**Figure 6.3.3.2:** The texture mapping bounding shapes as displayed in the preview window:
- Texture origin at (1) center of object and (2) origin of world space.
- Texture mapping type: (a) flat, (b) cubic, (c) cylindrical, and (d) spherical.
The images in the preview window are color coded:

• The bounding shapes and two of the axes of the texture coordinate system are shown in the default light green color used for the reference plane grid lines in the main form•Z windows.

• The third axis is shown in the color used for the world axes in the main form•Z windows and is red by default. This is the axis about which rotations can be executed, and corresponds to the axis currently selected in the Axis/Plane pop up menu (see below).

• The faces of the active texture group (see section 6.3.7) are drawn in black. Faces of other texture groups are drawn in the project’s ghost color, which is gray by default.

The three check boxes under the preview window determine what is displayed in it, and how the display is refreshed:

**Show Object**: When this option is on (default) the object is drawn. If off, the object is not drawn and its faces cannot be selected to define texture groups (see section 6.3.7).

**Show Mapping Type**: When this option is on (default), the texture coordinate system and the bounding shape that represents the mapping type are drawn.

**Draw Tiles**: When this option is selected and an wire frame is displayed, the texture tiles are also displayed.

**Snap**: When this option is on and you move the mouse after you have initiated a move origin or a rotate axes operation (see below), the mouse snaps to points, midpoints, and segments of the object. You can temporarily cancel the snapping by pressing the **shift** key. Likewise, if **Snap** is off, you can temporarily activate it by pressing the **shift** key.

The view in the preview window can be manipulated using tools from the preview tool palette located under the preview window. All these tools are the same as those in other preview windows. In addition when the tool is selected, the texture coordinate system can be moved and rotated, as discussed in more detail below. It is also used to select faces for the current texture group.

The **Texture Map Controls** dialog offers both numeric and graphic methods for positioning textures onto objects. For both methods, two basic operations are available: **moving** the origin of the texture coordinate system, and **rotating** the texture coordinate system about one of its axes.
Numeric input

The text fields and buttons to set the position of the texture coordinate system are located in the upper right corner of the Texture Map Controls dialog (Figure 6.3.3.3).

**Origin**: The values entered in the $X$, $Y$, and $Z$ fields represent the position in the world coordinate system to which the origin of the texture coordinate system will be moved. This is true for all mapping types except parametric and UV coordinates. For parametric the position of the origin is expressed as a percentage of the surface where a texture will be mapped, along the $X$ and $Y$ direction. For parametric, the $Z$ parameter is dimmed and inactive. For UV coordinates these fields do not apply and are disabled.

**Center**: Clicking on this button moves the origin of the texture to the center of the object. The coordinates of the center are also displayed in the $X$, $Y$, and $Z$ fields. Clicking on this button while *option* (Macintosh) or *ctrl+shift* (Windows) is pressed, moves the origin of the texture to the average coordinates of all the faces which are assigned to the currently active texture group and highlighted in the preview window. This button is not available for parametric and UV coordinates mapping.

While **Center** is a button in the Texture Map Controls dialog, it is a check box in the Texture Map Options dialog. Selecting it (which is the default) causes the texture’s origin to be initially placed on the object’s center, when the Texture Map tool is used.

**Rotation**: The values entered in the $X$, $Y$, and $Z$ fields are in degrees and represent the angle the texture coordinate system will be rotated relative to its axes. These rotations are always executed in the $Z$, $Y$, $X$ order. These parameters are not available for parametric and UV coordinates mapping.

**Reset**: Clicking on this button cancels all the rotations, and resets the values in the $X$, $Y$, and $Z$ fields to $0^\circ$. Note that this button does not affect any other operations that may have been applied to the texture, either numerically or graphically.
Graphic input

When the position of the texture coordinate system is manipulated through numeric input, the image in the preview window of the **Texture Map Controls** dialog is adjusted to reflect the changes made. The position of the texture can also be manipulated graphically, by moving and rotating its coordinate system directly in the preview window.

When the Arrow tool (🔑) from the preview tool palette is selected and the cursor is moved inside the preview window, its icon changes depending on which part of the texture coordinate system or the object it is closest to.

- The cursor changes to this icon when it is on or close to the **origin** of the texture coordinate system. It signifies that the origin can be moved. You move it by clicking the mouse once, dragging the cursor to the desired position. Clicking the mouse a second time finishes the operation.

- The cursor changes to this icon when it is on or close to an **axis** that can be rotated. Which axes can be rotated depends on the current axis of rotation, which is selected from the **Axis/Plane** pop up menu located under the preview window. The axes which are not the current axis of rotation can be rotated. For example, if **Z/XY** is selected, then X and Y can be rotated about the Z axis. In the preview window, the axis of rotation is displayed in red, and the other axes in light green. An axis is rotated about the axis of rotation by clicking the mouse button once and dragging it to a new position. Clicking the mouse a second time finishes the rotation.

- The cursor changes to this icon when you press the button to execute either a move origin or a rotate axis operation. It returns to its previous icon as soon as the button is released to complete the operation.

- As the mouse cursor is moved either to move the origin or to rotate an axis, if **Snap** is on, it will snap to the points, segments, or segment mid-points of the object. If **Snap** is off, it will move freely, relative to the plane selected in the **Axis/Plane** pop up menu. For example, if **Z/XY** is selected, the move will be relative (parallel) to the XY plane. Both states of the **Snap** can be temporarily reversed by pressing the **shift** key. When the mouse snaps, the cursor changes to this icon.

A short cut method for positioning the X and Y axes of a texture onto the surface of a face is also available. It is applied by pressing **option** (Macintosh) or **ctrl+shift** (Windows) when a face is selected to make it a member of a texture group. This method is discussed in detail in section 6.3.6.
6.3.4 Texture mapping methods

Wrapped textures can be mapped onto objects using one of six available methods. These types are selected from the Mapping Type pop up menu located on the middle right of the Texture Map Controls dialog (Figure 6.3.4.1). The fifth method (parametric) can be used only with parametric objects, such as spheres, cylinders, cones, tori, nurbz objects and c-meshes. The sixth (UV Coordinates) can be applied only after one of the other methods has been applied and is derived from them. Note that the texture mapping methods apply only to wrapped textures, and are ignored by the solid textures.

**Flat**: This type of mapping uses a single placement of the texture which can be on any plane in 3D space. By default, this texture is initially placed on the XY plane of the Cartesian coordinate system. This position can be subsequently changed using the positioning tools available in the Texture Map Controls dialog.

A texture has its own coordinate system whose X and Y axes lie on the plane of the texture. A texture is generated by first placing the lower left point of a properly scaled tile onto the origin of its coordinate system, then repeating the tile horizontally and vertically until an area of texture that covers the entire object is generated. This texture is then projected onto the faces of the object in a direction parallel to the Z axis of its coordinate system. In other words, the texture on the faces of the object results from the intersection points of the projection lines with the planes of the faces.

The flat method of mapping a brick texture is illustrated in Figure 6.3.4.2. Two 3D forms are used: a complete cube and a cube with one of its corners cut out. The texture is positioned parallel to the XY plane (default) and it is projected in a direction parallel to the Z axis. Note that it appears normal on the horizontal faces of the object, but it is distorted as it is projected onto the side faces, which are perpendicular to the plane of the texture.

The two objects are shown to the left as they are displayed in the preview window of the Texture Map Controls dialog, when the Tiled display option is used. To the right, the same objects are shown rendered in full z-buffer mode. This format will be used with all the examples in this section. The first object used will have the form which most closely matches the type of mapping being illustrated. The cube with the cut out corner will be used as the second object for all mapping types except for parametric and UV coordinates.

---

**Figure 6.3.4.1**: The Mapping Type menu.

**Figure 6.3.4.2**: **Flat** texture mapping.
**Cubic**: This type maps textures by using a cube which bounds the object. The texture is generated on each of the faces of the cube, and is projected in a direction perpendicular to the face of the cube on which it was generated. Each face of the object receives its texture from that face of the bounding cube to which it is most parallel. Cubic mapping is illustrated in Figure 6.3.4.3.

![Cubic texture mapping](image)

*Figure 6.3.4.3: Cubic texture mapping.*

**Cylindrical**: This type maps the textures using a cylinder which bounds the object. The texture is generated on the round surface of the cylinder, and is projected towards its central axis. In other words, the projection lines converge onto the axis of the cylinder. Cylindrical mapping is illustrated in Figure 6.3.4.4.

![Cylindrical texture mapping](image)

*Figure 6.3.4.4: Cylindrical texture mapping.*
**Spherical:** This type maps textures using a sphere which bounds the object. The sphere’s origin is at its center, and its pole is on the Z axis of the texture coordinate system. The texture is generated on the surface of the sphere starting with its largest diameter, which corresponds to the horizontal direction. As the lines in the Y direction converge to the poles of the sphere, the texture is scaled as rows are laid out. The spherical texture is projected towards the center of the sphere. That is, all projection lines converge onto the center of the sphere. Spherical mapping is shown in Figure 6.3.4.5.

*Figure 6.3.4.5: Spherical* texture mapping.
**Parametric**: This mapping technique can be applied only to parametric objects. The horizontal or length direction of the parametric object corresponds to the horizontal direction of the texture tile, and the vertical corresponds to the vertical or depth direction. The number of times the tile is repeated along the length and depth of a parametric object is determined by the values entered in the **Horizontal** and **Vertical Tiles** fields of the **Texture Map Controls** dialog, respectively. Examples of parametric mappings are shown in Figure 6.3.4.6.

![Parametric texture mapping.](image)

An alternative for defining the size of a texture for parametric mapping is by expressing it as a percentage of the length and depth of the parametric object. Note that the percentage notation is also used to define the origin of the texture tiles. For example, an origin at $X = 25\%$ and $Y = 75\%$ places the texture origin at $1/4$ along the length and $3/4$ along the depth of a nurbz surface. As with the other mapping types, the origin of the texture tiles can also be moved graphically by clicking on the origin with the pointer tool active, and dragging it to the new location. Note that with this mapping type, the location of the origin is confined to within the boundaries of the parametric object.

The parametric texture mapping procedures for c-meshes are based on NURBS and, subsequently, only those **form•Z** c-meshes that can be expressed as NURBS can accept parametric texture mapping. These are the NURBS, B-Splines, Bezier, Continuous Bezier, Quick Cubic and Quick Quadratic curves. Broken Bezier and Tangent Curves cannot be accurately simulated for parametric texture mapping. In addition, all control lines of the c-mesh should have the same smoothing type assigned to them, including the same degree. Parametric texture mapping will not work with c-meshes that have rounded ends or c-meshes that were generated with the **Close Ends** or **All Closed** options. Also, for parametric texture mapping to be possible, all the faces of the c-mesh must have the same color. That is, no face of the c-mesh can be assigned a different color.

Parametric mapping treats the complete object as a single entity, and different faces of the object cannot be assigned different colors, which can be done by using one of the other methods of texture mapping.

How a texture is stretched over a c-mesh or nurbz object is determined by the location of the control points and control lines of the mesh. An equal interval of the texture is stretched between two control points in the length direction and between two control lines in the depth direction.
**UV Coordinates**: After one of the other five mapping types has been assigned to an object, it can be converted into UV coordinates, by selecting this item. That is, the horizontal (U) and vertical (V) values of the textures on an object are “frozen” and are permanently stored at the vertices of the object. For this mapping method the **Tiles** parameter is not used, and the **Size** parameter works relative to the size used for the mapping type from which the UV coordinates are derived. That is, it is expressed as a percentage. 100% generates a texture the same size with the one the UV coordinates are derived from. Larger and smaller values apply a scaling factor.

UV coordinates derived from cubic, cylindrical, spherical, and parametric texture mappings are shown in Figure 6.3.4.7. Note that they generally produce images very similar with those produced with the other texture mappings, but they are frequently not identical, as Figure 6.3.4.7(d) illustrates. In specific, UV mapping derived from cubic or flat mapping produces identical results. UV mapping derived from cylindrical, spherical, and parametric mapping works best if the object contains a high density of faces. Low density objects will show considerable artifacts.

*Figure 6.3.4.7*: UV coordinates derived from (a) cubic, (b) cylindrical, (c) spherical, and (d) parametric texture maps. (1) original mapping and (2) UV coordinates.
6.3.5 Defining the size of textures

When a texture is generated by its default parameters, which is when the Texture Map tool is not used to articulate its mapping onto an object, the size of the texture tile is determined by the values in the Default Texture Map Control tab of the Project Rendering Options dialog. When the Texture Map tool is used, the size of the tiles is determined by the values in the Size fields of either the Texture Map Options or the Texture Map Controls dialogs. Note again that distinct Size parameters are available for wrapped and solid textures.

**Flip:** When this option is selected, the horizontal and vertical directions of the texture are reversed. For example, consider a brick texture mapped onto an object using cylindrical mapping. If the Flip option is off, the horizontal mortar joints run along the circumference of the cylinder. If it is on, the horizontal mortar joints run parallel to the Z axis of the cylinder.

**Horizontal/Vertical Tiling:** The values entered in these fields determine the size of the wrapped texture tiles.

**Size:** The meaning of the values entered in the two Size fields depends on the type of mapping that is applied, as follows:

- **Flat** and **cubic:** The size values represent lengths expressed in the current unit of measurement (English or metric). The Horizontal Size determines the x and the Vertical Size the y dimensions of the texture tile.

  In cubic mapping, the faces of the bounding cube are always parallel to the planes of the texture map coordinate system. For its side faces, the texture’s X is parallel, and its Y perpendicular to the XY plane. For the cube’s top and bottom faces, the texture’s X and Y axes are parallel to the X and Y axes.

- **Cylindrical:** The Horizontal Size value represents an angle expressed in degrees. This angle is measured on the XY plane of the texture coordinate system, and has its tip on the axis of the bounding cylinder. The Vertical Size value represents a length measured along the Z axis of the texture’s coordinate system.

- **Spherical:** Both Size values represent angles in degrees. Horizontal Size measures an angle on the XY plane and its tip on the Z axis. Vertical Size measures an angle on the ZX plane and its tip on the Y axis.

- **Parametric:** The Size values are expressed as percentages of the surface (parametric object) where the texture will be mapped. For example, 100% for Horizontal Size means that the texture is wrapped once around the equator of a parametric sphere.

- **UV coordinates:** The Size values are expressed as percentages, which are relative to the Size value of the mapping type from which the UV coordinates are derived. A 100% value maintains the size of the texture as in the base mapping. Higher and lower than 100 values scale the texture up and down, respectively.
**Tiles**: These parameters offer an alternative method for determining how many texture tiles will be generated. They represent the number of tiles to be placed in the horizontal and vertical directions. For the cubic and UV coordinates methods the **Tiles** parameter is not used and is dimmed.

The **Size** and **Tiles** parameters in the **Texture Map Controls** dialog are interdependent, and when one changes the other is adjusted automatically. In the **Texture Map Options** dialog, they are independent. The settings in **Texture Map Options** do not yet relate to a particular object, thus, there is no way to calculate the number of tiles after the tile size is set, and vice versa.

**Center**: When this option is on, the texture tile is centered in the horizontal and/or vertical direction, relative to its origin. When off, the lower left point of the tile is placed on the texture origin.

**Mirror**: With this option, the texture tile is mirrored in the horizontal and/or vertical direction.

**Lock Size To**: This menu contains five entries, allowing you to lock the proportions of the texture tile to a specific value. That is, if the horizontal tile size is edited, the vertical tile size is adjusted automatically to maintain the selected proportions, and vice versa.

- **None**: When this option is used, the proportions of the texture tile are not adjusted.
- **Square Tile**: With this option, the horizontal and vertical tile sizes are kept equal.
- **Current Proportions**: When this option is selected, the horizontal or vertical tile sizes are adjusted to the proportions present at the time this menu item was last selected.

**Color Map, Transparency Map, Bump Map**: When this option is selected, the horizontal or vertical tile sizes are adjusted to the proportions of the image map, transparency map or bump map used in the surface style displayed in the lower left corner of the dialog. If the surface style does not use any of these maps, the respective menu item is dimmed. Note that when one of these items is selected and the current surface style is changed by clicking in the arrow fields below the box, the menu item is automatically switched to **Current Proportions**. This ensures that the current tile size is not accidentally changed by selecting a new surface style.

When cylindrical mapping is used, the horizontal size is expressed in degrees while the vertical size is expressed in feet or meters. The equality of these different units of measurement is calculated as follows: When calculating degrees from length units, the length units are used to subdivide the perimeter of the bounding cylinder, and the end points of one subdivision are projected onto the axis of the cylinder. The angle between the two projection lines is the angle that is considered to be equal to the length unit. When calculating a length from an angle, it is given by the distance between the two points where the sides of the angle intersect the curved surface of the bounding cylinder. These calculations are also used when switching from a mapping whose sizes are expressed as units of length (such as cubic), to a mapping whose sizes are expressed in degrees (such as spheric), and vice versa.

**Solid Textures Size**: The value entered in this field determines the size of a solid texture. Recall that this single value is uniformly applied to all 3 dimensions (X, Y, and Z) of a solid texture.
6.3.6 Transforming textures with objects

When one of the geometric transformations, namely move, rotate, scale or mirror, is applied to an object to which one or more textures have been applied, or when such an object is copied, the transformed object carries the texture and the mapping of the original object. That is, the textures and their positions are generally transformed with the object.

For example, moving an object also moves the origin of the associated texture map control. This guarantees that the origin and rotation of all texture map controls associated with the object remain the same relative to the object, after the transformation is complete. Note, however, that if transformations are applied on other topological levels, such as the Point level, the origin and rotation of the texture map control are not affected, even if all points are transformed. Transformations applied to objects with both wrapped and solid textures are illustrated in Figure 6.3.6.1.

While moving, rotating, and mirroring objects with texture is straightforward, resizing (scaling) them requires additional considerations. That is, when objects with textures are scaled you may also choose to scale the texture or to keep its original dimensions. This is controlled by an option located in the lower left corner of the Texture Map Controls dialog.
**Scale With Object:** When this option is selected at the time a wrapped texture is mapped onto an object, and then the object is resized using one of the Scale tools, the horizontal and vertical size of the texture tile is scaled with the object by the same factor, and the number of tiles placed in both the horizontal and vertical directions remains the same. If this option is not selected, the texture tile size is not affected by the Scale operation, and remains the same, while the number of tiles placed changes. With UV coordinates this option is dimmed and behaves as if it is always on. That is, UV coordinates are always scaled with the object.

For wrapped textures, this is generally true when using either the Uniform Scale or the Independent Scale tools. However, applying an Independent Scale to an object with cubic mapping may produce results different than might be expected. This is because the X and Y directions of a texture tile do not necessarily coincide with the X and Y axes of the 3D space when the texture is mapped onto the faces of the bounding cube.

When a solid texture is applied to an object which is then resized using the Uniform Scale tool, the same two options are available. If **Scale With Object** was on when the texture was mapped, the texture will be scaled by the same factor applied to the object. If it was off, then the solid texture will retain its original size, even when the object is scaled. Contrary to the wrapped textures, applying the Independent Scale tool to an object to which a solid texture was assigned has no effect on the texture, which retains its original size. How a solid texture is affected when an object to which a texture has been assigned is scaled, with and without the **Scale With Object** option selected, is illustrated in Figure 6.3.6.2.

![Figure 6.3.6.2:](image)

**Figure 6.3.6.2:** (a) Original object with solid texture. With **Scale With Object** on, the object is scaled (b) up and (c) down using the Uniform Scale tool. (d) With **Scale With Object** off, the object is scaled up. (e) With **Scale With Object** on and using the Independent Scale tool, the object is scaled.

How a wrapped texture is affected when an object is resized is illustrated in Figure 6.3.6.3. A grid texture is applied to a 8' x 8' x 2' slab with the flat mapping method, using a 1' x 1' tile size. Then, with **Scale With Object** off, the Uniform Scale tool is used to double the size of the slab. The size of the tile remains the same, but the number of tiles placed in each direction is doubled. The same operation is repeated with **Scale With Object** on. With this option selected, the size of the tile is doubled in both directions and the number of tiles generated remains the same.
Figure 6.3.6.3: Scaling an object to which wrapped texture has been assigned:
(a) The original object with a 2'-0" x 2'-0" tile size assigned to it.
After applying a uniform scale of 2.0 with Scale With Object option (b) on and (c) off.
In general, UV coordinate mapping produces different results than the other mapping types when objects are geometrically transformed.

When transforming a vertex (point) of an object, the UV value associated with it is not changed. For example, when an object is deformed, its UV textures are stretched or compressed. This is illustrated in Figure 6.3.6.4. which shows eight copies of the same object. Cubic mapping has been applied to the four objects in row (a) and UV mapping to the objects in row (b). The UV mapping was derived from the cubic mapping that was originally applied to all the objects.

The objects in column (1) are in their original shape and they look the same. Those in the other columns have been deformed, using the same operation for the pair of objects in each column.

- In column (2) the objects are stretched by moving points.
- In column (3) the objects are deformed using the twist operation.
- In column (4) the objects are deformed using a bend operation.

The cubic mapping is able to better adjust to the deformations than the UV mapping is. That is, the cubic mappings of the textures on the objects in row (a) are adjusted to the new shapes before they are rendered. The UV mappings of the objects in row (b) are not adjusted and the textures simply follow the new positions of the points where they were initially placed by the UV mapping.

Contrary to the other mapping types, UV mapping is unable to adjust when certain modeling operations produce new faces, not in the original object. For example, a Boolean operation frequently produces new faces that are not in the original object. Also, other faces are trimmed and their shape changes. In both cases, the UV values at the vertices cannot be maintained and are therefore dropped. In these cases the default global mapping is applied to the object.

UV mapping works best when an object is made from a fairly dense mesh of faces. Objects with few large faces, such as a cube, may produce unexpected texture patterns.

Lastly, UV texture maps follow the surfaces they are on, when they are unfolded. This is illustrated in Figure 6.3.6.5. Note again that only UV textures unfold properly. The other types of mapping do not.

Figure 6.3.6.4: Rendering the same object with (a) cubic and (b) UV mapping, (1) before and (2,3,4) after the object has been deformed.

Figure 6.3.6.5: Two cuboids with different UV texture maps unfolded.
6.3.7 Texture groups

Textures may be assigned and mapped onto complete objects or groups of faces, where a group may consist of one or more faces. Texture groups are defined using the Texture Groups pop up menu and buttons located at the lower right portion of the Texture Map Controls dialog (Figure 6.3.7.1).

When the Texture Map tool is first applied to an object, the texture is, by default, assigned to the complete object. This is indicated by the word Object that appears in the Current Group pop up menu, which contains no other entry. This menu expands as new texture groups are defined.

New...: This button allows you to define a new texture group and add it to the Current Group menu. Clicking on it invokes the New Texture Group Name dialog (Figure 6.3.7.2) where a default name for the new texture group appears. A different name can be entered in the usual manner. After closing the dialog, the new texture group appears in the Current Group menu and becomes the active group. However, the texture group is not yet completely defined.

As soon as a new texture group is defined, the image of the object displayed in the preview window is dimmed to indicate that no face is currently contained by the group. Next, you need to select the faces that will be in this group. Select the faces in the usual manner. That is, if the Clicking On Edges option is selected in the Pick Options dialog, click on two segments of the face. If the Clicking Inside Boundaries option is on, click once inside the boundary of the face. One or more faces can be selected, and these faces will automatically be stored with the group, together with the position of the texture which you would normally assign next. If a face is selected which is already part of the current texture group, it is deselected and returned to the Object texture group. Note that faces can also be picked using the Lasso or Frame method.

The faces belonging to a group are shown in the preview window in black when the Wire Frame display option is used. When the Tiled display type is used, only the faces that belong to the active group will be tiled, if they are facing front. If they are facing back, you can rotate the view to see their tiling. When the Rendered display type is used, the complete object is rendered.

To select faces that may be behind other faces, the pick parade method can be used, as discussed in section 4.3.3 of the form•Z User’s Manual. That is, with the Clicking Inside Boundaries option selected in the Pick Options dialog, you click on a face while pressing the shift key. This will select a face. Continuing to click the mouse will parade through all the faces that contain the click point. Continue clicking until you select the face you desire.
When a face is picked to make it a member of the active texture group, the texture coordinate system can also be placed on it. This is done by pressing the *option* key on the Macintosh or *ctrl+shift* on Windows when selecting the face. The origin of the texture coordinate system is always placed at the center of the face, the X and Y axes are placed on the surface of the face, and the Z axis is always perpendicular to the face. The orientation of the X and Y axes depends on how the face is picked.

When the **Clicking On Edges** method (**Pick Options** dialog) is used to select the face, then the orientation of the X axis is determined by the orientation of the first segment selected. The location of the positive portion of the Y axis is determined by the position of the second segment selected, relative to the first segment selected. When the **Clicking Inside Boundaries** picking method is used, the X axis is parallel to the segment that is closest to the pick point. The positive Y axis is always to the left of the positive X axis. Pressing the option key when selecting faces with the **Lasso** or **Frame** method has no effect.

This method of automatically mapping the texture coordinate system onto a face when it is selected offers convenience, particularly when a texture needs to be mapped onto a surface which is not parallel to one of the Cartesian planes. This is illustrated in Figure 6.3.7.3, where a flat texture is mapped to each of the three visible faces of a pyramid.

After a new **Texture Group** is defined and the **Flat** mapping type is selected, the texture coordinate system is positioned with its X and Y axes parallel to the world coordinate’s X and Y axes, as shown in Figure 6.3.7.3(a). It would usually take a number of moves and rotations to place the texture’s X and Y on the surface of a face. However, pressing the *option* (Macintosh) or *ctrl+shift* (Windows) keys when the face is selected maps the texture coordinate system onto the surfaces of the faces, as shown in Figure 6.3.7.3(b), (c), and (d). The pyramid rendered with brick texture mapped to individual faces is shown in Figure 6.3.7.3(e).

---

**Figure 6.3.7.3:** Mapping textures to individual faces: (a) Original position of texture. (b), (c), and (d) Texture mapped to individual faces when the faces are selected for texture groups. (e) The resulting rendering.
After one or more texture groups have been defined and faces assigned to them, and Object is selected in the Current Group menu, you notice that the faces assigned to other groups have been eliminated and do not appear. It is possible to assign all the faces of an object to other texture groups, resulting in an empty Object texture group.

The texture mapping parameters, including the mapping type, can be completely different for each of the texture groups. Also, different surface styles can be assigned to all faces in a texture group. This is done by clicking on the Assign... button located at the lower left portion of the Texture Map Controls dialog.

**Delete**: Clicking on this button deletes the active texture group. This is the group whose name appears in the Current Group menu. The Object group cannot be deleted, and when Object is the active group, this button is dimmed and inactive.

**Rename...**: This button is used to change the name of a previously defined texture group. Clicking on it invokes the New Texture Group Name dialog where a new name is typed. The Object group cannot be renamed and, when Object is the active group, this button is dimmed and inactive.

**Select All Faces**: When this button is pressed all faces of the object are assigned to the current texture group.

**Deselect All Faces**: When this button is pressed all faces of the current texture group are returned to the object group.

Texture groups can also be defined on the basis of the surface styles already assigned to faces of objects. Recall that the Color tool can be used with topological level set to Face to assign different surface styles to different faces of an object. After the New button is used to create a new, initially empty texture group, the Pick By... button, located at the bottom of the Texture Map Controls dialog, can be used to select the faces that have been assigned the same surface style.

**Pick By...**: This button is active when a texture group name appears in the Current Group menu. It is dimmed and inactive when Object is displayed. Clicking on it invokes the dialog shown in Figure 6.3.7.4. It displays the surface styles that are currently assigned to the object. Any of the displayed surface styles can be selected by clicking on it. When you exit the dialog by clicking OK, all the faces to which the selected surface style is assigned will become members of the texture group whose name appears in the Current Group menu.

**Figure 6.3.7.4**: The dialog for selecting surface styles to construct texture groups.
Assigning and editing surface styles

Surface styles can be assigned to objects using the Color tool. They can be edited by double clicking on them in the Surface Styles palette. Both of these operations can also be executed through the Texture Map Controls dialog, which is invoked by the Texture Map tool, when it is used in edit mode (Edit is on in the Texture Map Options dialog).

The surface style that is currently assigned to the object (or the active texture group, see above) is displayed in a box at the lower left corner of the Texture Map Controls dialog. Next to it is the Assign... button.

Assign...: Clicking on this button invokes the Select A Surface Style dialog, shown in Figure 6.3.7.5. Note that it is identical to the Surface Styles dialog invoked when you click in the Surface Styles palette while pressing option (MacOS) or ctrl+shift (Windows).

This dialog contains buttons through which new surface styles can be defined or existing ones can be deleted, copied, and edited. When the dialog is closed by clicking on OK, the selected surface style is assigned to all the faces of the current texture group and is displayed in the preview box.

Note that this command does not associate a surface style with a particular texture group, but assigns the surface style as an attribute to the faces in the texture group. This process is equivalent to preselecting a number of faces in the main window and applying the Color tool.

If there is more than one surface style in use, the next or previous surface style can be displayed by clicking in the respective arrow below the box. The displayed surface style can also be edited by double clicking on it, which invokes the Surface Style Parameters dialog, where changes to the surface style can be made.
6.3.8 Mapping solid textures

From the parameters in the **Texture Map Controls** dialog, the options that have significance for the solid textures are the positional parameters **(Origin and Rotation)**, **Solid Textures Size**, and **Scale With Object**. Solid texture parameters can also be assigned to **Texture Groups**. Solid textures are not affected by the **Wrapped Textures** options, such as the **Mapping Type** and the **Tiling** sizes.

Note that, while solid textures can be positioned relative to objects in different ways to generate different variations of their patterns, these differences are most visible with solid textures that have directions, such as streaks, wood, checkers, or polka dots. The wood texture is used in the examples of this section.

When a solid texture, such as wood, is first assigned to an object when it is created, its Z axis, which corresponds to its direction, is placed by default on the Z axis of the world coordinate system. An example of this default positioning of the wood texture is shown in Figure 6.3.8.1(a), where each wood board is modeled as a solid object. One can quickly observe that this positioning is not natural for the particular wood planks shown. They appear to have been cut the wrong way from the trunk of the tree, since the direction of wood grain is typically along the length of the boards. Such an adjustment can be made using the Texture Map tool.

In the example of Figure 6.3.8.1(b) each texture’s coordinate system has been rotated so that the Z axis is parallel to the length of the boards. The origin of each texture is placed at the center of each object. Note that the wood already looks more natural. However, all the boards that are in the same direction have exactly the same wood texture. This is because the generation of the texture starts at exactly the same relative position for all the boards, and all the boards are exactly the same size. While this is inappropriate in this particular case, it does illustrate an advantage, which is the ability to reproduce a specific solid texture when the mapping parameters are kept constant.

In the example of Figure 6.3.8.1(c) the origin of each texture has been set to the origin of the world coordinate system for all the objects. Each board now has a distinctly different texture, since the solid wood texture was intercepted at different stages of its generative process.
Note that in the last two examples the texture looks too “straight.” This is because the direction of the texture has been kept parallel to the lengths of the boards. The example in Figure 6.3.8.1(d) illustrates two variations of more natural wood textures. They were derived by placing the texture origin at different positions for the different boards, and by rotating the texture coordinate system so that its Z axis (direction) is not parallel to the length of the board.

Figure 6.3.8.1: Mapping the directions of solid wood textures:
(a) Default mapping with direction of texture placed on the world Z axis.
(b) Direction of textures placed along the lengths of the boards and the origins at their centers.
(c) Directions along lengths of boards and origins at world origin.
(d) Directions roughly parallel to lengths of boards, and origins at different positions.
A more elaborate rendering using solid textures is shown in Figure 6.3.8.2. As in the previous example, a texture map has been assigned to each object, which controls the size and direction of the wood grain.

Figure 6.3.8.2: A train model mapped with solid wood textures.

Once an object is assigned its own texture map parameters, they can be cleared by pressing the **Clear To Default** button in the **Texture Map Controls** dialog. This will post a warning message, and pressing **OK** will close the **Texture Map Controls** dialog. This will remove any previously assigned texture map parameters, and the global default parameters, as displayed in the **Default Texture Map Control Options** tab in the **Project Rendering Options** dialog, will be used for the next rendering.
6.3.9 Examples

Simple mapping of textures

This example shows how the image in Figure 6.3.9.1 was rendered using simple brick textures that were wrapped around the objects. Two different surface styles are used, both of which are based on the Simple Brick color shader. One lines up the mortar joints vertically. The other has the vertical mortar joints offset by 50%. Depending on the desired brick effect, one or the other surface style is used.

- Figure 6.3.9.2 shows the Texture Map Controls dialog for the floor. Flat mapping type is used. The origin is aligned with the lower left corner of the object. By typing 6 for # of Tiles, a grid of 6x6 tiles is generated.

- Figure 6.3.9.3(a) shows the Texture Map Controls dialog for one of the column bases. Cubic mapping is used and the origin is aligned with the center of the object. Note that, in the rendering, the edges of the object show mortar joints, which is not correct for a real stone base. By slightly increasing the tile size, the problem is corrected (Figure 6.3.9.3(b)).
• Figure 6.3.9.4 shows the **Texture Map Controls** dialog for one of the columns in the rendering. **Cylindrical** mapping is used to wrap the brick texture around the object. With the **Lock Size To** option set to **Square Tile**, only the vertical size needs to be entered, in this case 2'-0". The horizontal angular dimension is automatically computed to generate a square tile on the cylindrical surface. In a similar fashion, the stones on the arch can be mapped. Note that the orientation of the cylindrical mapping space is aligned with the axis of the arch, as shown in Figure 6.3.9.5.

![Figure 6.3.9.4: Setting the texture for the columns.](image)

![Figure 6.3.9.5: Setting the texture for the arch.](image)

**Using texture groups**

In some instances, different mapping types need to be used on the same object. The following example illustrates how a simple brick texture is mapped on an object using texture groups.

• The rendering in Figure 6.3.9.6 shows a simple set of walls with a brick texture. Note how the texture follows exactly the angled and curved walls.

![Figure 6.3.9.6: A simple model rendered with a brick texture](image)
• Figure 6.3.9.7(a) shows the **Texture Map Controls** dialog for the object. The main mapping type is **Cubic** with a tile size of 1'-0". This will map 12 inch bricks correctly on all orthogonal surfaces.

• Figure 6.3.9.7(b) shows the **Texture Map Controls** dialog with a new texture group. Its mapping type is set to **Flat** and the orientation of its axes is aligned with the direction of the angled wall. All the faces of this angled wall are selected to become members of this texture group.

• In the same fashion, two more texture group are set up to map the texture on the other angled wall and on the curved wall, with **Flat** and **Cylindrical** type of mapping, respectively. The setting of these textures is shown in Figures 6.3.9.7 (c) and (d).

---

**Figure 6.3.9.7**: Setting the textures for (a) **Object**, (b) **Texture Group 1**, (c) **Texture Group 2**, and (d) **Texture Group 3**.
Textures which follow curved surfaces

In form\textbullet\textsc{Z}, certain controlled objects are based on analytic or spline surfaces. For these object types, the parametric mapping type is available. Figure 6.3.9.8 shows a snake, where the skin texture is mapped in such a way that it follows exactly the shape of the body of the snake. It also stretches and compresses in the same manner as the snake’s skin. The snake object is modeled as a nurbz controlled object. In the Texture Map Controls dialog for this object, shown in Figure 6.3.9.9, the Parametric mapping type is selected. Also note that the number of repetitions of the texture is set to 1 horizontally and 15 vertically.
6.4 Decals

Additional surface styles can be assigned to objects as *decals*. After a basic surface style has been assigned and mapped to an object using the Color and Texture Map tools, the Decal tool can be used to assign additional patterns. This feature is useful for rendering stickers or labels, and for arranging a variety of different textures on a single surface. Note that the Color tool can only be used to assign a single surface style to an object (or face of an object). The Decal tool allows you to assign multiple surface styles, which may even overlap. Using this tool, a maximum of 32 additional surface styles can be assigned per object.

The decal feature offers the ability to produce a number of different effects that go beyond the simple placement of adhesive stickers or mixed repetition of different surface styles. Decal surface styles can also be mixed with the underlying surface style. This allows you, for example, to apply just the bump shader of the decal surface style. How a decal surface style is mixed with the underlying surface style is determined by the *Transparency* shader of the decal surface style. Areas of the decal surface style which are completely transparent are rendered using the underlying surface style. Areas which are completely opaque are rendered with the decal surface style. Areas with a partial transparency are rendered by mixing the decal surface style and the underlying surface style according to the level of transparency. Examples of different applications of decals are presented at the end of this section.
6.4.1 The Decal tool

Decals are applied to an object using the Decal tool, which is located next to the Texture Map tool in the modeling tool palette.

This tool can be used to apply and map procedural and pre-captured textures onto surfaces of objects, on top of other surface styles that may have been assigned and mapped either by default or by using the Color and the Texture Map tools. It can be used with either the prepick or postpick method.

When using the postpick method, with the Decal tool active, click on the object to which you wish to attach a decal. This invokes the Decals dialog (Figure 6.4.1.1), which allows you to place, size, and orient decals on an object. The postpick method always picks at the Object level, regardless of the topological level currently selected. To use the prepick method, with the Pick tool active and the topological level set to Object, preselect any number of objects. Then activate the Decal tool and click anywhere in the graphics window. The Decals dialog will be invoked once for each of the objects preselected.

The Decals dialog is very similar to the Texture Map Controls dialog. It contains parameters for defining the decal origin, rotation, size, and mapping type, a preview window, and commands for manipulating the display of the preview and for mapping a decal on the object. It also contains a group of options which are unique to this dialog. These are the options that allow you to create and name new decals, and to associate a particular surface style with them. These options are at the lower end of the dialog and are discussed in section 6.4.2.

The three groups of options at the upper right portion of the Decals dialog, Origin, Rotation, and Mapping Type, are identical to those in the Texture Map Controls dialog (see section 6.3.2, Figure 6.3.2.2).
The options for sizing the decals, located in the middle right portion of the dialog, are very similar to those in the **Texture Map Controls** dialog, but contain additional parameters for determining how many times the decals will be repeated. These parameters are similar to parameters found in the **Color Map Options** dialog (section 6.2.3, Figure 6.2.3.59).

**Horizontal, Vertical**: These groups of parameters determine the size for wrapped (surface) decals, and the number of repetitions in the horizontal and vertical directions.

**Size**: The values entered in these fields determine the size of a single decal tile, in the horizontal and vertical directions, respectively. Note that wrapped textures at a scale factor of 100% fit exactly once in the size of the decal. Depending on the mapping type, the **Size** values are expressed as length units, degrees, or percentages, as follows:

- **Flat** and **Cubic**: both horizontal and vertical sizes are expressed in length units.
- **Cylindrical**: the horizontal size is expressed in degrees and the vertical size is expressed in length units.
- **Spherical**: both sizes are expressed in degrees.
- **Parametric**: both sizes are expressed as percentages which represent the amount of coverage in each direction, relative to the entire length/depth of a parametric object. Recall that Parametric mapping can only be applied to certain form•Z parametric objects.

**Center**: When this option is on, the decal is centered about the origin.

**Mirror**: When this option is selected, each alternating repetition of a decal is mirrored.

The option to place a single, or a specific number of decals, or to instruct the program to repeat the decal as many times as necessary to completely cover a surface is controlled by the following two mutually exclusive radio buttons:

- **Infinite**: When this option is selected, the decal is repeated as many times as necessary to cover the surface, in the horizontal and vertical directions, respectively.

- **Times**: When this option is selected, the decal is repeated as many times as indicated in the text field next to the radio button.

**Solid Texture Size**: The value entered in this field determines the size of solid textures. Note that, as in the **Texture Map Controls** dialog, solid textures are sized independently from wrapped textures and that the same size value is applied to all three directions.

**Scale With Object**: When this option is selected, the decal size is scaled proportionally when the object is scaled using one of the Scale tools. Recall that the size of tiles applied with cubic mapping and the size of solid textures are adjusted only when the Uniform Scale tool is applied. The Independent Scale operation has no effect on them.

**Visible**: When this option is selected, the active decal is made visible and is displayed when a rendering is produced. When off, the respective decal is skipped.
6.4.2 Creating decals

Decals are created through button commands found at the lower portion of the Decals dialog. They are listed in a palette-like window, to be referred to as the decal list. After new decals are created and a name is assigned to them, they are added to the list, in the order they are created. When the decals list is empty, all the fields in the Decals dialog appear inactive. This is the case, for example, when the Decal tool is applied to an object for the first time. One of the decals in the list is the active decal and its parameters are displayed in the dialog. Another decal can be made active by clicking on its name in the list. There can only be one active decal at any given time. The surface style with which the active decal is associated is displayed in an icon at the lower left end of the Decals dialog. This image changes each time the active decal changes.

An object can have up to 32 decals, which may or may not overlap. When they do, the order in which they are displayed at rendering time is determined by the order in which they appear in the decal list. The decal at the end of the list is applied first, and the decal at the top of the list is applied last and covers all the decals below it. The order of the decals is initially determined by the order in which they are created. This order can be changed by clicking on the name of a decal and dragging it to a new position in the list.

New...: This button allows you to create a new decal and add it to the decal list. Clicking on it invokes the New Decal Name dialog, where a default name for the new decal appears. A different name can be entered in the usual manner. After closing the dialog, the name of the new decal appears in the decal list and becomes the active decal.

Delete: When this button is pressed, the active decal is deleted.

Rename...: When this button is pressed, a new name can be assigned to the active decal. Clicking on it invokes the New Decal Name dialog, where a new name can be entered.

Select All Faces: When this button is pressed all faces of the object are assigned to the current decal.

Deselect All Faces: When this button is pressed all faces of the current decal are removed from it.

When new decals are created, by default, they are associated with the currently active surface style. They can be associated with another surface style using the following button command.

Set Style...: When this button is pressed, a dialog that displays the surface style palette is invoked and a new surface style can be selected. This new surface style becomes the surface style associated with the active decal, and is displayed in the square icon above it. This surface style can be edited by double clicking on the icon, which invokes the Surface Style Parameters dialog.

Color, Reflection, Opacity/Transparency, Bumps: These four options can be used to vary the effects of the decal shaders when they are rendered. That is, when an option is on, the settings of the respective component of a shader will be applied. When it is off, they will be ignored. By default, Color and Reflection are on, and Opacity/Transparency and Bumps are off.
Note that three of these options are plain check boxes that can be either on or off. The **Opacity/Transparency** option is a check box and also a pop up menu from which either **Opacity** or **Transparency** can be selected. Each results in a significantly different effect. These pop up menu items determine whether the transparency shader component of a decal will be used in an additive or subtractive manner.

**Opacity**: When this item is selected from the pop up menu, the transparency of the surface on which the decal is placed is decreased, or the transparency of the decal is subtracted from the transparency of the surface. Areas of the decal that have 0% transparency will generate 0% transparency on the surface covered by the decal. Areas of the decal with 100% transparency will not affect the transparency of the underlying surface. Transparency levels between 0 and 100% will cause the transparency of the surface to be decreased. For example, if a surface with 80% transparency is covered by a decal with 50% transparency, the resulting transparency will be $0.80 - ((0.0 + 0.80) \times 0.50) = 0.40 = 40\%$.

This method is useful when placing opaque stickers on a transparent surface. Figure 6.4.2.1 shows an example, where a decal in the shape of a star is placed on a bottle. The decal surface style uses a star shaped transparency map. The transparency of the bottle is decreased to 0% where it is covered with the star shape of the transparency map. The area of the decal which is covered with the background of the star, is left at its original transparency.

**Transparency**: When this item is selected, the transparency of the surface on which the decal is placed is increased. Areas of the decal that show 100% transparency will generate 100% transparency on the surface covered by the decal. Areas of the decal with 0% transparency will not affect the transparency of the underlying surface. Transparency levels between 0 and 100% will cause the transparency of the underlying surface to be increased. For example, if a surface with 30% transparency is covered by a decal with 50% transparency, the resulting transparency will be $0.30 + ((1.0 - 0.30) \times 0.50) = 0.65 = 65\%$.

This method is useful when cutting out transparent patterns from an opaque surface. The example in Figure 6.4.2.2 uses the same star shaped transparency map as the previous example. However, the star now has full transparency and the background is opaque. When placed on an opaque object, the decal produces star shaped holes in the object.

More decal examples are shown in section 6.4.4.
6.4.3 Determining the faces to be affected by a decal

When a new decal is created, it initially affects all the faces of the object to which the Decal tool was applied. However, any number of faces can be exempted from it.

You turn a face off relative to the active decal by selecting it in the preview area. Faces exempted from a decal are drawn in the project’s ghost color when the decal is active. Picking an exempted face again turns it back on and the face is drawn in black. Faces can be picked as in the Texture Map Controls dialog. All faces of an object can be turned on relative to the active decal, regardless of their current status, by pressing the +a keys on the Macintosh or ctrl+a on Windows. Vice Versa, they can be all turned off by pressing +tab on the Macintosh or ctrl+tab on Windows.

Examples of decals affecting all or only some of the faces of an object are shown in Figure 6.4.3.1, where a decal is applied to an object using flat mapping. In (a), where no faces were turned off for this decal, it appears on all the surfaces on which a projection of the decal occurs. In (b) only two faces are turned on for this decal, which is rendered only on these surfaces.

Figure 6.4.3.1: A decal applied (a) to all faces, and (b) only to selected faces of an object.
6.4.4 Examples

Creating a bottle with labels

This example outlines how the bottle shown in Figure 6.4.4.1 was rendered by applying three decals. All the decals were applied using surface styles with texture maps, where the **Color Map** was generated from the color channels of image files. For one of these surface styles, the **Transparency Map** used the alpha channel of an image file.

- The bottle itself was rendered using a surface style with a **Glass** reflection shader, to generate accurate transparencies and reflections.

- Chardonnay label: As shown in Figure 6.4.4.2, this decal is applied using one tile (**Times** = 1 for both **Horizontal** and **Vertical**), centered in both directions, and cylindrical mapping. The decal surface style is set to use the **Color**, **Reflection**, and **Opacity**, which causes the decal to override the color, reflection, and transmission of the bottle glass.

*Figure 6.4.4.1:* A bottle rendered with decals.

*Figure 6.4.4.2:* The Chardonnay label and the parameters set in the **Decals** dialog to render it.
- Circular label: The surface style defined for this decal, for its **Color Map**, uses the color channel of the image shown in Figure 6.4.4.3. For its **Transparency Map**, it uses the alpha channel of the same image. In the Decals dialog, the same settings as the previous label are used. In the decals list, this label is listed above the Chardonnay (called main) label, which causes it to be rendered on top.

*Figure 6.4.4.3:* The circular label and the parameters set in the Decals dialog to render it.

- Neck label: As shown in Figure 6.4.4.4, for this label, all the parameters in the Decals dialog are set as for the other labels, except that **Infinite** is selected for the Horizontal direction. This causes the label to be repeated as many times as necessary to cover the perimeter of the neck of the bottle.

*Figure 6.4.4.4:* The neck label and the parameters set in the Decals dialog to render it.
Surfaces with partial reflections

Imagine a marble which has silver streaks. While most of the marble has a matte appearance, the silver streaks are reflective. This example describes how such a material can be applied to a spherical object and how the image in Figure 6.4.4.5 can be rendered.

Note that when assigning a surface style to an object with the Color tool, a surface such as marble can be made reflective and matte using reflection parameter shaders as discussed in section 6.2.4. Alternatively, this effect can be achieved with the Decal tool.

• Two surface styles, called Stone and Mask, are defined using the same shader, *Marble, Textured*.

• For Stone, *Marble, Textured* is used as a Color shader, and *Matte* is used for Reflection (see Figure 6.4.4.6).

• For Mask, *Plain Color* is used for the color shader, *Mirror* is used for the reflection shader, and *Marble, Textured* for the transparency shader (see Figure 6.4.4.7). The parameters of the *Marble, Textured* transparency shader are set to the same values as those used for the *Marble, Textured* color shader of the first surface style.

• Surface style Stone is assigned to the sphere using the Color tool.

• Surface style Mask is assigned to the sphere as a decal, using the Decal tool (Figure 6.4.4.8). Note that only the reflection component of the surface style is used. The transparency pattern of the decal surface overlays precisely with the streak pattern of the *Marble, Textured* color shader of the first surface style, restricting the reflectivity to the silver streak areas of the surface style.

Figure 6.4.4.5: Partially matte and partially reflecting marble.

Figure 6.4.4.6: Defining the Stone surface style.

Figure 6.4.4.7: Defining the Mask surface style.

Figure 6.4.4.8: Placing the Mask surface style as a decal.
Mixing bumps

This example illustrates how to render an image such as that shown in Figure 6.4.4.9, by using decals to mix bumps of different types.

• First, a surface style that uses Brick, Simple both as a color and as a bump shader is applied to the wall, as shown in Figure 6.4.4.10. Note that the bump shader raises the bricks and recesses the mortar, but leaves the brick and mortar surfaces themselves smooth.

• Next, three decals are applied to the wall object.

• The first decal surface style uses Simple as transparency shader, with transparency factor set to 50%, and Rough a bump shader. When the decal is applied, only the bumps component of the surface style is turned on in the Decals dialog (Figure 6.4.4.11). As a result, the wall renders with the original color, reflection, and transparency, but mixes the brick bumps with the rough bumps of the decal surface style. Also note that the repetition for this decal is set to Infinite for both Horizontal and Vertical, so that the bumps cover the entire object.

• The second decal, shown in Figure 6.4.4.12, applies a single tile of a texture map to the wall. The surface style for this decal uses the same image for both its Color and its Opacity component. The former uses the color information and the latter the alpha channel of the image. This decal is applied by turning on only the color and reflection shaders of the surface style. When rendered, it allows the mixture of the rough and brick bumps of the wall to show through it, but applies the color of the texture map, masked by the transparency map. That is, the decal looks like it was sprayed on the wall with a can of spray paint.
The third decal again uses a single tile of a texture map. The decal surface style uses an image for its color shader, Matte for reflection, no opacity, and no bumps. When this decal is applied, the color, reflection, and bumps component of the surface style are turned on (Figure 6.4.4.13). Since no transparency mask is used in the decal surface style, the decal overwrites the color, reflection, and bumps of the wall. This decal gives the impression that a picture on a sheet of paper is pinned on the wall.

Figure 6.4.4.12: (a) The image map used for decal “graffiti”, and (b) its parameters as set in the Decals dialog.

Figure 6.4.4.13: (a) The image map used for decal “picture,” and (b) its parameters as set in the Decals dialog.
6.5 Render Textures

6.5.1 The Render Textures tool

For certain applications, it is useful to convert the procedural RenderZone textures of a surface style assigned to an object’s faces to one or more pre rendered texture maps and assign these texture maps back to the faces. This process is also known as texture baking. It can be executed by using the Render Textures tool. This tool can be used in prepick or postpick mode. After the user selects the objects and clicks in the project window, one or more texture map files are generated, and placed in a designated folder. A new surface style is created for each object, and the generated texture maps are used in the various image based shaders of the surface style. The surface style is assigned to the object together with a new texture map control attribute, which uses the UV coordinates mapping mode. The rendered texture map contains the textures of all faces of the object, as if the faces were laid out flat and arranged in a jig saw puzzle fashion, to optimally fit in a square area. That is, each face of the object represents a rectangular area in the texture map. The UV coordinates assigned to the face reference the shape of the face in the texture map. Once rendered again, the object will show little or no difference in its appearance, except that the textures are now based on a pre rendered image map. A simple object and its pre rendered texture map for the color shader is shown in Figure 6.5.1.1.

Figure 6.5.1.1: (a) An object rendered before the Render Textures tool is applied to it. (b) The resulting color texture map, and (c) the shape of the faces contained in the alpha channel of the texture.
In the above example, one can see that the color texture does not actually show the shape of each face, but shows just a rectangular area for each face. This is done on purpose, to avoid artifacts when the texture is used with MIP mapping and summation table sampling. In order to see how a face is positioned in the rectangular area in the texture, the face is rendered as a white shape into the alpha channel of the texture. Naturally, this can only be done if the format in which the texture is saved supports an alpha channel. This is shown in Figure 6.5.1.1(c).

Note that this tool is not intended for every day use. It is useful, for example, for previewing objects that may be exported to a file format that does not support procedural textures. It may also be used to permanently fix a texture on an object. A user may then modify the pre rendered texture in an image processing software to add additional effect. For example a user may pre render an object that represents a wall and that uses the procedural brick shader, and then use Photoshop to paint more detail on the pre rendered brick texture. Note that the pre rendered texture will usually result in a lower quality rendering of the object, as the pixel resolution of the texture will limit the amount of detail that can be shown. If the area of the texture assigned to a face becomes small, rendering artifacts may become quite noticeable. Therefore, the tool works best on objects that have few faces.

The Render Textures tool generates textures in two different ways. First, the RenderZone procedural textures for each shader can be rendered into separate textures. In the new surface style that is assigned to the object, each resulting texture map is chosen in the respective class of shaders. Only those procedural shaders that make sense to be rendered actually generate a texture map. For example, if the object uses the Plain color shader, no texture is generated, since the texture map would show nothing but a solid color. Up to 9 texture maps may be generated for each object:

- A color map
- An ambient reflection map
- A diffuse reflection map
- A specular reflection map
- A mirror reflection map
- A transmission reflection map
- A glow reflection map
- A transparency map
- A bump map

The names of the texture maps are unique for each object and contain a set of 32 characters that make up a unique pattern. This avoids the possibility that two objects will generate a texture map with the same name. Since each texture map is highly customized for a particular object, it usually cannot be shared and it is important that the correct texture map stays assigned to an object.
The second method of generating pre rendered textures creates only one texture per object. This one texture contains the effects of all the shaders. This includes shading and shadow casting. The texture is assigned as the Image color shader to the new surface style of the object. The reflection shader of the new surface style is set to \textbf{Constant}, since the shading is now contained in the color texture. All other shaders are set to \textbf{None}. Note that a rendering with this texture may not be entirely accurate, as certain shading effects, such as specular highlights or mirrored reflections, are view dependant. Therefore, if the object is viewed from a different position, or if the lighting conditions change, the object will not be rendered faithfully. A pre rendered texture with all shaders combined is shown in Figure 6.5.1.2.

Note that the pre rendered textures contain the effects of the RenderZone shaders only. However, once the tool has been applied, other rendering modes that support image based textures may be used to show the object. For example, the Interactive Shaded mode can also display objects that previously had a procedural color texture. It is not recommended though, to convert all objects in a scene with this tool, just to view all textures in the Interactive Shaded display mode. Since the texture mapping mode will be UV coordinates for the objects, editing of the texture location is very limited. In addition, using lots of textures will consume considerable amounts of memory and may slow down the performance of a computer. While it is possible to revert the effects of the Render Textures tool on an object by using the \textbf{Undo} command, the new surface styles remain in the Surface Style palette and the texture maps stay on the disk.

\textbf{Figure 6.5.1.2:} The pre rendered texture of the same object as in Figure 6.5.1.1, but with the effects of all shaders combined.
The options that determine the execution of the tool are found in the Render Textures dialog, shown in Figure 6.5.1.3, that is invoked from the Render Textures tool.

**Texture Resolution**: The value entered in this field determines the resolution of the texture in pixels. For example, if 512 is entered, the texture will measure 512 pixels wide by 512 pixels high. The higher the pixel resolution, the more accurate the rendered texture will be, but more memory will be consumed.

**Separate Textures**: When this option is selected, the procedural textures of each shader of the surface styles assigned to an object’s faces are rendered into separate texture maps as described earlier.

**Combined Textures**: When this option is selected, the combined effects of all shaders of the surface styles assigned to an object’s faces are rendered into a single texture, which becomes the color texture map of an object.

**File Format**: The image file format in which the textures are saved can be chosen from this menu. All installed formats that support pixel based images are available in this menu.

**Destination**: There are two options for the location of the textures. If **With Project** is chosen, the textures are located in the folder where the form•Z project is saved. If the project has not been saved yet, the form•Z application folder is used. If **Custom** is selected, the user may choose a folder manually. This is done by pressing the Set Destination Folder... button, which will invoke the standard folder selection dialog. The currently active destination folder is shown at the bottom of the dialog.
6.6 Lights

In form·Z RenderZone Plus multiple light sources can be defined as one of eight available types: distant, point, cone, projector (gel), area, custom, line, and environment lights. One of the distant lights is designated as the sun, and is used by the lower rendering levels which have only one light (Quick Paint* and Surface Render*). These are in addition to the ambient light that illuminates all surfaces equally. Light color and intensity of the ambient light are set in the Lights dialog (see section 6.6.1). The eight light types, the type of light they emit, and the type of shadows they cast are shown in Figure 6.6.0.1(1). They appear in wire frame display and they are marked visible in the Lights palette (section 6.6.2).

**Distant** (or direct) light emits parallel rays from an infinitely distant light source, such as the sun. The intensity of this light remains constant throughout a scene. Atmospheric light, which simulates light from the sky and which is non uniform in direction can be added.

**Point** lights emit rays from a given point outward in all directions. The intensity of these lights may or may not decrease as the distance from the source increases. An example of a point light is a candle.

**Cone** lights emit rays from a given point in the direction defined by their conic shape. The intensity of these lights may or may not decrease as the distance from the source increases. The intensity of these lights also decreases at the perimeter of the cone. This area of decreased intensity is defined by a second cone inside the outer cone. Car headlights are an example of a cone light.

**Projector** lights emit rays from a given point in the direction defined by a pyramid that is associated with them. The intensity of these lights may or may not decrease as the distance from the source increases. The rays emitted by such lights are also filtered through an image map, projecting that image onto a scene. An example of a projector light would be a slide projector.

**Area** lights are associated with physical objects whose surfaces are emitting light. For example, it is possible to model a curved neon tube and turn the resulting object into an area light. The intensity of these lights may or may not decrease as the distance from the source increases.

**Custom** lights represent variable intensities in different directions about a light source. For example, a point light, which is used to represent a bulb hanging in a room is assumed to emit rays of equal intensity in all directions. However, light sources do not work this way in reality. A bulb has a higher intensity in the direction of the socket, and its intensity falls off toward the socket, and it is zero directly behind the socket. Custom lights take into consideration these real world “irregularities.” The intensity of these lights may or may not decrease as the distance from the source increases.
*Line* lights are derived from and are associated with open or closed surface objects in the same way area lights are derived from surface or solid objects. However, the object of a line light is not rendered in the same way it is for an area light. Light rays are emitted from the segments of an object rather than the object as a whole. The intensity of a line light may or may not decrease as the distance from the source increases.

*Environment* lights allow you to use images that may be mapped on an environment as lights themselves, resulting in more accurate lighting effects.

At start-up, *form•Z RenderZone Plus* automatically creates one distant light, representing the sun. In addition, a scene always contains ambient light. Distant, point, cone, projector, area, custom environment and line lights and their names are displayed in the Lights palette. Lights can be defined and existing lights can be deleted or edited through the Lights dialog. The parameters and the positions of lights can be set in the Light Parameters dialog. Their positions can also be manipulated using the geometric transformation tools of *form•Z*. The ability to cast or not cast shadows at different quality levels is also available. The specific light functions are discussed in the remainder of this section.
Figure 6.6.0.1: The form•Z RenderZone Plus light types: (a) distant, (b) point, (c) cone, (d) projector, (e) area, (f) custom, and (g) line. (1) Lights as shown in wire frame. Renderings (2) without and (3) with shadows.
6.6.1 Measuring light

In Physics, a basic unit of measurement is **power**. It represents energy per unit time interval. For light calculations, power is used to determine how much electric energy is used by a light source. Electric power is measured in **watts** (volt x ampere), which in itself gives no indication about how much light is emitted by a light source. To measure the light output, we also need to specify how much of the electric power supplied to a light is converted to visible radiation. This is referred to as the **efficiency** factor. A combination of this factor and watts is used to describe the output power of a light. For example, a 60 watt light bulb at 40% efficiency outputs 24 watts of light energy.

**Luminous flux** is the quantity of light energy emitted by a light source in a unit time interval. It is measured in **lumens**. This measurement is used frequently by manufacturers of light fixtures and light bulbs as an indicator of the output power of a light. For example, a standard 60 watt incandescent light bulb has about 800 lumens. Unlike the energy specification of light sources in watts, lumens is a direct indicator of the brightness of a light source. That is, it describes how much energy is emitted.

**Luminous intensity** is the quantity that describes how much light energy is emitted by a light source in a unit time interval in a specific direction. It is measured in **candelas**. While luminous flux indicates the overall light output, luminous intensity measures how strong a light is in a certain direction. This measurement also takes into account the volumetric shape of the light.

For example, a standard light bulb can be described as emitting light in a spherical fashion, with the light intensity being equal in all directions. To keep it simple, we will ignore the fact that a light bulb does not emit any light in the direction of its socket. If the luminous flux of this light is defined as 1000 lumens, the luminous intensity of the light can be calculated by dividing 1000 lumens by the volumetric angle of the light, in this case $4 \times \pi$ for the spherical shape of the light bulb. This yields a luminous intensity of about 80 candelas, which tells us how strong the bulb shines in any given direction. Next consider a reflector type light bulb with the same luminous flux. Its volumetric shape can be approximated by a conical sector of a sphere. Let us assume that the volumetric angle of this cone is one quarter of a sphere. The luminous intensity of this light source would yield 360 candelas. In other words, while the reflector light has the same luminous flux at 1000 lumens, its light energy is concentrated in a smaller area and therefore has more luminous intensity. In **RenderZone Plus**, all three quantities can be used to specify the output energy of lights.

**Luminance**: This quantity describes how much of the light energy that hits a surface is reflected back into the environment at a given direction per square area. Its units are measured in **candelas** per square foot or square meter.

**Illuminance**: This quantity describes the luminous flux (lumens) that hits a surface in a unit square area. Its units are **lux**, defined as one lumen per square meter, or **footcandle**, which is defined as one lumen per square foot. Illuminance is a useful measure for determining the light intensity on a unit area of a surface. For example, when designing lighting for interior spaces, building code requirements specify the light intensity on a working desk in lux or footcandle, to ensure that enough light is present on the desk surface for proper working conditions.

The Luminance and Illuminance of the surfaces in a scene can be visualized in a color coded rendering, by selecting the **Analysis** checkbox in the **RenderZone Options** dialog, as discussed in subsection 6.1.3 of this manual.
6.6.2 The Lights palette and dialogs

All currently defined light sources are displayed in the Lights palette shown in Figure 6.6.2.1. If not open, this palette can be invoked from the Palettes menu in the usual manner.

The Lights palette of RenderZone Plus is identical to that in the regular version (see section 2.10.1 in form•Z User’s Manual) of the program, except that eight different icons are used to display the light type:

= Sun
= Distant Light
= Point Light
= Cone Light
= Projector Light
= Area Light
= Custom Light
= Line Light
= Environment Light

You can create a new light by clicking in the white space in the Lights palette. After a default name appears in text edit mode you can type another name if you prefer. Then you click in the white space again, which exits the name edit mode and displays the name with an icon in front of it. The default type when new lights are created is distant. This can be changed by double clicking on its name to invoke the Light Parameters dialog (Figure 6.6.2.3).

New lights can also be created and defined, existing lights can be deleted, copied, and edited through the Lights dialog, shown in Figure 6.6.2.2. It can be invoked by clicking in the “Light Name” bar of the lights palette, by option (Macintosh) or ctrl+shift (Windows) clicking in the palette, or by selecting the Lights... item from the Options menu. All the features in this dialog work as in the regular version of form•Z (see section 2.10.1 in form•Z User’s Manual).

Double clicking on a light name in the Lights palette invokes the Light Parameters dialog (Figure 6.6.2.3), which allows you to change the attributes of a light. This dialog is also invoked each time you execute the New... or Edit... commands in the Lights dialog.

Figure 6.6.2.1: The Lights palette.

Figure 6.6.2.2: The Lights dialog.
The **Light Parameters** dialog is very similar to its counterpart used in the regular version of *form-Z* where only one type of light (distant) is available (see section 2.10.3 of the *form-Z* User’s Manual). It has the same 3 tabs: **Intensity**, **Location**, and **Shadows**. In addition it has a **Parameters** tab where the options specific to each light type are shown.

**Type:** One of eight available light types may be selected from this pop up menu: **Distant**, **Point**, **Cone**, **Projector**, **Area**, **Custom**, **Line** and **Environment**.

![Light Parameters dialog](image)

### Intensity tab

This group of options allows you to set up the intensity of lights and their falloff through more or less detailed methods, depending on the type of light.

**Falloff:** If the current light is a **distant** or **environment** light, this menu does not apply and is dimmed. Distant and environment lights always use constant falloff. For all other types, this menu allows you to select how the intensity of the light will decrease as the distance of the light from an illuminated surface increases. Five options are available, all of which use different formulas to compute the light intensity of a surface pixel when illuminated by a light source.

When the **Simple** intensity option is chosen, the radius parameter (point, custom, area, line lights) or the distance from the light’s origin to the light’s center of interest (cone, projector lights) is taken into account for the intensity calculation. This distance will be identified as $r$ in the description below.

**Constant:** The light intensity of a surface pixel is the same anywhere in the scene.

**Linear Clamped:** The light intensity of a surface pixel is the same as the value entered in the **Simple** field, if the surface pixel is at a distance of $r$ from the light source. If the pixel is further away, the intensity decreases linearly. For example, if the **Simple** intensity is set to 100% and the pixel is twice as far as $r$, the intensity is 50% ($100 \% / 2$). If the pixel is eight times as far way as $r$, the intensity is 12.5% ($100 \% / 8$). If the pixel is closer than $r$, the intensity increases approximately linearly, but will not exceed a maximum value. For example, if the pixel is half as far away as $r$, the intensity is about 200% ($100 \% / 0.5$). However, as the pixel gets closer to the light source, the intensity does not increase linearly anymore and is clamped against a maximum value. This will guarantees, that a pixel very close to the light source will not get an unreasonably high illumination and avoids unexpected overexposure in a scene.
**Square Clamped**: The light intensity of a surface pixel is the same as the value entered in the **Simple** field, if the surface pixel is at a distance of $r$ from the light source. If the pixel is further away, the intensity decreases with the square of the distance. For example, if the **Simple** intensity is set to 100% and the pixel is twice as far as $r$, the intensity is 25% ($100\% / (2 * 2)$). If the pixel is eight times as far away as $r$, the intensity is 1.5625% ($100\% / (8 * 8)$). If the pixel is closer than $r$, the intensity increases approximately with the square of the distance, but will not exceed a maximum value. For example, if the pixel is half as far away as $r$, the intensity is about 400% ($100\% / (0.5 * 0.5)$). However, as the pixel gets closer to the light source, the intensity does not increase with the square anymore and is clamped against a maximum value. This will guarantees, that a pixel very close to the light source will not get an unreasonably high illumination and avoids unexpected overexposure in a scene. Square falloff is a physically accurate representation of a light's illumination as long as the pixel intensity is not clamped at a closed range, as described above.

**Linear Unclamped**: This is the same as the **Linear Clamped** option, with the exception, that the intensity at pixels which are closer than $r$ away from the light source is not clamped. Therefore these pixels may get overexposed easily. If this is not desirable the **Linear Clamped** option should be used.

**Square Unclamped**: This is the same as the **Square Clamped** option, with the exception, that the intensity at pixels which are closer than $r$ away from the light source is not clamped. Therefore these pixels may get overexposed easily. If this is not desirable the **Square Clamped** option should be used. **Square Unclamped** falloff is the physically most accurate representation of a light’s illumination.

When the **Accurate** intensity option is chosen, the radius or distance parameters of a light are not used to determine the intensity. The **Photometric** or **Radiometric** intensity in the **Accurate Intensity** option serves as the basis for the illumination calculation. When rendered, the computer screen serves as the photo film in a camera which is exposed by the light source. The film's sensitivity is set in such a way, that an interior scene with a number of average brightness light bulbs is well exposed. Under and over exposed images can be corrected with the **Exposure Correction** option as described in Section 6.1.10. Note, that the **Falloff** menu is available for both, **Simple** and **Accurate** intensity. For a physically accurate exposure, it is necessary to choose the **Square Unclamped** falloff option. However, to generate visually pleasing images physically accurate illumination is often times not desirable and therefore the **Accurate** intensity option may very well be combined with different falloff methods.

**Simple**: When this option is selected, the value entered in the **Brightness** field determines the intensity of the light, expressed as a percentage. While this parameter is entered in the same way for all types of lights, its application to the different types varies, as follows:

- **Distant, environment**: These lights always have a constant intensity and the intensity value is applied uniformly.
• **Point, custom:** When **Constant** is selected for this light from the **Falloff** menu, its intensity remains uniform throughout. If a **Linear** or **Square** falloff is selected, the value entered in the **Simple** field is applied to the points of the spherical surface defined by the **Location** of the origin of the light and the value entered for **Radius** (see section 6.6.3). The light intensity increases between the spherical surface and the origin of the light. It decreases as it moves away from the spherical surface (at distances from the light origin greater than the **Radius**). The degree at which the light intensity increases or decreases depends on whether **Linear** or **Square** falloff has been selected. The distribution for point light is uniform in all directions, while the intensity distribution of a custom light is dependent on the parameters defined in the **Intensity Distribution** dialog (see Figure 6.6.6.5). This is discussed in detail in subsection 6.6.6.

• **Cone, projector:** When **Constant** falloff is selected for these lights, their intensity remains uniform throughout. If **Linear** or **Square** is selected, the value entered for **Simple** is applied at the **Center Of Interest** (see next section). The intensity increases between the light’s origin and the center of interest, and decreases beyond the center of interest at a rate dependent on whether **Linear** or **Square** falloff has been selected.

• **Area, line:** When **Constant** is selected for these lights from the **Falloff** pop up menu, their intensity remains uniform throughout. If a **Linear** or **Square** falloff is selected, the value entered in the **Simple** field is applied to all shaded points that have the same distance as the **Radius** (see section 6.6.3). The light intensity increases between these points and the light source. It decreases as it moves away from these points (at distances from the light source greater than **Radius**). How the light intensity increases or decreases depends on whether **Linear** or **Square** falloff is on.

Note that when a surface is in the area where the intensity of a light increases, there is always a risk of overexposure. If such an effect is not desirable, then a **Constant** intensity should be used.

**Accurate...:** When this option is selected, the intensity of the light is described through physically accurate units. The **Accurate Intensity** parameters are shown in the area below the option. They vary between the different light types. The **Accurate Intensity** parameters for distant lights are unique. The other seven types of light share similar parameters. However, the **Brightness** parameter for cone and point lights can be defined using either lumens or candelas. Only lumens can be used to set the brightness of area, line, environment, and custom lights. Achieving accurate intensity for the different types of lights is discussed in more detail in subsequent sections.
**Location tab**

This group of options contain the **Origin** and **Center Of interest** of a light. It is discussed in more detail in section 2.10.3.

**Shadows tab**

This group of options specify if the light casts shadow and of what type. Shadows are cast when the **Shadows** box is checked, but additional options in other dialogs need to be also set, as discussed in detail below. A rather extensive discussion of **Shadows** can also be found in section 3.6.7, where display modes other than **RenderZone** are presented for the basic version of form-Z.

The type of shadow is set in the menu below the **Shadows** check box:

**Type:** **Soft (Mapped), Hard (Raytraced), Hard (Accelerated):** When one of these items is selected, shadows are created through shadow maps, raytracing, and a combination of raytracing and shadow maps, respectively. In the latter case, shadow maps are used to accelerate the generation of raytraced shadows, which has additional memory requirements.

**Transparent:** This option is only available when raytraced shadows are generated. When on, a light generates transparent shadows when shining through transparent surfaces. Such shadows take longer to render, even if there are no transparent surfaces in a scene. A rendering can be accelerated by turning this option off, if there are no transparent surfaces visible, or if the light does not illuminate any transparent objects. Recall that if the **Per Light** item is chosen from the **Shadows** menu in the **RenderZone Options** dialog, the **Transparency** shadow option in each individual light source is considered. If the **All Opaque** or **All Transparent** option is chosen from the menu, the **Transparency** option for each individual light is overwritten.

In form-Z **RenderZone Plus**, objects in a scene can cast shadows when the z-buffer or raytrace display method is used from the **Rendering Type** menu in the **RenderZone Options** dialog. For shadows to be cast, the following parameters must be set properly:

- The shadow attribute of an object must be on. It is on by default when a new object is created.

- The shining attribute must be turned on for at least one light (other than the ambient light). This attribute is turned on in the **Light Parameters** dialog and in the Lights palette.

- The **Shadows** option must be selected for this light in the **Light Parameters** dialog, and the type of shadow (**Soft (Mapped), Hard (Raytraced) or Hard (Accelerated)**) must be selected from the **Shadows** pop up menu. More than one light can cast shadows at the same time, but only those lights whose shining and shadow parameters are on will.
• The Shadows option in the RenderZone Options dialog must be turned on when the RenderZone* command is executed. Recall that this dialog can be invoked directly from the RenderZone* menu item before executing the operation.

These requirements are the same as those in the regular version of form•Z, as are the Quality, Softness, Tolerance, Resolution, and Limit Map To options that must also be set in the Shadows tab when Soft shadows are used. However, note that the Limit Map To options apply to distant lights only and have no effect on other lights. That is, all soft shadow casting objects will cast shadows from the other lights, regardless of which Limit Map To option is selected and regardless of whether they are in the view to be rendered or not. We strongly recommend that you read the section titled “Warnings” in section 3.6.7 of the form•Z User’s Manual that refers to Soft shadows, and take a look at the examples in Figure 3.6.7.1. Given their importance, we also summarize them here.

When using Soft shadows, if your scene contains very large objects such as ground planes that do not cast shadows, or objects which are outside the view and do not cast shadows onto the visible objects, you should turn their shadow attribute off. This will result in crisper shadows.

The shadow maps used for the generation of Soft shadows can be very memory intensive, and should be used conservatively if your machine has a relatively limited amount of memory. For each pixel in a shadow map, 4 bytes of storage is required. For example, a shadow map of 500 x 500 pixels requires 1 Megabyte of memory. Assuming a constant shadow map size, cone and projector lights generate the best shadow quality, compared to the distant, point, and custom lights. Point and custom light shadow maps have the highest memory requirements, which is six times more than that required for cone, projector, or distant lights. Given that a separate shadow map needs to be produced for each light, using too many shadow-casting lights increases the risk of running out of memory. Note that area and line lights do not offer the option to use shadow maps. They only cast raytraced shadows.

**Hard (Raytraced)** shadows do not use shadow maps, but rather ray tracing procedures. Distant, point, custom, cone, and projector lights produce crisp raytraced shadows. Area and line lights create soft raytraced shadows. Their memory requirements are minimal, but they generally require more processing time.

**Hard (Accelerated)** shadows are raytraced but also use shadow maps to speed up the raytracing procedures. Their memory requirements are the same as soft shadows.

The type of shadow used in a rendering scene depends on the effects a user wishes to achieve. Shadows are attributes of the light, and their type is set independently for each light, which allows you to mix the two types of shadows within the same rendering scene.
Parameters tab

The content of this tab varies according to the light type selected and is discussed with each type in the next sections.

Options outside the tabs

Most of the options outside the tabs, on the right side of the Light Parameters dialog, also apply to the Shaded Render* mode, available in the basic version of form•Z and are discussed in section 2.10.3. They are Preview, the Invisible, Visible, Ghosted menu, Locked, and Shining. The following two options are only available in RenderZone Plus.

Lens Flare: This option determines whether the light is able to cause lens flares during the RenderZone postprocess lens flare operation, which is discussed in section 6.1.10 of the RenderZone Plus User’s manual.

Glow: When this option is on, the light displays a glowing volume when rendered in the z-buffer and raytrace rendering modes. It is discussed in more detail in section 6.6.4 of this volume.

Glow Options...: This button invokes the Glow Options dialog, which is discussed in section 6.6.4 of this volume.
6.6.3 Distant, point, cone, and projector lights

The types of lights available in form•Z RenderZone Plus may be viewed under two categories. First is the category of simpler lights, which assume idealized sources that distribute light more or less uniformly. Second are four types of more complex lights: area, line, environment, and custom. In the first category are the distant, cone, point, and projector lights which are discussed in this section. The area, line, environment, and custom lights are discussed in separate sections.

Distant lights

Selecting Distant for Type creates a distant light whose positional definition is completed in the X, Y, and Z values for Origin and Center Of Interest.

Intensity tab

When Distant is selected in the Type menu, the Accurate options in the Intensity tab are as shown in Figure 6.6.3.1.

Output Power: For distant lights, the intensity is not defined as an output quantity of the source, but as a quantity which indicates how much energy is received at a square unit on a surface perpendicular to the light. In most cases, distant lights are used to simulate the sun light. Instead of specifying the exact distance of the sun from the scene and the exact output power of the sun, it is more meaningful to specify the effect of this type of light on a surface.

Parameters tab

Apply Atmospheric Light: When this is selected, indirect light in the atmosphere, which for example reflects off the clouds, is added to the scene.

Consider, for example, a room with a window. Even if there is no light shining directly into the room, the environment still submits a considerable amount of light into the room. Figure 6.6.3.3 shows a scene rendered with a distant light using atmospheric effects. The distant light does not shine directly into the room. Illumination is generated only by the atmospheric light. Note, that the atmospheric light is especially effective when used in a final gather rendering. This is described in more detail in section 6.1.6.
**Type:** This pop up menu contains two options, **Simple** and **Acurate**.

**Simple:** When this option is selected, a simple atmospheric light is applied. It is generated by allowing a number of additional light rays to illuminate the scene, which are randomly distributed in the hemisphere.

**Color:** This is the color of the atmospheric light cast onto the scene.

**Intensity:** The light intensity of all combined rays is entered in this field. This is a simple intensity factor, expressed as a percentage in the same fashion as the **Simple Intensity** is defined in the **Light Parameters** dialog. Setting the intensity parameter to a proper value depends on the scene rendered. If the scene depicts an exterior, for example a building sitting on a site, the intensity can be kept around 100%, as the entire hemisphere contributes to the illumination of the building. However, when rendering an interior scene and the light is shining through a comparatively small window, only a small portion of the atmospheric light actually reaches the interior. Therefore, the intensity must be increased. By how much needs to be determined experimentally, by executing a few fast preview renderings with a low number of samples (see below). It may not be uncommon to increase the intensity above 3000%.

**# Of Samples:** The number of randomly directed light rays which illuminate the scene is entered in this field. The higher the number, the better the quality of the illumination will be, but the slower the rendering will execute. For very high quality image it may be necessary to enter values of 1000 and above. To test the brightness of the illumination, it is advisable to start with a lower **# Of Samples** parameter, such as 50, and only enter a high number for the final rendering. A scene rendered with 50 and 1000 as the **# Of Samples** parameter is shown in Figure 6.6.3.3.

![Figure 6.6.3.3](image)

**Figure 6.6.3.3:** A scene rendered with (a) 50 and (b) 1000 **# Of Samples**.
**Noise**: To reduce the artifacts resulting from a lower # Of Samples parameter, it is possible to add artificial noise to the shading generated by the simple atmospheric light. The noise manifests itself as a subtle pattern of light and dark spots, which will hide the hard transition from light to shadow. A scene with a low number of samples with noise added to it is shown in Figure 6.6.3.4.

![Figure 6.6.3.4: A scene rendered with noise added to the simple atmospheric light.](image)

**Accurate**: When this option is selected, the atmospheric light is described more accurately. Real world values are used to define the light color and intensity.

The color of the accurate atmospheric light may be defined in two different ways. If the **Color Temperature** option is selected, the temperature value in degrees Kelvin is entered in the text field. Standard values can be selected from the pop up menu next to the text field, which contains the settings for **Light Blue**, **Medium Blue**, and **Dark Blue** skies. If the **Color box** option is selected, the atmospheric light color can be set directly through the standard color picker, invoked by clicking on the color box.

**Cloud Coverage**: From this menu, **Overcast**, **Mixed**, or **Clear** may be selected. **Overcast** generates a less uniformly directed light, whereas **Clear** orients the light more in the direction of its source. **Mixed** generates an intermediate result.

**Sky Luminance**: This value indicates the brightness of the sky directly above in **Candelas** per square foot, or square meter in a metric project.

**Quality**: This slider bar controls the depth of the atmospheric light simulation. The higher the quality, the longer it takes to process this light during the execution of the radiosity solution.

**Noise**: This option is the same as for the Simple atmospheric light.
Point lights

**Intensity tab** (Figure 6.6.3.5)

**Radiometric**: When this option is selected, the total **Output Power** of the light is defined in **watts**, using two parameters.

**Input Power**: This parameter value specifies the energy supplied to the source in **Watts**.

**Output Efficiency**: This parameter, expressed as a percentage, determines how efficient the light source is.

When multiplying the two parameters, the output energy of the light source can be calculated. It is common for light bulb manufacturers to specify the input energy of a light bulb. For example, a 60 Watt incandescent bulb consumes 60 Watts of electric energy, but outputs much less light energy. The efficiency of incandescent bulbs is around 40%, which means that a 60 Watt bulb creates about 24 Watts of light energy, with the rest being mostly converted to thermal energy.

**Photometric**: When this option is selected, the light energy is specified in **Lumens** or **Candelas**. Which unit is used can be selected from the **Brightness** pop up menu. Recall that **Candelas** determine the light intensity at any given angle of the source. It is a unit which indicates the brightness of a light source at a given point. **Lumens** specifies the overall intensity of the light source and is a measure of the strength of the light. When specifying the light intensity in photometric units, it is also possible to determine the color temperature of the light.

**Color Temperature**: When this option is selected a value can be entered in the text field, which indicates how hot the light burns. This temperature, measured in degrees Kelvin, is converted into a color, which is mixed with the light’s color selected in the **Light Parameters** dialog. Low color temperatures generate an orange color. The hotter the temperature, the closer to white the color of the light is. Color temperatures for some common lamp types are listed in subsection 6.6.8.

**Brightness**: This value is expressed in lumens or candelas as discussed above. Note that the higher the value is the brighter the light is.
Parameters tab (Figure 6.6.3.6)

**Radius**: The value entered in this field determines a spheric surface, centered at the origin of the light, on which the light’s intensity takes the value specified in the **Intensity** field. This is also the spherical surface shown when point lights are displayed in wire frame.

Note that the **Center of Interest** fields have no meaning for point lights and are dimmed.

A scene with a point light is shown in Figure 6.6.3.7. In this example the point light is used to simulate a hanging light fixture. Figure 6.6.3.8 shows the **Light Parameters** dialog for that light. The falloff is set to **Square Clamped**, which relates to the behavior of light in the real world. Note that the radius parameter is set to 6'-0", which is approximately the distance of the light fixture from the floor. Setting this radius means that the floor receives light at 100% intensity. Surfaces closer than 6’, such as the table top, will get more light, surfaces further away will get less. You can recreate this rendering and look at the light parameters in more detail by loading sample file “Point.fmz.”

![Figure 6.6.3.7: A scene rendered with a point light.](image)

![Figure 6.6.3.8: The **Light Parameters** dialog for a point light with the (a) **Intensity** and (b) **Parameters** tab open.](image)
Cone lights

When **Cone** is selected in the **Type** menu, the **Cone Light** options are presented as shown in Figures 6.6.3.9 and 6.6.3.10.

**Intensity tab**

The **Accurate Intensity** options for a cone light are the same as those for a point light.

**Parameters tab**

The two parameters in this group of options define the physical shape of the cone light, which consists of two cones, an outer and an inner cone. The inner cone defines the volume where the light intensity remains constant relative to its distance from the axis of the cone. From the inner to the outer cone, the light intensity decreases, and fades completely at the outer cone.

**Outer Angle, Inner Angle**: These slider parameters determine the angle for the outer and inner cones of a cone light, and range from 0° to 180°. The two are interdependent, but the inner angle cannot be greater than the outer angle, and it cannot be less than half of the outer angle. If one slider is set to a value that violates the above conditions, the other is adjusted automatically. The values entered for the outer and inner angles and the coordinates for the **Location** and the **Center Of Interest** are also used to construct the shape of a cone light displayed in wire frame.
A scene with a cone light is shown in Figure 6.6.3.11. In this example the cone light is used to simulate a desk lamp. Figure 6.6.3.12 shows the **Light Parameters** dialog for that light. The falloff is set to **Square**, which relates to the behavior of light in the real world.

Note that the **Center Of Interest** of the light, drawn with an arrow in the project window, is located approximately on the tabletop surface. At that point, the cone light illuminates the surface at the intensity set in the **Simple** intensity field, in this case 150%. As with the distant light example, an additional fill light is used to improve the shading of areas not illuminated enough. You can recreate this rendering and look at the light parameters by loading sample file “Cone.fmz.”

**Figure 6.6.3.11:** A scene rendered with a cone light.

**Projector lights**

**Intensity tab** (Figure 6.6.3.13)

This is identical to the one for point lights, except that **Photometric Brightness** is always set in **Lumens**.

**Figure 6.6.3.12:** The **Light Parameters** dialog for a cone light with the (a) **Intensity** and (b) **Parameters** tab open.

**Figure 6.6.3.13:** The **Accurate Intensity** options for projector lights.
Parameters tab

The parameters in this tab define the shape and orientation of the pyramid associated with this light and the texture map through which the light is projected.

**Angle:** This parameter defines the angle of the projector pyramid. It may vary from 0 to 180°.

**Spin:** This sets the rotation of the image the light projects. The rotation is about the axis of the light, defined by its origin and center of interest. With a spin angle of 0 degrees the vertical direction of the texture points upward towards the world Z axis. In wireframe display, the up direction of the projector light is indicated by a small arrow from the center of interest along the vertical direction of the texture map.

To the right of the Projector Light box, an image is displayed. Clicking on it invokes the Projector Map Options dialog that can be used to load a new texture map. The map may be of any image format supported by form-Z.

A scene with a projector light is shown in Figure 6.6.3.15. In this example the projector light is used to simulate a slide projector. Figure 6.6.3.16 shows the Light Parameters dialog for that light. As with the distant light example, an additional fill light is used to add improved shading to areas not illuminated by the projector light. You can recreate this rendering and look at the light parameters in more detail by loading sample file “Projector.fmz.”

**Figure 6.6.3.14:** The Projector Light parameters.

**Figure 6.6.3.15:** A scene rendered with a projector light.

**Figure 6.6.3.16:** The Light Parameters dialog for a projector light with the (a) Intensity and (b) Parameters tab open.
6.6.4 Making lights glow

All lights have the ability to glow in a rendered scene. This is done by selecting the Glow option in the Light Parameters dialog.

**Glow**: When this option is on, the light displays a glowing volume when rendered in the z-buffer and raytrace rendering modes. For point and custom lights, this volume is a sphere. Cone and projector lights show a conical volume. The volume of the point light has a maximum brightness at the center of the sphere, and diminishes toward its perimeter. The volume of the cone and projector lights has maximum brightness at the origin of the light. The brightness diminishes along the direction of the light, and disappears at the light’s center of interest. The brightness also diminishes in the area between the inner and outer cones. Area and line lights are represented by a spherical glowing volume if the Simple glow is used. For Accurate glow, the shape of the glowing volume follows the shape of the geometry associated with the area or line light. Distant lights can only be used in combination with the Accurate glow option. Examples of glowing lights with the Simple option are shown in Figure 6.6.4.1. All lights in the example have a white color and are shown on a black background.

The parameters for the light glow can be set in the Glow Options dialog (Figure 6.6.4.2), which is invoked by clicking on the Glow Options... button. There are two parts in this dialog: Simple and Accurate.
**Simple**: This group of options contains relatively simple methods for specifying the glow parameters.

**Intensity**: This slider parameter controls the brightness of the glow. At 0%, the glow is not visible. At 100%, the glowing volume shows maximum brightness at the light’s origin.

**Falloff**: This parameter determines how quickly the brightness of the glow diminishes from the light origin toward the perimeter of the glowing volume. **Linear** falloff gives an even distribution. With **Square** and **Cubic** falloff the brightness diminishes more quickly.

**Glow Area**: From this category, one of two options may be selected to determine the area and distance of the glow.

- **Use Light’s Radius** (point, area, custom, line light), **Use Light’s Distance** (cone, projector light): When this option is selected, the radius of the glow for point lights is determined by the **Radius** parameter in the **Light Parameters** dialog. If the light is a cone or projector light, the length of the glowing cone is determined by the distance between the light’s origin and its center of interest.

- **Radius** (point, area, custom, line light), **Distance** (cone, projector light): When this option is selected, the radius for the glow of point lights or the length of the glowing cone for cone and projector lights is determined by the value entered in the numeric field next to it.

Accurate volumetric glowing effects are achieved by using an alternative algorithm, which simulates the interaction of light and “dust” particles in the vicinity of the light source. This not only leads to a physically accurate illumination of these particles, it also generates proper volumetric shadows. Examples of volumetric glows specified with the simple and the accurate methods are shown in Figure 6.6.4.3.

*Figure 6.6.4.3:* Examples of (a) simple and (b) accurate light glow.
**Accurate:** This section contains the parameters for accurate light glow.

**Intensity:** This parameter determines how bright the glowing light volume appears.

**Attenuation:** This value, expressed as a percentage, determines how fast the light volume decreases as it travels farther from the source. The higher the percentage, the faster the volume decreases. In other words, lights which are far away from the viewer show less glow than lights closer to the viewer if the Attenuation value is set to a high value. If the Attenuation value is 0, lights always glow, no matter how far away from the viewer they are.

**Max Distance:** If a light uses the Constant falloff option for its simple intensity specification, this parameter determines the maximum distance from the origin of the light at which glow is generated. This is necessary, since Constant falloff causes light to be equally intense at any distance from the source. If Linear or Square falloff or accurate intensity are used, the max distance parameter is not used, since glow is not generated at a distance from which the light intensity is too weak to create any visible illumination.

**Quality:** A quality level between 1 and 10 can be selected from this pop up menu. Level 1 produces fast results, but may show visible artifacts in the quality of volumetric shadows. Level 10 will create the most accurate results, but also takes the longest to compute. In general, accurate light glow is at least one order of magnitude slower than simple glow. Therefore, it is advisable to choose the minimum quality level possible for acceptable results in order to optimize rendering speed.

**Noise:** If this option is selected, the dust in which the light travels will show irregular patterns. Otherwise the light glow will appear smooth.

**Size:** The value entered in this field determines the size of the irregularities. This value is interpreted in the same fashion as the Solid Textures Size parameter in the Texture Map Controls dialog, which determines the size of the pattern for solid shaders such as Marble or Mist.

**Type:** This option determines the quality of the noise. Three levels are available: Simple, Better, and Best. They are the same as in all the shaders with random patterns.

**# Of Impulses:** As with the Type menu, this numeric field is the same as in all the shaders with random patterns.

**Contrast:** This parameter determines the contrast between dense and thin areas of the light glow. An example with low and high contrast is shown in Figure 6.6.4.4.

**Detail:** Low values for detail will keep the irregular pattern smooth, whereas a high value will cause “curly” noise. An example with low and high detail is shown in Figure 6.6.4.5.
Some advice on using accurate glow

**Accurate glow and rendering speed:** Accurate glow can significantly slow down a rendering. It should only be used on those light sources which would contribute a visible glowing effect to the image. The **Quality** option should be used to optimize the speed at which glowing lights are rendered. For fastest results, the lowest quality setting, which produces acceptable images should be chosen.

When turning on the **Noise** option, rendering times will further increase. **Simple** and **Better** Noise types are usually enough to create acceptable noise patterns. The **Best** Noise type should be used with caution. While it generates noise patterns without any recognizable regularities, it is a very expensive option, especially when the number of impulses is set to a higher value.

**Accurate glow and constant light falloff:** As mentioned above, accurate glow simulates the illumination of small dust particles in the scene. If combined with **Constant** light falloff, each dust particle will receive the same illumination value. This usually leads to images with a uniformly bright glare. In the real world, light intensity falls off with the square of the distance that the light travels. In order to simulate realistic glowing results, the **Square** light falloff should therefore be used.
6.6.5 Area and line lights

Cone, point, projector, custom, and distant lights are idealized sources, assuming that the light originates from a single, infinitely small point. While this is adequate to simulate, for example, an incandescent light bulb, other light fixtures such as neon tubes are hard to recreate with these light types. Area and line lights are associated with physical objects whose surfaces or segments are emitting light. For example, it is possible to model a curved neon tube by using the sweep tool and a circular source shape, then turning the resulting object into an area light. This light source behaves exactly like a glowing neon tube. A scene rendered with such a light is shown in Figure 6.6.5.1.

Creating an area or line light

You create an area/line light as follows:

• Create an object that will become a light. That is, this object represents the shape of the area or line light.
• Preselect the object with the Pick tool.
• Create a new light. Recall that you do this by either clicking in the Lights palette under its last entry, or by clicking on the New... button in the Lights dialog.
• When the Light Parameters dialog is invoked, select Area or Line from the Type pop up menu. This will cause the Area or Line Light options to be displayed in the Parameters tab of the dialog, as shown in Figure 6.6.5.2.
• Select the desired settings in the Area/Line Light group of options, whose functionality is discussed on the following page.
• Set additional parameters in the Light Parameters dialog if desired and click on OK.

Figure 6.6.5.1: A scene rendered with an area light.

Figure 6.6.5.2: The Light Parameters dialog and (a) the Area Light and (b) the Line Light parameters.
After the definition of an area/line light is completed its name is displayed in the Lights palette. To edit and change its parameters, double click on its name in the Lights palette. Any previously defined light of a different type can also be changed to an area/line light by double clicking on its name, provided an object is also preselected when editing the light. The object used for an area light can be a solid, a surface object, or a surface solid (double-sided surface). It cannot be an open vector line, a symbol, or plain text. Typically, the object chosen for a line light is a single face with an open or closed surface. However, it is also possible to select an object with more than one face, in which case only the first face of the object is used as the shape for the line light. Symbol or plain text objects are not valid selections for a line light.

Once an object has been converted into an area or line light, it is automatically deleted from the Objects palette. That is, no more modeling operations can be performed on the object defining the light. In order to retrieve the object, the area or line light must be converted to another light type. This can be done by executing the following steps:

- Double click on the light’s name in the Lights palette to invoke the Light Parameters dialog.
- In this dialog, select a type other than Area or Line from the Type menu.
- After clicking OK, the new light type is drawn in place of the area or line light, and the object reappears as a regular form•Z object. The object preserves its original color, and is placed on the active layer.

Surfaces of an area light shine only in the positive direction, as indicated by the direction arrow. For example, if a surface object, which consists of a single face, is defined as an area light, the part of the scene which is behind the surface is not illuminated directly by this light.

**Intensity tab**

As with the other light types, the intensity of Area/Line lights can be specified as Simple or Accurate.

The Accurate Intensity options for the Area and Line lights are almost identical to those used for Point and Cone lights. The only difference is that, in the former dialogs, the Brightness option can only be set in lumens. In these dialogs the pop up menu next to Brightness is not available. Candela values are only meaningful if the light source has an equal intensity in all directions. This is not the case with area and line lights, since the directions and sizes of the faces or segments determine the intensity of the light at a given point in space.
**Shadows tab**

As with all lights, the area and line lights can cast shadows. Note that unlike other light sources, raytraced shadows from area or line lights appear fuzzy. **Soft (mapped)** shadows from area and line lights are generally faster than raytraced shadows but also less accurate.

**Parameters tab**

**Render Light Object**: When this option is on, the object of the area light is rendered in the image. Otherwise, the object is invisible. However, even if the object is invisible, the energy emitted by the area light is processed.

Recall that, in wire frame, an area light and its object are displayed when its visibility attribute in the **Lights** palette is on. In addition, its directions may be also displayed if the **Show Direction Of Area Lights** option in the **Interactive** tab of all interactive rendering dialogs is on. The direction of the light leaving all surfaces is indicated by an arrow starting at the center of each face and pointing outward. An area light drawn in wire frame is shown in Figure 6.6.5.4.

**Ignore Light Object Color**: When this option is on, the area light emits light in the color shown in the color box. When off, the light color is filtered through the color of the surface styles used by each face of the light object.

For example, a cube with different colors on each face, once converted into a white area light, illuminates the scene with six different colors. Each color is most intense in the direction of the face with the color. Note that this is always a solid uniform color, regardless of the color shader of the surface style used by the faces. That is, if a texture map is used as the color shader, an average color is determined from the texture. In other words, the area light does not project the color pattern, but always uses a plain color.

**Quality**: The quality with which an area light is rendered can be set by the slider bar. Note that a high quality setting may lead to significantly longer rendering times.

**Radius**: This parameter determines the distance from which points are shaded at full intensity if the **Simple Intensity** option is used.

When a **Line** light is selected in the **Light Parameters** dialog (Figure 6.6.5.2(b)), two parameters need to be set: **Radius** and **Quality**. They both work as for the area lights (see above).
A scene with area lights is shown in Figure 6.6.5.5. In this example the area lights are used to simulate diffuse day light entering through the window. Figure 6.6.5.6 shows the Light Parameters dialog for one of the lights. The intensity is set to 200%. Since the light intensity is emitted from a large area, instead of a single point, the light intensity has been increased to sufficiently illuminate the interior. Quality is turned on and set to a high value. This creates very accurate shading, but also increases the rendering time noticeably. You can try this rendering by opening sample file “Area.fmz.”

A scene with a line light, which is used to simulate a neon tube, is shown in Figure 6.6.5.7. The settings for this light are shown in Figure 6.6.5.8. The falloff is set to Square Clamped, which relates to the behavior of light in the real world. As with the area light, the intensity is set to a value higher than the default 100%, since the light is emitted from a linear element rather than a single point. The Radius parameter is set to 6'-0", which is approximately the distance of the light fixture from the floor. Thus the floor receives light at 200% intensity. Surfaces closer than 6’, such as the table top, will get more light, surfaces further away will get less. You can recreate this rendering by loading sample file “Line.fmz.”
6.6.6 Custom lights

In conventional computer renderings, lights are assumed to be distributing their rays uniformly. For example, a point light, which is typically used to represent a bulb hanging in a room is assumed to emit rays of equal intensity in all directions. However, light sources do not really work this way in the real world. For example, a bulb has a higher intensity in the direction of the socket, and its intensity falls off toward the socket, and it is zero directly behind the socket. The custom light type is specifically intended to take into consideration these real world “irregularities.” The custom light is essentially a method which allows us to represent variable intensities in different directions about a light source.

The custom light representation

The varying intensities of a custom light can be visualized and represented as one or more curves called distribution curves. Such a curve is drawn on a plane relative to a radial grid, as shown in Figure 6.6.6.1. The curve is defined by a set of points, each of which is defined by its angle from the horizontal orientation of the radial grid and its distance from the center of the grid. The distance represents the intensity of the light at the direction where the point is positioned. A custom light is completely defined by a set of planes, that intersect on the axis of the light. The axis is defined by the origin and the center of interest of the light. This is illustrated in Figure 6.6.6.2.

The distribution curves and their planes are displayed when the intensities of a custom light are defined. When a custom light is displayed on the screen, it is represented as shown in Figure 6.6.6.3. It is represented as a sphere, by three circles that lie on perpendicular planes. Two of these planes intersect at the axis of the light, which has a direction determined by the origin and center of interest of the light. These points and their direction are represented by a line that connects them and an arrow head at the position of the center of interest. Another line that starts at the origin and is perpendicular to the axis, represents the zero angle position relative to which the angles of the planes used for the representation of the light intensities are measured. The zero angle line also ends at an arrow head which is smaller than the direction’s arrow head.
Creating a custom light and setting its parameters

A custom light is generated as with all the other lights (except the area light). Clicking under the last entry in the Lights palette or clicking on the **New...** button in the **Lights** dialog invokes the **Light Parameters** dialog. In this dialog, select *Custom* from the **Type** pop up menu, which causes custom light specific options to appear in the **Parameters** tab (Figure 6.6.6.4).

![Light Parameters dialog](image)

*Figure 6.6.6.4:* The **Light Parameters** dialog when a custom light is selected.

**Parameters tab**

**Radius:** As with the point light, the value typed in this field determines a spheric surface, centered at the origin **Location** of the light, on which the light’s intensity takes the value specified in the **Intensity:** **Simple** field. This is also the spherical surface shown when custom lights are displayed in wire frame.

**Spin:** This slider bar is used to set the spin angle, which is also displayed in the edit field next to it. The spin angle represents a rotation of the light about its axis and is measured relative to the zero angle line.
Setting the intensities of a custom light and their distribution

Unlike the other lights, the custom light requires that both its intensity and the distribution of that intensity be defined, which is done in two different ways. The accurate intensity options are shown in the **Intensity** tab and are identical to those for a point light (Figure 6.6.3.5).

Clicking on the **Intensity Distribution...** button in the **Parameters** tab invokes the **Intensity Distribution** dialog shown in Figure 6.6.6.5. This dialog is used to display a set of distribution curves representing the intensities of the custom light in different directions.

![Intensity Distribution dialog](image)

*Figure 6.6.6.5: The Intensity Distribution dialog.*

The **Intensity Distribution** dialog is a preview dialog. Its left portion contains a graphic area that displays the distribution curves of the custom light. A set of default curves are initially generated by the program, which vary depending on the selections from the **Symmetry** and **Hemisphere** pop up menus (see below). The distribution curves can be edited and changed according to the desired intensities.

The following conventions are used when displaying the distribution curves in the preview area:

- One of the distribution curves is the **active curve** and is displayed in black. There can be only one active curve at a time. When the **Show Grid** option is on, the grid of the active curve is also displayed.
- When there is more than one distribution curve, the inactive curves are displayed in the ghost color of the project (default is gray).
- One of the points of the active distribution curve is the **active point** and is displayed in the highlight color of the project (default is red).
- All the points of a curve that are not active are displayed in black.

The tools under the preview area can be used to manipulate the displayed view as in the other preview displays (see section 2.1.6 in the form-Z User’s Manual).
The Arrow tool can be used to execute the following graphic operations:

- Clicking on a distribution curve makes it active. If the **Show Grid** option is on, the displayed grid switches to that of the new active curve.
- Clicking on a point of the active distribution curve makes it the active point.
- Clicking on the active curve while pressing the **option** (Macintosh) or **shift+ctrl** (Windows) keys inserts a new point at the position of the click.
- Clicking on a point while pressing the **control** (both platforms) key deletes the point. However, note that the endpoints of a distribution curve cannot be deleted.
- Clicking on a point and dragging it moves it along the plane of the distribution curve.

The active curve and active point affect the values displayed in the fields on the right portion of the **Intensity Distribution** dialog. They are also the entities to which changes made through numeric input are applied. The options to the right of the dialog are as follows:

**Symmetry**: One of four distinct symmetry patterns is selected from this pop up menu. The symmetry pattern establishes how the distribution curve planes will be used to completely describe the intensities of a custom light. Each symmetry pattern has its own minimum plane requirements, thus, each time a different symmetry pattern is selected a warning message is issued. After responding **OK**, the program proceeds and generates the default set of planes for the particular symmetry, as follows:

**None**: When this item is selected, which does not actually involve any symmetry, the light distribution requires at least three distribution curves. The first must be on a plane at 0 degrees. The second can be at any angle. The third must be at a plane angle larger than 180 degrees. When first selected, four planes are created automatically, at 0, 90, 180 and 270 degrees around the light direction.

**Axial**: When this item is selected, the light distribution requires only one distribution curve. The angle of the plane of the curve is always 0 and cannot be edited. This option produces a description of the light which is of a rotational symmetry. In other words this is equivalent to an object of revolution.

**Quadrant**: When this item is selected, the light distribution is symmetrical in each quadrant and requires at least two distribution curves, one at 0 and the other at 90 degrees. When first selected, curves at these two plane angles are created automatically.
**Plane**: When this item is selected, the light distribution is symmetrical relative to the 0 - 180 degree plane and requires at least two distribution curves, one at 0 and the other at 180 degrees. When first selected, curves at these two plane angles are created automatically.

Light distribution curves for all four symmetry options are illustrated in Figure 6.6.6.6.

![Figure 6.6.6.6: Distribution curves for the (a) None, (b) Axial, (c) Quadrant, and (d) Plane symmetry options.](image)

**Hemisphere**: One of three mutually exclusive options is selected from this pop up menu. The selection determines the extent of the light relative to its hemispheric plane. This is the plane that is perpendicular to the axis of the light and passes through its origin. It splits the sphere of a light into two hemispheres.

**Bottom, Top, Top And Bottom**: Selecting one of these items causes the custom light to shine only from its bottom, top, or all around, respectively. These variations are illustrated in Figure 6.6.6.7. Note that even a hemisphere which generally shines may contain points of zero intensity that do not shine. These areas can be established by editing and manipulating the default distribution curves that are initially generated according to the selection from **Hemisphere**.

![Figure 6.6.6.7: Distribution curves located in (a) the Top, (b) the Bottom, and (c) the Top And Bottom hemispheres.](image)
**Plane Angle**: This editable field displays the angle of the plane of the *active distribution curve* relative to the zero angle plane. Changing its value repositions the plane of the active distribution curve. If the angle of this plane cannot be changed (as for the single distribution curve used by the *Axial* symmetry), this field is dimmed and inactive.

**Intensity Angle**: This editable field displays the angle of the *active point* (on the active distribution curve) from the origin of the light. Changing this value repositions the point. If the angle of an active point cannot be changed, this field is dimmed. This occurs when one of the end points of the distribution curve is active, or the Lock option next to this field is on.

**Intensity Value**: This editable field displays the intensity of the light at the position of the *active point*. It is expressed as a percentage of the overall intensity defined by the radiometric or photometric units of the light. In the preview diagram of the custom light the intensity is represented as a relative distance from the origin of the light.

**Apply To All**: When this option is off, which is the default, any change made to the **Intensity Value** will be applied to the active distribution curve only. If this option is on, the change will be applied to all the distribution curves.

**New Plane...**: Clicking on this button invokes the **New Plane Angle** dialog (Figure 6.6.6.8) where the angle at which a new distribution curve plane will be created can be entered. The dialog also includes a message informing what the range of acceptable angles is, which depends on the type of symmetry that is currently used. These conditions are summarized in the table of Figure 6.6.6.9. When you click on **OK**, a plane is created, a copy of the active distribution curve is made on it, and this new curve becomes the active distribution curve.

**Delete Plane**: Clicking on this button deletes the active distribution curve and its plane. If the active curve is one that cannot be deleted, this button is dimmed and inactive.

**Figure 6.6.6.8**: The **New Plane Angle** dialog.

<table>
<thead>
<tr>
<th>Symmetry</th>
<th>Acceptable angle for new plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0° - 360°</td>
</tr>
<tr>
<td>Axial</td>
<td>no new plane can be created</td>
</tr>
<tr>
<td>Quadrant</td>
<td>0° - 90°</td>
</tr>
<tr>
<td>Plane</td>
<td>0° - 180°</td>
</tr>
</tbody>
</table>

**Figure 6.6.6.9**: Acceptable angles for new planes per type of symmetry.
**Show Grid**: When this option is on, the radial reference grid of the active distribution curve is also displayed. When off, the grid is hidden.

**Grid Options...**: Clicking on this button invokes the Radial Grid Options dialog (Figure 6.6.6.10) where the parameters of the grid can be set. These parameters are the **Angle Spacing**, expressed in degrees, and the **Distance Spacing**, expressed as a percentage.

**Snap To Grid**: When this option is on and the position of the active point is graphically manipulated using the Arrow tool under the preview area, it can only be moved to the points of the grid. These are the points where the concentric circles intersect with the radii of the grid.

**Load...**: When this option is selected, a data file containing the parameters for a custom light may be loaded. This file may be in one of three industry standard formats.

- IES: Illumination Engineering Society of North America
- CIE: International Commission on Illumination
- CIBSE: Chartered Institution of Building Services Engineers (British)

These data files contain angles and values for one or more distribution curves as well as optional parameters for the output energy of the light. If energy output parameters are supplied, they will be entered in the respective fields. If not supplied, the parameters currently present are maintained. Most light fixture manufacturers supply data files in one of those three formats with their products.

**Save...**: When this option is selected, the current description of light distribution can be saved to a data file, formatted in one of the three standards.
A scene with a custom light is shown in Figure 6.6.6.11. In this example the custom light is used to simulate a hanging light fixture as with the point light. Figure 6.6.6.12 shows the **Light Parameters** dialog for that light and Figure 6.6.6.13 the **Intensity Distribution** dialog. Unlike a point light, which emits light equally strong in all directions, this custom light is set up to emit less light upwards, since a real hanging light fixture would shield the upward direction through the socket and hanging hardware. The falloff is set to **Square**, which relates to the behavior of light in the real world. Note that the radius parameter is set to 6'-0", which is approximately the distance of the light fixture from the floor. Setting this radius means that the floor receives light at 100% intensity. Surfaces closer than 6’, such as the table top, will get more light, surfaces further away will get less. You can recreate this rendering and look at the light parameters in more detail by loading sample file “Custom.fmz.”

*Figure 6.6.6.11:* A scene rendered with a custom light.

*Figure 6.6.6.12:* The **Light Parameters** dialog.

*Figure 6.6.6.13:* The **Intensity Distribution** dialog.
6.6.7 Environment lights

An environment light uses image based illumination. Not unlike with environment reflections (see section 6.1.8), an imaginary cube or sphere surrounds the scene on which an environment map is placed. From the pixels of the environment map, light is emitted toward the objects in the scene. The intensity of the light from each pixel is determined by the brightness of the pixel. As a result, bright areas in the environment map will cause a stronger illumination of the scene from the corresponding side of the environment. However, since even low intensity areas of the map emit some light, the illumination of the scene results in very subtle shading, achieving realistic looking images with comparatively short rendering times. This lighting technique can be used to place a model against a photographic background and have the model illuminated in such a way that it appears to be part of the photographic scene, since both the color and brightness of the photograph will affect the illumination. An example of a scene rendered four times, each time with a different environment light, is shown in Figure 6.6.7.1. Note how the different maps, shown next to the rendering, give the scene a different illumination.

To support this new lighting technique, two new image formats have been added, which use high dynamic ranges: HDRI and OpenEXR. Unlike standard image formats, such as TIFF, BMP or JPEG, high dynamic range images (or short HDR images) store color information for each pixel that goes beyond just plain white color. For example, in a TIFF image, the range for a pixel color will go from the RGB value 0,0,0, which is black, to the RGB value 1,1,1, which is considered white. Values past 1,1,1 do not exist. In a HDR image, the maximum value for a pixel can go well beyond 1,1,1. For example, a pixel value of 25,25,25 would represent an intensity from a very bright area in the image, such as a direct view into a light source, or a very bright spot of a glossy reflection. When viewing this image on the scene, the pixel will still appear white, just as if it were represented by an RGB value of 1,1,1. However, when this image is used as an environment map for the new light type, the real RGB value is used for the illumination of the scene, providing the correct exposure, as if the model were placed in the scene represented by the HDR image.
Figure 6.6.7.1: A scene rendered four times, each time with a different environment light, shown in column (a). The corresponding environment maps are shown in column (b).
Creating an environment light and setting its parameters

An environment light is created as with all the other light types. Clicking under the last entry in the Lights palette or clicking on the **New...** button in the **Lights** dialog invokes the **Light Parameters** dialog. In this dialog, select **Environment** from the **Type** menu, which causes the environment light specific options to appear in the dialog (Figure 6.6.7.2). Several of the options available to other light types are not applicable to environment lights and are dimmed. For example, the **Origin** and **Center Of Interest** fields don't apply. In addition, the environment light only supports raytraced shadows. Therefore the shadow type menu is disabled and the **Shadows Type** option is always set to **Hard (Raytraced)**.

Note that raytraced shadows from an environment light usually do not appear hard as the menu suggests. Due to the nature of the algorithm, a surface pixel is illuminated multiple times, according to the **# Of Samples** parameter, which is described in more detail below. This results in a very subtle illumination and soft shadows. For other light types, such as a cone light, a surface pixel is illuminated only once for each light and shadows need to be artificially softened through the use of shadow maps. As the **Atmospheric** option for distant lights, environment light works very well in a final gather based rendering. This is described in more detail in section 6.1.6.

**Parameters tab**

**Map Type**: From this menu, either **Cubic** or **Spheric** is selected. This is equivalent to the **Map Type** menu in the **Environment** section of the **RenderZone Options** dialog. When clicking on the **Options...** button next to the menu, the respective dialog to select a spherical or cubic environment map is invoked. This is described in further detail in section 6.1.8.

**Center And Size**: When clicking on this button the **Environment Center And Size** dialog is invoked. This is the same dialog as described in section 6.1.8. It defines whether the environment is considered infinite, or whether it has a fixed size and location.
**# Of Samples**: When rendered, the environment light is represented by a number of individual lights, called **samples**. The samples are evenly distributed over the environment and take on the intensity and color from the area in the environment which they represent. One can think of these light samples as small Christmas tree lights, with different intensities and colors, arranged evenly in a spherical fashion around the scene. The sum of intensities of all samples represents the overall intensity of the environment light. Note that this parameter has a direct impact on the rendering speed and quality. A low number of samples, such as 50 will result in a fast rendering, but the shading will be inaccurate, especially if shadows are on. A high number of samples will result in better quality images, but rendering times will also increase. For very high quality images, a value of 1000 and beyond for the **# Of Samples** parameter may not be uncommon. A simple scene rendered with 100 and 1000 samples is shown in Figure 6.6.7.3.

**Softness**: This parameter controls the size of the area in the environment map that is used to determine the intensity for a light sample. The larger this value, the less lighting details will appear. Note that using small values for the softness parameter will require a larger number of samples to produce good results. A simple scene rendered with 0 and 100% softness is shown in Figure 6.6.7.4. It is important to note that increasing the softness parameter does not create softer shadows. It may be quite possible that a scene renders almost identical with a low and a high softness. This would be the case, when the environment map does not contain areas of strong contrast. Only when the map has areas of strong intensity and areas of low intensity will a high softness parameter blur the transition between these areas and create softer lighting.
**Color Factor:** This parameter controls how much the light emitted by the environment map is colored by the pixels of the map. At a value of 0%, the environment map emits white light only. At 100% the light takes on the color of the pixels of the environment map. At values between 0 and 100% the light is scaled between white and full color. This option is intended to take out some of the color that exists in an environment map. For example, the map may have a blue sky in it. The light emitted from such a map would not be expected to be blue, as the sky emits mostly white light. For such a map the **Color Factor** parameter should be set to 0% or close to it. If the environment map shows an interior scene, for example a room with brightly painted walls, the light reflected off the walls does take on a certain amount of the wall color. In this case, the **Color Factor** should be increased. Note that at 100%, scenes usually receive too much color from an environment map, even if the map represents a brightly colored interior. It is up to the user to find an appropriate amount of coloring for a given map. This is best done by rendering a few sample images of the scene, keeping the **# Of Samples** parameter low and turning shadows off. This will create images quickly but with sufficient information to judge the coloring.

**Spin:** This parameter controls the rotation of the environment light around the world Z axis.
Setting up a scene with environments

It is quite common that a single environment is used as the background, as the environment for reflections and as the environment for the illumination. To achieve this, an environment light is created in the Lights palette. In the Light Parameters dialog, a particular spherical or cubic environment is set up, as illustrated in Figure 6.6.7.5. In the RenderZone Options dialog, the Environment option is turned on. The same environment mapping and image is chosen as for the environment light (Figure 6.6.7.6). From the Background menu, the Environment shader is selected (Figure 6.6.7.7). This will render the otherwise invisible environment as the background. When clicking OK from the RenderZone Options dialog, the scene will be rendered with the environment light providing the illumination, the global environment providing the reflections and the Environment background shader supplying the pixels in the background of the scene (Figure 6.6.7.8).

Figure 6.6.7.5: Choosing an environment map for the environment light, in (a) the Light Parameters dialog and (b) the Spherical Environment Map Options dialog.
Figure 6.6.7.6: Choosing the same environment map for the global environment in (a) the RenderZone Options dialog and (b) the Spherical Environment Map Options dialog.

Figure 6.6.7.7: Choosing the Environment background shader in the RenderZone Options dialog.
Unlike other light types, there is no graphic representation of an environment light in a project window. Since the environment is considered infinitely large and without any particular direction, wireframe graphics as drawn for other types of lights do not apply.

There are a number of places on the web to download free HDR images, which can be used as maps for the new environment light. LightWork Design offers a set of these maps from different sources as a starter collection. Their address is http://www.lightworks-user.com/hdri_starter_collection.htm. The renderings and maps shown in this section were created using HDRI data courtesy of Sachform Technology GbR (www.sachform.de) from the LightWorks HDRI Starter Collection (www.lightworkdesign.com).
6.6.8 Specifying the correct light intensity

The light intensity of all light types except distance lights can be specified in either lumens or watts. While the photometric intensity in lumens is the preferred unit with respect to generating accurate lighting results, in many cases the lumens rating of a bulb may not be known, while the electric input power in watts is. Unfortunately, there is no direct conversion from watts to lumens, since watts represents the electric input power and the light’s output also depends on the efficiency with which electric energy is converted into visible light energy. The charts in Figure 6.6.8.1 contain the watts/lumens correspondence for various light types.

Note that the lumens rating of fluorescent tubes varies for the same input power because of differences in the color spectrum. Typical color temperatures for some standard lamp types are listed in the chart of Figure 6.6.8.2.

<table>
<thead>
<tr>
<th>Lamp type</th>
<th>Color temperature in degrees Kelvin</th>
</tr>
</thead>
<tbody>
<tr>
<td>warm white fluorescent</td>
<td>3020</td>
</tr>
<tr>
<td>white fluorescent</td>
<td>3450</td>
</tr>
<tr>
<td>cool white fluorescent</td>
<td>4250</td>
</tr>
<tr>
<td>daylight fluorescent</td>
<td>6250</td>
</tr>
<tr>
<td>clear mercury</td>
<td>5710</td>
</tr>
<tr>
<td>improved color mercury</td>
<td>4430</td>
</tr>
<tr>
<td>low pressure sodium</td>
<td>1740</td>
</tr>
<tr>
<td>high pressure sodium</td>
<td>2100</td>
</tr>
<tr>
<td>tungsten halogen</td>
<td>3190</td>
</tr>
</tbody>
</table>

**Figure 6.6.8.2:** Color temperatures for standard lamp types.

**Figure 6.6.8.1:** Watts/lumens correspondence for various light types.
6.7 Tutorial examples

6.7.1 Bulb

Light bulb glass

**Color:** Plain
R: 100 %, G: 100 %, B: 0 %

**Reflection:** Constant

**Transparency:** Center-Edge
- Center Transparency: 75 %
- Edge Transparency: 47 %

**Bump:** None

Metal socket

**Color:** Plain
R: 81 %, G: 58 %, B: 2 %

**Reflection:** Metal, Accurate
- Ambient Reflection: 10 %
- Diffuse Reflection: 80 %
- Specular Reflection: 80 %
- Reflectivity: 31 %
- Roughness: 10
- Refraction: R: 0.5, G: 0.5, B: 0.5

**Transparency:** None
**Bump:** None

Bulb filament

**Color:** Plain
R: 100 %, G: 100 %, B: 0 %

**Reflection:** Constant

**Transparency:** None
**Bump:** None

Filament glow

**Color:** Plain
R: 100 %, G: 98 %, B: 38 %

**Reflection:** Constant

**Transparency:** Neon
- Intensity: 69 %
- Falloff: Linear

**Bump:** None

Bulb Glow

**Color:** Plain
R: 100 %, G: 98 %, B: 38 %

**Reflection:** Constant

**Transparency:** Neon
- Intensity: 60 %
- Falloff: Linear

**Bump:** None

Filament holders

**Color:** Plain
R: 48 %, G: 49 %, B: 48 %

**Reflection:** Metal, Simple
- Ambient Reflection: 100 %
- Diffuse Reflection: 100 %
- Roughness: 25

**Transparency:** None
**Bump:** None

RenderZone rendering options

**Rendering Mode:** Z-Buffer

**Shadows:** Off

**Bump Mapping:** Off

**Transparencies:** On

**Reflections:** On

**Environment:** Off

**Depth Effects:** Off

**Background:** Project Color (black)
6.7.2 Landscape

Mountains
- **Color**: Color Map A
- **Reflection**: Plastic
  - Ambient Reflection: 100%
  - Diffuse Reflection: 75%
  - Specular Reflection: 30%
  - Roughness: 31%
  - Specular Color: R: 100%, G: 100%, B: 100%
- **Transparency**: None
- **Bump**: Casting
  - Scale: 100%
  - Casting Amplitude: 10%
  - Dented Amplitude: 30%
  - Dented Scale: 100%
  - Dented Threshold: 40%
  - Detail: 30%

**Texture Map Controls**:
- Texture Group: Object
- Origin: Object Center
- Rotation: None
- Mapping Type: Cylindrical
- Horizontal Size: 60°
- Horizontal Center: On
- Horizontal Mirror: Off
- Vertical Size: 34'-0"
- Vertical Center: On
- Vertical Mirror: Off

Water
- **Color**: Plain
  - R: 30%, G: 40%, B: 69%
- **Reflection**: Mirror
  - Ambient Reflection: 10%
  - Diffuse Reflection: 28%
  - Specular Reflection: 90%
  - Reflection: 56%
  - Roughness: 6.25
- **Transparency**: None
- **Bump**: Rough, Solid
  - Scale: 100%
  - Amplitude: 4%
  - Detail: 30%
  - Sharpness: 0%
Planet

**Color:** Plain  
R: 100 %, G: 72 %, B: 8 %

**Reflection:** Matte  
Ambient Reflection: 250 %
Diffuse Reflection: 100 %

**Transparency:** None

**Bump:** Casting  
Scale: 100 %
Casting Amplitude: -33 %
Dented Amplitude: 0 %
Dented Scale: 0 %
Dented Threshold: 0 %
Detail: 20 %

---

Planet's ring

**Color:** Marble, Simple  
Scale: 100 %
Detail: 30 %
Color: R: 100 %, G: 35 %, B: 12 %

**Reflection:** Constant

**Transparency:** Transparency Map B

**Bump:** None

---

**Texture Map Controls:**
- Texture Group: Object
- Origin: Object Center
- Rotation: None
- Mapping Type: Flat
- Horizontal Size: 141'
- Horizontal Center: On
- Horizontal Mirror: Off
- Vertical Size: 141'
- Vertical Center: On
- Vertical Mirror: Off
- Solid Textures Size: 14'-0''

---

**RenderZone rendering options**

**Rendering Mode:** Raytrace

**Shadows:** Off

**Bump Mapping:** On

**Transparencies:** On

**Reflections:** On

**Environment:** Cubic, Environment Map

**Depth Effects:** Off

**Background:** Sky

**Environment Maps:**  
Image C on each side of environment cube

**Horizon:**

**Top Sky Color:**  
R: 26 %, G: 6 %, B: 39 %

**Bottom Sky Color:**  
R: 100 %, G: 47 %, B: 7 %

**Top Earth Color:**  
R: 2 %, G: 20 %, B: 13 %

**Bottom Earth Color:**  
R: 44 %, G: 55 %, B: 44 %

**Horizon:** 65 %
6.7.3 Bottles

Green (tall) bottle

<table>
<thead>
<tr>
<th>Color</th>
<th>Plain</th>
</tr>
</thead>
<tbody>
<tr>
<td>R: 62 %, G: 89 %, B: 70 %</td>
<td></td>
</tr>
<tr>
<td>Reflection</td>
<td>Glass, Accurate</td>
</tr>
<tr>
<td>Ambient Reflection</td>
<td>0 %</td>
</tr>
<tr>
<td>Diffuse Reflection</td>
<td>0 %</td>
</tr>
<tr>
<td>Specular Reflection</td>
<td>90 %</td>
</tr>
<tr>
<td>Transmission</td>
<td>88 %</td>
</tr>
<tr>
<td>Reflectivity</td>
<td>80 %</td>
</tr>
<tr>
<td>Roughness</td>
<td>2</td>
</tr>
<tr>
<td>Refraction</td>
<td>1.591</td>
</tr>
<tr>
<td>Transparency</td>
<td>None</td>
</tr>
<tr>
<td>Bump</td>
<td>None</td>
</tr>
</tbody>
</table>

Ash tray

<table>
<thead>
<tr>
<th>Color</th>
<th>Marble, Simple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale: 5000 %</td>
<td></td>
</tr>
<tr>
<td>Detail: 30 %</td>
<td></td>
</tr>
<tr>
<td>Color: R: 59 %, G: 44 %, B: 60 %</td>
<td></td>
</tr>
<tr>
<td>Reflection</td>
<td>Glass, Accurate</td>
</tr>
<tr>
<td>Ambient Reflection</td>
<td>28%</td>
</tr>
<tr>
<td>Diffuse Reflection</td>
<td>11%</td>
</tr>
<tr>
<td>Specular Reflection</td>
<td>90%</td>
</tr>
<tr>
<td>Transmission</td>
<td>90 %</td>
</tr>
<tr>
<td>Reflectivity</td>
<td>90 %</td>
</tr>
<tr>
<td>Roughness</td>
<td>3</td>
</tr>
<tr>
<td>Refraction</td>
<td>1.591</td>
</tr>
<tr>
<td>Transparency</td>
<td>None</td>
</tr>
<tr>
<td>Bump</td>
<td>None</td>
</tr>
</tbody>
</table>

Red vase

<table>
<thead>
<tr>
<th>Color</th>
<th>Plain</th>
</tr>
</thead>
<tbody>
<tr>
<td>R: 96 %, G: 37 %, B: 42 %</td>
<td></td>
</tr>
<tr>
<td>Reflection</td>
<td>Glass, Accurate</td>
</tr>
<tr>
<td>Ambient Reflection</td>
<td>0 %</td>
</tr>
<tr>
<td>Diffuse Reflection</td>
<td>10%</td>
</tr>
<tr>
<td>Specular Reflection</td>
<td>90 %</td>
</tr>
<tr>
<td>Transmission</td>
<td>90 %</td>
</tr>
<tr>
<td>Reflectivity</td>
<td>57 %</td>
</tr>
<tr>
<td>Roughness</td>
<td>2</td>
</tr>
<tr>
<td>Refraction</td>
<td>1.591</td>
</tr>
<tr>
<td>Transparency</td>
<td>None</td>
</tr>
<tr>
<td>Bump</td>
<td>None</td>
</tr>
</tbody>
</table>

Wine glass

<table>
<thead>
<tr>
<th>Color</th>
<th>Plain</th>
</tr>
</thead>
<tbody>
<tr>
<td>R: 90 %, G: 100 %, B: 99 %</td>
<td></td>
</tr>
<tr>
<td>Reflection</td>
<td>Glass, Accurate</td>
</tr>
<tr>
<td>Ambient Reflection</td>
<td>0%</td>
</tr>
<tr>
<td>Diffuse Reflection</td>
<td>0%</td>
</tr>
<tr>
<td>Specular Reflection</td>
<td>90 %</td>
</tr>
<tr>
<td>Transmission</td>
<td>90 %</td>
</tr>
<tr>
<td>Reflectivity</td>
<td>90 %</td>
</tr>
<tr>
<td>Roughness</td>
<td>2</td>
</tr>
<tr>
<td>Refraction</td>
<td>1.591</td>
</tr>
<tr>
<td>Transparency</td>
<td>None</td>
</tr>
<tr>
<td>Bump</td>
<td>None</td>
</tr>
</tbody>
</table>

Whiskey glass

Same as above
Whiskey bottle

**Color:** Plain
- R: 100 %, G: 96 %, B: 72 %

**Reflection:** Glass, Accurate
- Ambient Reflection: 0 %
- Diffuse Reflection: 0 %
- Specular Reflection: 90 %
- Transmission: 100 %
- Reflectivity: 50 %
- Roughness: 6
- Refraction: 1.591

**Transparency:** None

**Bump:** Rough, Solid
- Scale: 8000 %
- Amplitude: 8 %
- Detail: 10 %
- Sharpness: 0 %

Table surface

**Color:** Marble, Simple
- Scale: 20000 %
- Detail: 50 %
- Color: R: 47 %, G: 55 %, B: 60 %

**Reflection:** Mirror
- Ambient Reflection: 100 %
- Diffuse Reflection: 100 %
- Specular Reflection: 80 %
- Reflectivity: 30 %
- Roughness: 6.25

**Transparency:** None

**Bump Shader:** None

**RenderZone rendering options**

- **Rendering Mode:** Raytrace
- **Shadows:** Off
- **Bump Mapping:** On
- **Transparencies:** On
- **Reflections:** On
- **Environment:** Cubic, Environment Map
- **Depth Effects:** Off
- **Background:** Project Color (white)

**Environment Maps:**
- White image map on each side of environment cube
6.7.4 Magnifying glass

Magnifying lens

**Color:** Plain
- R: 100 %, G: 100 %, B: 100 %

**Reflection:** Glass, Accurate
- Ambient Reflection: 0 %
- Diffuse Reflection: 0 %
- Specular Reflection: 90 %
- Transmission: 90 %
- Reflectivity: 19 %
- Roughness: 2
- Refraction: 1.591

**Transparency:** None

**Bump:** None

Magnitude lens frame

**Color:** Plain
- R: 95 %, G: 70 %, B: 39 %

**Reflection:** Metal, Accurate
- Ambient Reflection: 20 %
- Diffuse Reflection: 70 %
- Specular Reflection: 30 %
- Reflectivity: 35 %
- Roughness: 10
- Refraction: R: 0.186, G: 0.59, B: 1.518
- Absorption: R: 3.756, G: 2.297, B: 1.897

**Transparency:** None

**Bump:** None

Sheet of paper

**Color:** Color Map of scanned text

- Horizontal Reps: Infinite
- Vertical Reps: Infinite
- Scale: 100%
- MIP Map Sampling: On
- Cropping: None

**Reflection:** Matte
- Ambient Reflection: 100 %
- Diffuse Reflection: 100 %

**Transparency:** None

**Bump:** None

RenderZone rendering options

- **Rendering Mode:** Raytrace
- **Shadows:** On
- **Bump Mapping:** Off
- **Transparencies:** On
- **Reflections:** On
- **Environment:** Off
- **Depth Effects:** Off
- **Background:** White
6.8 Rendering with radiosity

form•Z RenderZone Plus can also be used to render with radiosity, which can accurately simulate the lighting conditions in a scene and incorporate them in a rendering.

Radiosity techniques were first used in thermal engineering to simulate the transfer of heat energy between surfaces. With the advance of computer graphics in recent years, radiosity has also been used to simulate the transfer of light between surfaces. In fact, radiosity simulates the distribution of light in an environment in a physically accurate manner. The representation of this simulation, called the *radiosity solution*, describes the distribution of light from all sources across all surfaces in the scene, and it also accounts for light that bounces off surfaces and illuminates other surfaces. It should be noted that radiosity is not by itself a rendering algorithm, such as raytracing or z-buffer. It is an accurate representation of the illumination model of a scene. After such a model has been derived, it is rendered using a raytrace or z-buffer rendering procedure. In form•Z RenderZone Plus this is done by applying Shaded Render* or one of the RenderZone* rendering modes to a radiosity solution. A radiosity solution can also be used to produce an Open GL rendering using the Interactive Shaded rendering mode.

The RenderZone rendering mode offers a number of methods that are known as *global illumination*. One of these methods is an algorithm, called *final gather*, that performs its own calculation of reflected light, although not as accurately as radiosity. For increased illumination quality, final gather can utilize a lighting solution calculated by radiosity and produce the most realistic images, with minimal setup and without the rendering artifacts that may appear when a radiosity solution is rendered directly.

Non-radiosity based rendering methods use a simple illumination model to simulate the interaction of light with the environment. During such a rendering, the light intensity at a pixel is calculated by shooting a ray from the pixel's position back to a light source. Taking into account the falloff of the light intensity, the distance of the pixel from the light source determines the light intensity at the pixel. If this ray is blocked by a surface which is located between the pixel and the source, the pixel is in shadow and no illumination of the pixel occurs. A scene rendered with this approach only illuminates pixels which are directly hit by a light source. In order to give all other pixels some light intensity, a constant global ambient light is added. This ambient light is a crude approximation of the light reflecting off surfaces, illuminating other surfaces indirectly. Areas in a scene which are illuminated by the constant ambient light alone show no variance in intensity, which is responsible for the artificial appearance of such renderings.
The radiosity solution correctly simulates this ambient light by computing light distribution which results from light reflecting off surfaces. It is this secondary illumination that gives radiosity based renderings their high degree of realism.

The radiosity solution is **view independent**. This means that, once a radiosity solution has been completed, any number of renderings can be produced from it without having to calculate the light intensities again. As a result, radiosity based renderings can even be faster than renderings using the simple illumination model, given the same amount of geometry to process.

**What is radiosity and how it works**

Formally, **radiosity** is a measure of energy leaving a surface, per unit area, per unit time. The underlying strategy of radiosity is to divide all surfaces in a scene into small polygons, called **patches** and to compute the radiosity value for each. The radiosity value of all patches is determined by first distributing energy from all primary light sources to all patches. Each patch absorbs some of the energy it receives and reflects the rest. The amount of reflection depends on the diffuse reflection parameter associated with the patch. After the distribution of light from the **primary sources**, the patch which reflects the most amount of energy is now considered a light source itself, called a **secondary source**. It distributes its reflected energy to all other patches in the scene. Next, the patch which reflects the second highest amount of energy is processed, and so on. Note that the first patch may reappear at some point in the list of light emitting patches, because it has received enough energy from other patches to become the brightest patch in the scene. Obviously, this process will continue forever, since surfaces typically emit a portion of the energy they receive. It is therefore necessary to specify some criteria, which will stop the radiosity process. These criteria may be a given amount of time that the process is allowed to run, a given number of cycles, or the amount of energy absorbed by a patch relative to the overall energy emitted by all primary sources.

By the time the radiosity process stops, a mesh of small patches with varying intensities has been generated. In most cases, the radiosity mesh will be used by a RenderZone rendering with the final gather global illumination method. Final gather will use the precomputed light intensities of the mesh as additional input to its own lighting calculations and produces a high quality realistic rendering. This mesh can also be rendered without final gather, substituting the per pixel intensity calculation of the simple illumination model with the precalculated intensity of the patch. The rendering is also able to smooth the intensity changes between neighboring patches, giving the impression of a gradual change of illumination and soft shadow boundaries. Such a rendering will show a varying amount of light in areas which are not directly illuminated by the primary light sources. In contrast, a rendering generated with the simple illumination model shows no such variation. A scene rendered with a simple illumination model and from a radiosity solution, using **RenderZone** and **Final Gather**, is shown in Figure 6.8.0.1.
Figure 6.8.0.1: A scene rendered with (a) a simple illumination model and (b) from a radiosity solution using RenderZone with Final Gather.
Reflections in radiosity

Photorealistic renderings typically simulate three kinds of reflections: diffuse and specular reflections of light, and mirrored reflections of the environment. Radiosity simulates diffuse reflections. Specular and mirrored reflections may be applied when using raytracing to render the radiosity solution.

Diffuse reflection of light is responsible for the gradual change of shading on a curved surface. A light ray which hits a surface at a given angle is scattered by the surface and an infinite number of new rays is emitted at equal intensity in all directions, as shown in Figure 6.8.0.2 (a). The intensity of the reflected rays is based on the angle of the incoming ray and on the ability of the material to produce diffuse reflections. In addition, the color of the reflected rays is composed of the incoming color filtered through the color of the surface material, generating what is commonly perceived as the color of a material. Surfaces which show effects from diffuse reflection alone will result in a dull, matte appearance. Figure 6.8.0.2 (b) shows a spherical surface which exhibits only diffuse reflection.

Specular reflection of light bounces the incoming light ray off the surface at the same, or close to the same angle, as illustrated in Figure 6.8.0.3 (a). That is, specular reflection can only be observed, if the outgoing ray hits the eye exactly. The reflected light ray of a specular reflection does not take on the color of the material, but maintains the color of the incoming light. As a result, specular reflections generate hotspots on curved surfaces in the color of the light, which is typical for highly polished and glossy materials. Figure 6.8.0.3 (b) shows a spherical surface which has a high degree of specular light reflection.

Mirrored reflections of the environment are generated by reflecting the line of sight off a surface at the incoming angle and intersecting the reflected ray with whatever surface it hits next. Mirrored reflections of the environment are commonly simulated in computer graphics through raytracing techniques. Figure 6.8.0.4 shows a sphere with mirrored environment reflections.

The final color of a pixel on a surface that exhibits reflections is a mixture of the colors generated by all three reflection types. As already noted, a radiosity based rendering receives its diffuse contribution from the precalculated intensities at the mesh patches, whereas specular and mirrored reflections, transparencies, and other rendering effects are calculated by the same algorithms used by the simple illumination model.
Usage of radiosity: final gather support, hybrid and pure

Computing a highly accurate radiosity solution can be an intensive task. Much effort is spent calculating the intensities at the radiosity mesh patches. On a given patch, a number of points are chosen randomly. From each point a light ray is cast to the vertices of all other patches in the scene. The illumination of all patches by a single source patch is called an iteration. The distance from the sampled point to the vertices determines the amount of energy received at the other patches. Recall that light intensity falls off with the square of the distance the light ray travels. In order for every patch to illuminate all other patches in the scene, \( n \times n \) iterations must be performed, where \( n \) is the number of patches in the scene. This is an exponential process. In other words, doubling the number of patches in the scene results in four times the number of iterations. It is apparent that a dense mesh of several thousand patches will significantly increase the time required to process patch illumination. Therefore, minimizing the amount of patches is a primary concern.

The accuracy of a radiosity solution is measured by the percentage of the light that has been absorbed by the radiosity patches over the total light initially emitted by all the active light sources. Since the computation of the radiosity solution is not a linear process, a solution, which is for example 70% accurate, can be achieved in a relatively short period of time, whereas the next 10% will take considerably longer. As the solution approaches 100% accuracy, increasingly more time will be required.

Radiosity is intended to be used in three distinctly different scenarios. This first, and most important application is as support for the final gather global illumination method of the RenderZone display mode. The second is a hybrid use, where radiosity supplies the secondary illumination in a scene and the primary (or direct) illumination is generated for each pixel at rendering time. The third is the pure use of a radiosity solution, where the mesh is rendered directly, displaying the precomputed illumination of each patch. It is expected, that the majority of users will only use radiosity as final gather support. For this purpose, it is simple to setup and most of the shortcomings and issues that arise with the other two uses will do not occur. A brief discussion of the three uses follows next. General concepts that apply to all three are discussed up to chapter 6.8.3. They may be of interest to all users. In chapter 6.8.3 only the section describing the Final Gather option may be of interest to the general user. The sections about Render Direct Illumination and Pure Radiosity in 6.8.3 may be skipped. The sections titled Termination tab, Preview tab, and Statistics tab in 6.8.3 are again of interest for the general user. Sections 6.8.4 and 6.8.5 contain important information about optimizing a radiosity solution.
The first method, final gather support, only attempts to create a rough radiosity solution, where illumination is generated quickly. The mesh can be rather coarse and it is not necessary to let the solution progress for a long time. It is enough, if radiosity distributes some light throughout the scene, which benefits the final gather rendering method. Many of the advanced parameters of the other two radiosity methods do not apply in this case and are never shown to the user. The radiosity mesh, when rendered without final gather, for previewing in Interactive Shaded or Shaded Render, will show inaccurate shading and rough shadows, due to the large size of the mesh patches. This is of no concern, since final gather will eliminate these artifacts. A radiosity solution generated under the final gather support method is shown in a preview rendering in Figure 6.8.0.5 and in a final gather rendering in Figure 6.8.0.6.

![Figure 6.8.0.5: A preview rendering of a radiosity solution intended for final gather](image1)

![Figure 6.8.0.6: A final gather rendering based on the same radiosity solution.](image2)

The second method combines the secondary illumination calculated by radiosity with the primary (direct) illumination calculated at rendering time. This allows the radiosity mesh to be coarser, because shadow boundaries from direct lighting is calculated at rendering time. Prior to the availability of final gather in form•Z, this was the preferred method of creating high photorealistic images. While this method is still supported, it is less important now. It is discussed in more detail in section 6.8.3 under the title: The Options tab for Render Direct Illumination.

With the third method, radiosity calculates primary and secondary illumination. To create fine contrast along boundaries of shadows from direct lighting, the radiosity mesh needs to become very fine in those areas. This leads to long computation times. This method is provided for those users that want to control a radiosity solution with all available parameters. It is not intended for everyday use. It is discussed in more detail in section 6.8.3 under the title: The Options tab for Pure Radiosity.
6.8.1 The radiosity process and commands

form•Z RenderZone Plus offers four menu items in the **Display** menu (Figure 6.8.1.1) for producing radiosity solutions and renderings. These are supported by dialogs where the different options can be set.

Radiosity is a mode within **form•Z**. Once you enter the radiosity mode, you cannot make changes to the model until you exit. While in the radiosity mode, you can adjust the radiosity parameters, change views, produce radiosity illuminated renderings (using **Interactive Shaded***, **Shaded Render***, or **RenderZone***), and save the **form•Z** project.

A radiosity session is typically executed in an iterative manner, where the solution and rendering are gradually refined until the desirable quality is reached. Following is an outline of a typical radiosity session:

- An interior scene is modeled, using multiple light sources.
- The **Radiosity Options...** menu command is selected.
- This invokes the **Radiosity Options** dialog, where the desired parameters are chosen.
- After clicking **OK** in the **Radiosity Options** dialog, the **Initialize Radiosity*** item is selected and **form•Z** enters the radiosity mode.
- Next the **Generate Radiosity Solution*** menu item is selected.
- A progress bar appears, indicating the density of the mesh and the progress of the solution towards a desired level of completion.
- At specified intervals a shaded preview is drawn, allowing you to visualize the solution.
If it turns out that the lighting is not as described, the radiosity progress is canceled by selecting **Exit Radiosity**. Changes to the light sources are made and **Generate Radiosity Solution** is executed again. This process of refinement may be repeated several times until the preview images show a satisfactory result. After the solution has reached the desired level, several renderings from different views may be produced. The images can be saved using the **Export Image** command in the **File** menu.

• The radiosity work is completed by selecting the **Exit Radiosity** menu item.

These basic steps are illustrated in a small example below and can be executed by the user. The sample file “radiosity_basic.fmz” is provided with the program. It contains a simple interior scene with a few pieces of furniture and several light sources. The file is preset so that a radiosity solution can be generated right away by following these steps:

• Open file “radiosity_basic.fmz.”

• Select **RenderZone** from the **Display** menu. This will render the scene with final gather but without radiosity, as shown in Figure 6.8.1.2.

• Select **Initialize Radiosity** from the **Display** menu. This will generate the basic mesh as shown in Figure 6.8.1.3.

• Select **Shaded Render** from the **Display** menu and then **Generate Radiosity Solution**. In intervals of about 20 seconds a preview rendering will be shown of the progress of the radiosity solution. A few of the renderings are shown in Figures 6.8.1.4(a) through (d). Notice how the lighting becomes increasingly more accurate.

• After the radiosity solution is complete, select **RenderZone** again. This time the final gather rendering utilizes the additional indirect lighting computed on radiosity. The final rendering is shown in Figure 6.8.1.5.
Figure 6.8.1.4: Radiosity renderings at four different progressive stages.
While in **Radiosity** mode, it is possible to make changes to both surface styles and lights. After editing a surface style, the radiosity solution may be continued, and from that point on it will use the new parameters of the surface style. For example, if the color of a surface style was changed, the light reflected off faces using this surface style will now be filtered through the new color. Note, that the light which was distributed in the scene prior to the editing of the surface style is not affected by the color change. Editing a surface style during a radiosity process is intended to make subtle changes to its parameters to fine tune the appearance of surfaces. In general, small changes to a surface style will not affect the accuracy of the solution. At the same time, there is nothing to prevent the user from drastically altering a surface style, which would result in a situation where the appearance of a surface and the corresponding lighting conditions of a scene do not correspond with realistic conditions.

In a similar manner, lights can also be edited while in radiosity mode. Unlike surface styles, changes made to a light source require that the current radiosity solution be deleted before the radiosity process is continued. Any changes made to the lights in the scene will cause the radiosity process to start over as soon as **Generate Radiosity Solution** is selected from the **Display** menu. The ability to edit lights while in radiosity mode is intended to be a shortcut. Instead of requiring you to select the **Exit Radiosity** menu command before editing a light, the lights can be changed directly. To prevent accidental changes to a light while in radiosity mode, a warning message is posted before the changes are accepted, which gives the user the option to cancel the changes.

![Image](image.png)

**Figure 6.8.1.5:** A final rendering of the complete Radiosity solution.
The radiosity commands

Radiosity Options...

Selecting this item invokes the **Radiosity Options** dialog discussed in section 6.8.3 and shown in Figure 6.8.3.1. This dialog, which can also be invoked from the next two menu items, contains the radiosity parameters and settings.

**Initialize Radiosity*/Enter Radiosity**

This item takes one of the names shown, depending on whether a previous radiosity solution exists or not. When a radiosity solution will be generated for the first time, it reads **Initialize Radiosity**. When an already existing solution is to be processed further, it reads **Enter Radiosity**. In both cases, selection of this item causes form•Z to enter the radiosity mode. Initializing radiosity constructs mesh patches from all of the objects in the scene based on the parameters in the **Radiosity Options** dialog. The form•Z project is frozen at this point and no changes can be made to the model. In addition, the content of the scene cannot be changed and objects and layers cannot have their visibility changed. To indicate that the radiosity mode is active, the form•Z environment dims all unavailable tools, menu items, and palettes (see Figure 6.8.1.6).

Once the **Initialize Radiosity** command has been selected, it can be chosen again at any time to, for example, correct some of the parameters set in the **Radiosity Options** dialog. When selected a second time before the solution is started, the initial mesh is simply regenerated. After the solution has been started, the current solution is discarded first before the solution is initialized again.

Since a radiosity solution is view independent, all of the tools and menu items relating to view control remain available. The lighting model that is represented in the radiosity solution can not be used in certain display modes, such as **Quick Paint**, **Hidden Line**, and **Surface Render**. These items appear dimmed in radiosity mode.

*Figure 6.8.1.6:* The form•Z environment when radiosity is initialized.
Generate Radiosity Solution*

When this item is selected, the radiosity process is started. If Initialize Radiosity* was executed prior to selecting Generate Radiosity Solution*, the radiosity process is started immediately. Otherwise radiosity is initialized first. Starting the radiosity solution invokes continuous iterations of patch illumination. A progress bar appears indicating how many mesh polygons currently exist in the scene and how far the solution has progressed toward the user defined goal. At a given interval, a shaded rendering or a wireframe rendering shows the current state of the solution. The shaded rendering uses the calculated patch intensities as the base for the diffuse reflection for the objects in the scene. The execution of the radiosity solution can be stopped at any time by clicking on the Stop button of the progress bar, or by hitting the shortcut key combination assigned to the Stop action.

Exit Radiosity

This menu item is initially dimmed. It becomes active and can be selected after the radiosity mesh has been initialized by choosing the Initialize Radiosity* or Generate Radiosity Solution* commands. When selected, the program exits the radiosity mode and all tools, palettes, and menu items become active again.

After a radiosity solution has been stopped by pressing the Stop button or by hitting the cancel shortcut key, it is possible to restart the solution by selecting the Generate Radiosity Solution* menu item. Typically, a solution is stopped in order to change some of the parameters in the Radiosity Options dialog or to fine tune the solution based on the visual feedback provided by the preview renderings and by the information shown in the progress bar. The solution may also be stopped when it is considered sufficiently advanced, in order to generate final renderings or real time renderings. The display commands, Wireframe*, Shaded Render*, RenderZone* and Interactive Shaded*, remain active during radiosity. They can be selected to produce a rendering using the radiosity solution. It is therefore possible, for example, to set up a new view in Wireframe* display and render a RenderZone* image, or select Interactive Shaded* and navigate through the scene in real time using standard view manipulation tools.

When a radiosity solution is interrupted, it is possible to preserve it at the current state, so that it may be continued later. This is done by writing it into a temporary file when Exit Radiosity is executed or when toggling between projects. The solution can also be saved more permanently by saving it with the project when Save or Save As... is executed. Whether or not these options are active is set in the Radiosity portion dialog (Figure 6.8.2.1), which is in the Preferences dialog.
6.8.2 The radiosity preferences

The Preferences dialog of form•Z RenderZone Plus contains a section for selecting certain radiosity options.

When clicking on the Radiosity item in the list on the left side in the Preferences dialog, the Radiosity Preferences are displayed on the right side (Figure 6.8.2.1).

Warn Before Entering And Exiting Radiosity: When this option is selected, a warning message is posted when one of the following conditions exist:

- A distant light is active, which does not cast shadows.
- Radiosity is exited by selecting Exit Radiosity from the Display menu, a window is closed, or you start a radiosity solution in another project, while Preserve Solution During Working Session is off (see below).

Save Solution In Project File: When this option is on and Save or Save As... is selected from the File menu the radiosity solution is saved with the form•Z project. When such a file is opened, and Generate Radiosity Solution* is selected from the Display menu, the previously saved radiosity solution is restored and the radiosity process continues from where it was interrupted when the file was saved. This option is on by default.

Preserve Solution During Working Session: When this option is selected, the current radiosity solution is not discarded when selecting the Exit Radiosity command from the Display menu, but is saved to a file on disk. If no change is made to an object or light and Generate Radiosity Solution* is selected from the Display menu, the radiosity process continues from where it was interrupted. However note that, if the geometry of an object or a light’s parameters change, the radiosity solution also has to change and can not be continued, regardless of whether this option is on or off. This option also affects whether a radiosity solution will be preserved to be continued later when we switch active projects. That is, given that only one radiosity solution may be in progress at any time, which is the solution of the active project, when we start a radiosity solution in another project, the radiosity solution of the previous project (when there is one in progress) cannot be continued. It either has to be discarded or saved on disk. With this option on, the solution is saved to disk and is restored when the project becomes active again. If this option is off, the current radiosity solution is always discarded when selecting Exit Radiosity or when starting a radiosity solutions in another project. Saving the solution on disk may require some noticeable time. Consequently, this option is off by default.
6.8.3 The common radiosity parameters

Selecting the Radiosity Options... item from the Display menu invokes the Radiosity Options dialog shown in Figure 6.8.3.1. At the top of this dialog is the Mode pop up menu from which one of three options can be selected: Final Gather Support, Render Direct Illumination, and Pure Radiosity. Each one of these items represents a different way of applying radiosity, as discussed earlier in section 6.8.

Final Gather Support

When this mode is chosen, the radiosity process is optimized to support RenderZone rendering with final gather, which is described in more detail in section 6.1.6. In this mode, the radiosity mesh is never rendered directly by a RenderZone rendering and therefore many of the complex parameters that control radiosity are hidden from the user and are set to fixed values.

Render Direct Illumination

When this mode is chosen, the radiosity process is set up to only calculate secondary (indirect) illumination. It is expected that a RenderZone rendering is performed afterward, during which the direct illumination is calculated for each pixel.

Pure Radiosity

Selection of this mode makes it possible for a user to access and directly set a variety of radiosity parameters. It is expected that this mode will only be selected by experienced users and in rather infrequent cases. While it allows complete control of the radiosity parameters it also requires a deep knowledge of the radiosity process.

Below the Mode menu, the dialog is organized in four tabs, Options, Termination, Preview, and Statistics. In the Options tab, a different set of parameters is displayed for each of the three modes. While most of the parameters are different for each mode, a few are the same and apply to all three modes. This is also true for the parameters of the other tabs, which remain the same for all modes.
The Common Options tab parameters

The following three parameters appear in the Options tab for all modes, as shown in Figures 6.8.3.2, 6.8.3.5, and 6.8.3.7.

Light Intensity: The items in this menu determine the intensity specifications of the light sources. All Simple forces all light sources to use the Simple intensity settings. All Accurate causes all light sources to use the Accurate intensity parameters. Per Light chooses the intensity specification favored by each individual light source.

View Dependent Solution: If only a single rendering is required from a radiosity solution, it is not necessary to generate dense areas of the radiosity mesh in regions that are not visible. Turning this option on will avoid additional mesh subdivision in invisible regions of the scene, which leads to shorter execution times of the solution. While it is still possible to render images from different views, optimal results are only achieved from the view which is set when the radiosity solution is executed.

Use Bounding Box: When this option is selected, the part of the scene where radiosity calculations are performed can be restricted by a 3D bounding box. Using this bounding box can significantly speed up a radiosity solution. For example, if a scene consists of a building, but a rendering of just one rooms needs to be generated, there is no need to let radiosity compute the illumination of all the other rooms. In this case, the bounding box can be fit tightly around the room, which causes all areas outside the box to be ignored by radiosity. The bounding box is set up with the Radiosity Bound Box tool and is discussed in more detail in section 6.8.5.
The Options tab for Final Gather Support

In this mode the Options tab contains one main control, as shown in Figure 6.8.3.2.

Quality: If this slider is set to a low value, the radiosity mesh is coarse and allows the solution to progress very quickly. However, the illumination generated is not very accurate and details may easily be missed. If the slider is set to a higher value, the mesh becomes denser. More illumination detail can be generated, but the radiosity solution progresses slower. A practical range is 50% to 75%, which works well for most scenes.

When using radiosity with Final Gather Support, which is its primary intended use, the objective is to spend as little time as necessary in processing a radiosity solution, while letting it distribute the maximum amount of light in the scene. Recall, that the final gather process may optionally use a radiosity solution as input, in addition to its own illumination calculations. Final gather works best when it has ample light to work with. Consider a scene with a room that has a small window through which the sun shines and causes a small patch of light on the floor. For final gather this is a tough scenario to work with, since the majority of the surfaces in the room do not receive any direct light. A radiosity process, however, is able to further distribute the light reflected off the floor onto the walls and ceiling, back on the floor. With this extra light, the result of a final gather rendering will be much improved. For this radiosity solution it is not important to generate accurate shadow boundaries, since the solution will never be rendered directly. It is more important for the radiosity process to spend its allocated time in distributing as much light in the scene as possible. A few simple steps can be taken to ensure that this is the case:

- Any objects in the scene that are small, but are defined by many polygons, should be set to not participate in the radiosity solution. The amount of light reflected off these object is minimal and the light received by the objects usually does not make much difference in the final rendering. This can be achieved with the Radiosity Attributes tool. Set the Object Radiosity option to No Receive And No Bounce and apply the tool to the objects in question. An example for this would be the silverware and plates sitting on a table in a room, when the entire room is rendered. A significant speedup in the radiosity solution can be gained. Objects, that are a bit larger and would show some effect from the light hitting them can be set to only receive light, but not bounce any back into the scene. Set the Object Radiosity of those objects to Receive Only. This too will speed up the solution, but preserve some illumination for those objects.

- If the image rendered shows only a portion of a larger scene, the Radiosity Bounding Box tool should be used. If the box is fit tightly around the visible part of the scene, such as a room inside a larger building, radiosity only computes the light that bounces inside the box, not in the invisible part of the rest of the model. This also can lead to a significant speed gain.
• The radiosity mesh can be kept at a rather coarse resolution, but should be fine enough to provide reasonable accuracy for the distribution of the light. Recall that the reflected light is stored at the vertices of the radiosity mesh patches. If the mesh is too fine, much time will be spent illuminating all the patches, without seeing any benefit in the final gather rendering. If the mesh is too coarse, the light is not distributed well enough and the illumination quality in final gather suffers. The Quality slider bar in the Radiosity Options dialog directly controls the mesh density. An example of a mesh density that is too fine, too coarse and appropriate, is shown in Figure 6.8.3.3 (a), (b), and (c).

• Usually, a radiosity solution suitable as input for final gather can be computed in less than 5 minutes, sometimes even in less than a minute. A good way to check whether the solution contributes any light, is to perform frequent, quick test renderings. The default setup in the Radiosity Options dialog is to use radiosity in Final Gather Support mode, let it run for 2 minutes, and perform a preview rendering every 20 seconds. This setup works well for average scenes. It is recommended to set the rendering mode to Shaded Render or Interactive Shaded before the solution is executed, to get visual feedback about the illumination. It is quite possible that, even before the 2 minutes processing time is over, the preview renderings show that no significant further illumination improvements can be achieved. It is fine to stop the solution at this point by hitting the Cancel button in the progress bar. It is also possible that the 2 minutes are not enough. Radiosity may spend time calculating the illumination in parts of the scene that are not relevant in the image to be rendered. If this is the case, the preview renderings will not show any increase in brightness. More time may have to be added and bounding box and object radiosity attributes may be used to speed up radiosity to progress faster to the part of the scene that is of interest. Naturally, scenes that are very complex need more time to process. The times listed here are guidelines for moderately complex scenes with no more than 20,000 radiosity mesh mesh patches.

Figure 6.8.3.3: Mesh density that is too fine (a), too coarse (b), and correct (c).
The Options tab for Render Direct Illumination

The radiosity process is well suited to compute the distribution of light once it is reflected from the surfaces. A low to medium density mesh is usually sufficient to capture this secondary illumination, since boundaries of sharp contrast do not occur during this phase. On the other hand, direct illumination is best computed by traditional raytrace and z-buffer algorithms. Raytraced shadows or mapped shadows can create crisp and soft shadow boundaries efficiently. This split up of tasks can be enabled by choosing the Render Direct Illumination item from the Mode menu. In general, the parameters guiding the adaptive meshing can be reduced to a much lower setting without losing the quality of the solution, if this option is selected. In many cases it will be possible to turn adaptive meshing off, especially, if the initial mesh is sufficiently dense. Once the radiosity process has been started, only the illumination which occurs from light reflected off all the surfaces in the scene is computed. When a Shaded Render or RenderZone image is produced from that solution, direct illumination, including the calculation of shadows, is performed by the rendering. In order for shadows to occur, the Shadow option must be selected in both, the Radiosity Options dialog and the RenderZone or Shaded Render Options dialogs. Figure 6.8.3.4(1) shows an image rendered with the Render Direct Illumination option selected. Note that although the radiosity mesh is not overly dense, shadow boundaries are still crisp. Figure 6.8.3.4(2) shows the same scene rendered with Pure Radiosity. Note the need for increased mesh density to maintain crisp shadow boundaries.

![Image 1](image1.png) ![Image 2](image2.png)

*Figure 6.8.3.4: Render Direct Illumination (1) and Pure Radiosity (2).*

(a) The radiosity solution mesh, and (b) the resulting rendering.
When using the **Render Direct Illumination** option, one can choose whether area and line lights are included in the direct illumination. If the **Include Area/Line Lights** option is not selected, the energy emitted by area and line lights is still computed during the radiosity solution. If the option is on, the direct illumination of area and line lights is also calculated at rendering time. In general, area and line lights do not cause sharp shadow boundaries, since their light is emitted in a diffuse fashion. Therefore, it is usually more efficient to let area and line lights be processed entirely by the radiosity solution. Only if artifacts in shadow boundaries from area and line lights are visible, the **Include Area/Line Lights** option should be used. As discussed in section 6.6, computing the direct contribution of area and line lights may reduce rendering speed significantly.

In this mode, it is possible to apply radiosity by either using the **Quality** slider only or by setting detailed parameters. Changing the slider sets several of the detailed parameters automatically. The detailed parameters can be displayed by selecting the **Show Details** button, as shown in Figure 6.8.3.5. The button changes to **Hide Details**, which can next be used to hide the parameters.

**Initial Meshing**: This group of options controls how the mesh that is the basis for the radiosity solution will be generated. This mesh consists of two separate parts whose densities may be set independently from each other. The first part describes the geometry of all objects and area lights. It is called **patches**. The patches are generated when the radiosity process is initialized and their number stays constant from then on. The second part is the actual **radiosity mesh**, which is a finer subdivision of the patches and may grow in density as the solution progresses. The light intensities that are calculated by the radiosity process are recorded at the points of this radiosity mesh, which is also referred to as the **recording mesh**.

**Patches**: This slider bar is used to specify the density of the patches. When the slider bar is all the way to the left, the minimum number of patches for surfaces is generated. The highest number of patches is created by moving the slider bar all the way to the right. The percentage of patch density can also be entered in the numeric field.
**Polygons n Times # Of Patches**: The number typed in this field specifies the density of the radiosity mesh at initialization. Since \( n \) is a multiplier, if the number is 1, the mesh density is equal to the number of patches. As the number increases, the mesh density increases.

While the patches are able to emit light, the subdivided mesh is only able to record light intensities. In general, it is not necessary to create a high density for the patches which emit light. However, to show sharp shadow boundaries or a rapid falloff in light intensity on a surface, the mesh which records the light intensities may have to be very dense to show visually acceptable results. The additional density of the recording mesh can be achieved in two ways. First, the **Polygon n Times # Of Patches** parameter can be used to create an initially high mesh density. Second, the **Adaptive Meshing** option may be used to increase the mesh density during the execution of the radiosity solution in areas of strong intensity contrast. During a single iteration of the radiosity solution, light is emitted from several points of a source patch and is recorded at each mesh polygon. There is a direct relationship between the density of the patches and the quality of the illumination. The denser the source patches are, the higher the quality of a rendering will be and the more computing time is required.

**Estimate Mesh Density...**: Before a radiosity mesh is initialized, the number of patches and mesh polygons can be estimated by clicking on this button. Note that this will only be a rough estimate. It is possible that the actual number of polygons is twice as large, or only half of the estimate.

**Adaptive Meshing**: When this option is turned on, the mesh may be subdivided into a finer resolution than its initial density. To what extent this subdivision occurs is determined by the next slider.

**Quality**: This parameter controls the level of subdivision of the adaptive meshing. Although the **Render Direct Illumination** method does not need an overly dense mesh for a decent rendering, some increase of mesh density along shadow boundaries is helpful. The higher this parameter is set, the denser the mesh will become, at the cost of longer processing time.
**Sampling**: This group of parameters determine the quality of the illumination performed during radiosity processing.

**Minimum #, Maximum #**: When light leaves a source patch, it is emitted from a number of randomly chosen points on the patch surface. The minimum and maximum number of sample points are entered in these fields. Note that large values entered in these fields may unnecessarily extend the progress of a solution. As a general guideline, minimum values between 1 and 5 and maximum values between 10 and 20 create acceptable results at a reasonable speed. How many points are sampled across the patch surface ultimately depends on the item selected from the **Quality** menu (see below) as well as other factors such as the size of a patch.

**Quality**: This pop up menu contains items which set the radiosity parameters to a predefined level of quality. The quality of a radiosity solution and the speed at which it is derived are inverse proportional to each other. That is, the lowest quality setting derives a solution quickly, but renderings based on such a solution are usually of poor quality. The highest quality setting takes longer to generate a solution, but the quality of images is higher.

- **Simple**: When this item is selected, the minimum number of samples is used.
- **Better**: When this item is selected a number of samples between minimum and maximum is used, but is more often closer to the minimum than the maximum number.
- **Best**: When this item is selected the number of samples used is more often closer to the maximum value.

While the previous options in the **Sampling** group determine the density of rays that leave the light emitting patches, the next two options control where the light intensities are recorded at the receiving patches.

**At Vertices**: When this option is selected, the light emitted from a primary or secondary source is recorded individually at each of the vertices of a mesh polygon. Given that all mesh polygons are triangles, the light calculation is performed three times.

**At Center**: When this option is selected, the light emitted from a primary or secondary source is recorded once, at the center of a mesh polygon. Center sampling, while faster, may require a denser mesh to avoid certain illumination artifacts. On the other hand this option generates softer shadows, which decreases the need for a dense mesh at shadow boundaries. Which of the two methods is more suitable depends greatly on the scene. In general, vertex sampling is more accurate, but may generate a denser mesh, whereas center sampling is faster at the expense of some illumination artifacts.
Include Area/Line Lights: When this option is off, the energy emitted by area and line lights is still computed during the radiosity solution. If the option is on, the direct illumination of area and line lights is also calculated at rendering time. In general, area and line lights do not cause sharp shadow boundaries, since their light is emitted in a diffuse fashion. Therefore, it is usually more efficient to let area and line lights be processed entirely by the radiosity solution. Only if artifacts in shadow boundaries from area and line lights are visible, this option should be used. As discussed in section 6.6, computing the direct contribution of area and line lights may reduce rendering speed significantly.

Light Intensity, View Dependent Solution, and Use Bounding Box: These options were discussed in "The Options tab common parameters" subsection earlier in this section.

Apply Reflected Light As Ambient: At any given time of the radiosity solution, only a percentage of the total amount of light has been assigned to the radiosity mesh polygons. Because of this and especially in the earlier stages, when rendering a radiosity solution, it is typically underexposed and appears dark. This may raise the need to temporarily add light, which is done with this option. When this option is on, the amount of light that has been emitted by all primary sources, but has not yet been recorded at the mesh polygons, is applied as a constant ambient light to all mesh polygons. During the early stages of a radiosity solution, this option will cause a scene to appear brighter than the final result. On the other hand, if this option is turned off, a scene only shows effect from light that has been recorded at the mesh polygons so far. For example, if a radiosity solution is 60% complete, only 60% of the light is rendered in a scene, which consequently looks darker than the final result.

Figure 6.8.3.6 shows two images based on a solution which is about 40% complete. The first image is rendered with the Apply Reflected Light As Ambient option turned on, whereas the second image does not use this option.

![Figure 6.8.3.6: Two images rendered with the Apply Reflected Light As Ambient option (a) on and (b) off.](image)
The Options tab for Pure Radiosity

When this mode is chosen, all detailed radiosity parameters are accessible to the user. As with the other modes, a simple Quality slider by default hides the detailed parameters. It is assumed that this mode will only be selected in rare occasions, when an experienced user needs to execute a radiosity solution for very specific purposes. All detailed parameters are displayed when pressing the Show Details button. The Options tab for the Pure Radiosity mode with the detailed parameters displayed is shown in Figure 6.8.3.7.

All the detailed parameters for Pure Radiosity are the same as for Render Direct Illumination except that Adaptive Meshing is now specified by more detailed parameters, as opposed to the single Quality slider that is used by Render Direct Illumination.

Adaptive Meshing: When this option is turned on, the mesh may be subdivided into a finer resolution than its initial density. To what extent this subdivision occurs is determined by the next three parameters.

Density: This slider bar indicates the maximum density the subdivisions can reach. The further to the right the slider bar is moved, the finer the subdivisions may become. Values of 50% and below will generate coarse subdivisions. Values above 80% should be used with caution, since they may quickly generate an excessive amount of mesh polygons, which may unnecessarily slow down the radiosity solution. Values around 60% are a good compromise between speed and quality, while values between 70 and 80% are appropriate for high quality images. Two pairs of a wireframe and shaded image are shown in Figure 6.8.3.8. The first pair uses a density setting of 60%, whereas the second uses a density of 80%. Note the difference in mesh density and the resulting image quality.
Threshold: This parameter determines when a mesh polygon will be subdivided, due to the difference of intensities at adjacent polygons. If the difference between the intensities of a mesh polygon and all its neighboring polygons is higher than this value, the polygon is subdivided. In other words, the lower the Threshold value, the more subdivision will occur.

Cutoff: This parameter can be used to avoid subdivision of mesh polygons in dark areas. Since an image does not show a great amount of illumination detail in underexposed areas, a dense mesh may not be necessary. A low Cutoff parameter causes mesh subdivision in dark areas and hence causes a denser mesh. Mesh density may be decreased by increasing the Cutoff parameter, which causes areas of increasing brightness to not be meshed.

Adaptive meshing can be specified globally for all objects in the scene or on a per object or face basis. In many cases a dense mesh may be required in a localized area of the scene, while the rest of the scene can achieve satisfactory results at a much lower resolution. This is, for example, the case when a cone light shines on the floor of a room. The mesh polygons around
the edges of the cone illumination need to be meshed densely to account for the rapid falloff of intensity in this region. The rest of the scene may be meshed at a much lower resolution than the floor, since the light reflected off the floor only generates very gradual light distribution on the walls and ceiling. Adaptive meshing alone already addresses this issue by meshing the area of sharp contrast more than other areas. The meshing can be even further reduced by assigning to objects or faces, which are known not to require extensive meshing, their own lower meshing parameters; or vice versa. That is, it is possible to specify low meshing parameters for the entire scene, and then assign higher meshing parameters to objects or faces which are known to require dense meshing.

One of four mutually exclusive options may be used to automatically adjust the sensitivity of the subdivisions as the radiosity solution progresses. A dense mesh is usually only necessary when light from primary sources is absorbed by the mesh, because the light intensity is relatively high and may generate sharp shadow boundaries. Light reflected from secondary sources is typically much lower in intensity and is largely responsible for a soft and gradual falloff of light on other surfaces. A lower density mesh is sufficient to display those intensities. One way to address this issue would be to let a solution progress with a certain set of parameters generating a dense mesh, while primary sources are handled. As soon as this stage is over, one could interrupt the solution, set the meshing parameters to values that will generate a much coarser mesh and restart the solution. This effect can be controlled automatically by selecting one of the following options.

**Continuous Meshing:** When this option is selected, the mesh parameters stay the same during the entire length of the radiosity execution. While this guarantees the highest quality results, the mesh density may be higher than necessary in areas of low contrast, and the radiosity process may be unnecessarily extended.

**Decrease Meshing:** When this option is selected, the mesh parameters are proportionally relaxed as the solution progresses. That is, the closer the solution has advanced toward full accuracy, the lower the meshing parameters become. This can be done safely, because towards the end of a solution process, light intensities reflected from secondary sources become increasingly lower, allowing for a lower density mesh. This option is a good compromise between speed of a radiosity solution and quality of the resulting images.

**No Meshing After Processing Lights:** The decrease of meshing parameters can be taken even one step further. When this option is selected, meshing is turned off completely, after all primary sources have emitted their light. In many cases, this will generate acceptable image quality, at the lowest possible mesh density.

**No Meshing After n Iterations:** Meshing can also be stopped after a given number of iterations. Recall that one iteration is defined by a single source patch, primary or secondary, emitting light to all polygons in the mesh. The value for the number of iterations to be completed before meshing is turned off is entered in the text field.
**Termination tab**

In this tab, three methods of controlling the duration of the radiosity process are presented, as shown in Figure 6.8.3.9.

% Of Light Absorbed: When this option is selected, the radiosity calculations terminate after a certain amount of light is absorbed. This amount is expressed as a percentage entered in its field. It indicates how much of the light, which is reflected after it was initially emitted by all primary light sources, must be absorbed by the radiosity mesh before the solution is stopped. Typically, a solution very quickly advances to a level of about 30 - 40% and takes exponentially longer as it progresses towards 100%. It is during the last 50% when the subtle change of illumination of indirectly lit surfaces is generated, which gives radiosity based rendering their realistic look. On the other hand, a solution which is 90% complete may sometimes hardly show any difference from a solution which is 80% complete. While the solution which is 90% complete may take a long time to finish, an 80% complete solution may be generated much faster. It is up to the user to decide how much time can be invested toward arriving at a visually acceptable result.

Iterations: When this option is selected, the radiosity calculations will terminate after a certain number of cycles, which is determined by the number entered in its field. While this method does not give an indication of completion, it may be more suitable for determining how long one is willing to let a solution execute. For example, if a solution executes 10 minutes, and during this time 1000 iterations are completed, one can interpolate the time that will be required for a given amount of iterations.

Time: When this option is selected, the radiosity calculations will terminate after the given amount of time. This option is on by default and is set to 2 minutes. It works well for an average scene when radiosity is used with **Final Gather Support**.
**Preview tab**

The **Preview** tab, shown in Figure 6.8.3.10, contains the options that determined how a radiosity solution and rendering may be previewed at different stages during its generation.

![Figure 6.8.3.10: The Preview tab in the Radiosity Options dialog.](image)

During the execution of a radiosity solution, it is important to receive visual feedback about the status of the current solution. The preview may be drawn in **Wire Frame**, **Interactive Shaded**, **Shaded Render**, or any plugin renderer that supports radiosity. Which type is used is determined by the currently selected display type in the **Display** menu. If any other display type is active when **Initialize Radiosity** or **Generate Radiosity Solution** is selected, it changes to wire frame. The preview is drawn in all open windows of the current project, allowing the preview of multiple views simultaneously. Whether the image is updated during the radiosity process is set in the **Preview** tab.

**Update Display**: When this option is on, a preview of the radiosity solution is produced as the radiosity progresses. How frequently it is updated is set by the following parameters.

- **Update Every $n$ Seconds**: This option causes the radiosity preview rendering to be updated every $n$ seconds, where $n$ is entered in the field of this option.

- **Update After $n$ Iterations**: When this option is selected, the frequency of the updates is determined by the number of iterations, which is entered in the text field of this option.

- **Update Each $n$ % Of Completion**: When this option is selected, a preview is drawn each time the specified amount of completion percentage has been reached.

- **Update After Each Primary Light Source**: When this option is selected, a preview is drawn after each primary light source has been processed.

Generating a preview rendering takes time away from the computation of the radiosity solution. It is therefore not advisable to set the time below 10 seconds, since in such a case more time would be spent on drawing than generating the solution.
Statistics tab

Given that the radiosity process is best executed in an iterative manner, it is frequently useful to know how far the radiosity solution has progressed at a given time. This is provided by the statistics contained in the Statistics tab, as shown in Figure 6.8.3.11. Note that these are simply informational fields and their content cannot be edited or changed.

**Absorption**: As a radiosity solution progresses, more and more light is “absorbed.” That is, it is incorporated in the intensities attached to the radiosity mesh. This statistic displays the quantity of the absorbed light as a percentage of the total amount of light that would be absorbed by a 100% complete solution. The amount of absorbed light is measured as soon as the light of all primary sources has been emitted.

**Iterations**: The radiosity process is iterative and this statistic displays the number of iterations completed at the current stage of the radiosity solution.

**Time Spent So Far**: This field shows how much time has been spent calculating the radiosity solution.

**Patches**: This field displays the number of patches generated by the radiosity initialization.

**Polygons**: The radiosity solution is based on a progressive and recursive subdivision of all surfaces, resulting in the radiosity mesh. How many there are at a given time is displayed by this statistic.

These statistics can be read at any time the radiosity process is interrupted (stopped), but not exited. Recall, that a radiosity solution may stop by itself or it may be interrupted manually. When the Radiosity Options dialog is opened after such an interruption, the statistical values are shown in their respective fields. Before a solution is initialized, all these fields show zero values. It is possible to alter most of the parameters in the Radiosity Options dialog after a solution has been interrupted and even after it has reached its final goal. For example, if it turns out that after the initially specified 500 iterations the image generated from the solution is not satisfactory, it is possible to change the number of iterations to 1000 and let the solution continue.

*Figure 6.8.3.11: The Statistics tab in the Radiosity Options dialog.*
6.8.4 Setting the radiosity attributes

This tool sets two radiosity related attributes. The first, Object Radiosity, applies to objects only. It determines whether and how objects participate in a radiosity solution. This attribute can be used effectively to make radiosity more efficient and works well when radiosity is used in the Final Gather Support mode. The second, Radiosity Meshing (Object/Face) applies to objects and faces. It is less useful for final gather support, since it increases or decreases meshing on a per object or face basis. Meshing control is more critical for the Render Direct Illumination and Pure Radiosity modes. When radiosity meshing has been applied to individual entities (objects or faces), these can be used instead of the global meshing conditions specified in the Radiosity Options dialog, when a radiosity solution is derived. The meshing parameters are set in the Radiosity Attributes Options dialog, shown in Figure 6.8.4.1, which can be invoked directly from the tool.

Radiosity Attributes

After the proper meshing parameters have been set, this tool is used to assign them to an object and/or face. The options that can be set in its dialog are as follows:

Object Radiosity: This is a menu with three items, which apply to the object topological level.

Receive And Bounce: When this item is selected, the object fully participates in radiosity. That is, the object is receiving indirect light and light is allowed to bounce off the object to illuminate other objects in the scene.

Receive Only: When this item is selected, the object is illuminated by indirect light, but no light is reflected off the object back into the scene.

No Recieve And No Bounce: When this item is selected, the object is neither illuminated by indirect light nor does it reflect any light back into the scene. The object is effectively excluded from radiosity processing.

The latter two options are designed to further optimize the progress of a radiosity solution. For example, a scene may depict a room with some furniture. A few thousand faces may be used to define the objects. In the middle of the scene may be a small statue that is modeled with several thousand faces. If a radiosity solution were to be executed in that scene, much time would be spent computing how much light is received and reflected by the statue. A final rendering of that scene, however, may not show any significant visual improvement from the extra light calculation on the statue. It would be far more efficient to spend the radiosity time on computing the light reflecting off the room and furniture surfaces. This can be achieved by assigning the No Recieve And No Bounce attribute to the statue.
In the final rendering, the statue will only receive direct illumination, whereas the room and furniture will have indirect illumination. The **Receive Only** option is a compromise setting. It can be used to show indirect illumination on an object, but time can be saved not calculating light bouncing off the object. Medium sized objects with a large amount of faces will benefit from this setting.

**Radiosity Meshing**: When this option is selected, the other parameters under it can be set and applied to the desired topological level (Object or Face) using the Radiosity Attributes tool. To cancel the radiosity meshing attribute from an object or face, deselect **Radiosity Meshing** and reapply the tool.

**Adaptive Meshing**: This option can also be selected when **Radiosity Meshing** is on. When selected, it works as when set in the **Radiosity Options** dialog and is specified by the same parameters **Density** and **Threshold**. What is significant is that it is possible to completely exclude selected objects/faces from adaptive meshing, while it is applied to the rest of a scene, and vice versa. That is, the ability to apply radiosity meshing globally, individually, or by combining the two offers extensive flexibility in controlling the level of the radiosity solution. The **Radiosity Meshing** of objects and faces can also be assigned and edited from the **Query Attributes** dialog.

Note, that the **Layer Attributes** dialog, shown in Figure 6.8.4.2, has an override option that allows all objects on a specific layer to receive the same radiosity attribute, if that option is checked. The **Layer Attributes** dialog is invoked from the Layers palette. This is discussed in more detail in section 2.7.2.

![Layer Attributes dialog](image)

**Figure 6.8.4.2**: The **Layer Attributes** dialog.
6.8.5 Specifying the extent of a radiosity rendering

Radiosity Bounding Box

With the Radiosity Attributes tool specifically selected objects can be excluded from a radiosity solution. With the Radiosity Bounding Box tool one can define a 3D box, that executes radiosity processing only on objects that are contained inside the box. For example, if a rendering of a lobby of a high rise building is produced, there is no sense in calculating the radiosity solution of any of the rooms not visible in the rendering. In this case, the bounding box would be sized in such a way, that it includes the lobby and all its walls, floor and ceiling, but none of the objects outside of it.

To execute this tool, activate it and click anywhere in the project window. This will display a 3D box in the ghost color with six handles (arrows). Clicking inside one of the arrows and dragging it with the mouse button pressed, resizes the box in that direction. Releasing the mouse button stops the resizing. Alternatively, an arrow can be selected with a single click, a second click inside the arrow starts the resizing, and a third click stops it. These two methods are consistent with editing parameteric object controls using the Edit Controls tool. The editing of the box is terminated by clicking anywhere away from the arrows. If the current rendering mode is Interactive Shaded, the box is drawn transparent so that one can see what is inside the box. As soon as the editing mode of the box is active, a palette is also displayed that contains the min and max x, y and z extents of the box. These values update automatically as the box is resized. The min and max dimensions can also be entered manually by typing new values in the fields. Hitting the Fit To Scene Extents button in the palette automatically resizes the box to include all objects in the scene.

The best way to execute this tool is to (1) open a new modeling window, (2) change the view to Axonometric, and (3) Fit All from the window tools palette. Next, size the box to include the desired parts of the scene. It should also be helpful to switch to a top or side view as appropriate to be able to precisely align the box.

For a radiosity process to use the bounding box, the **Use Bounding Box** option must be selected in the **Radiosity Options** dialog or in the **Final Gather** tab in the **RenderZone Options** dialog. An example of the radiosity box being edited is shown in Figure 6.8.5.1.

![Figure 6.8.5.1: The radiosity bounding box while being edited.](image)
6.8.6 Lights in radiosity

Setting up lighting correctly is one of the keys to generating a useful radiosity solution. In non-radiosity based renderings one often simulates the reflected light energy through the global ambient light or more specifically through fill lights. Such fill lights do not exist in the real world but are added to a scene to provide additional illumination in areas that are not lit directly. Naturally, fill lights are not necessary when computing a radiosity solution since radiosity generates the reflected light accurately.

In general, a radiosity solution is used for two major purposes. First, it provides the base for accurate reflection of light in a scene on top of which very realistic looking images can be rendered using traditional rendering techniques. As such, radiosity becomes a visualization tool that enhances the quality and accuracy of photo realistic images. Second, a radiosity solution can be used as an analytical tool. For example, it is possible to visualize the illuminance of a surface (that is, how much light is received by a surface in a unit square). The illuminance can be displayed in a color coded rendering which is useful for lighting design and light measurements. Depending on how a radiosity solution is intended to be used, lights may have to be set up differently. For generating visually pleasing images, it is not necessary to specify the output energy of a light source in physically accurate units. It is enough to use the simple intensity option, which tends to be more predictable. For illumination studies, of course, it is necessary to specify the light intensity accurately, if a correct light analysis needs to be generated.

By default, the light intensity of a light source is specified as a simple percentage. 100% intensity means, that surfaces illuminated by this light source display their true color in the image, if the light shines on them perpendicular and the surface’s diffuse reflection is also 100%. For constant falloff this is the case for any surface. For linear and square falloff, the true color is reached at the light’s radius as set in the Light Parameters dialog. Given this rule, it is fairly simple to predict the brightness of a scene. If accurate intensities are used, it is more difficult to predict the illumination of a scene in a final rendering. For this purpose, the Exposure Control post processing effect has been added to correct over or underexposed images. This is described in further detail in section 6.1.10.

All types of lights are available in radiosity. Depending on which kind of lighting effect is desired, a different light type needs to be used. This is discussed in detail in the following sections.
Distant lights

A distant light is normally used to simulate the sun light. With the simple intensity option, the light’s output can be set and estimated as with the other light types. When using the **Accurate Intensity** option, it is important to note that the default values are set to represent an average bright sunny day. Since the default exposure sensitivity of a rendering is set to interior lighting conditions, a radiosity solution generated with a distant light using the default accurate intensity tends to be overexposed. This is especially noticeable when rendering an interior that has large openings through which sunlight is entering.

![Image](image.png)

*Figure 6.8.6.1:* A scene rendered with a distant light using atmospheric effects.

When using the Accurate Intensity option for a distant light, the Atmospheric light option can be used, as described in section 6.6.3. The atmospheric light is non directional and generates soft shading. This effect, however, requires additional computing and may slow down the generation of the radiosity solution significantly, especially if the quality is set to a high value. Soft light shining through a window can be simulated more efficiently through the use of area lights, as described below. The example in Figure 6.8.6.1 shows a room with a window. Even if there is no light shining directly into the room, the environment still submits a considerable amount of light into the room, if a scene is rendered with a distant light using atmospheric effects. The distant light does not shine directly into the room. Illumination is generated only by the atmospheric light.
Area/line lights

While area and line lights can be used in renderings that are not based on radiosity solutions by consuming extra rendering times, they are much more efficient when used in a radiosity solution. Two different types of application are described here. First, area and line lights can represent light fixtures which emit light from a surface or a linear element. For example a neon tube can be modeled as a line light or a lamp shade as an area light. This is illustrated in Figure 6.8.6.2. Second, area lights can very efficiently simulate diffuse daylight which enters a room through a window. The same effect can also be achieved by using the Atmospheric Light option of distant light sources, as described before. However, since atmospheric light is present everywhere in the hemisphere, the illumination of each surface in the scene needs to be calculated, which, while more accurate, is also computationally more expensive. An area light used as a window pane for example, limits the diffuse light emission to a small portion of the scene, therefore only computing illumination where needed, resulting in faster processing times. The same example used for the atmospheric light is shown in Figure 6.8.6.3, but using area lights in the window openings instead of a distant light with the atmospheric light option.

**Figure 6.8.6.2:** A lamp rendered using an area light.

**Figure 6.8.6.3:** A scene rendered with area lights using atmospheric effects.
**Cone/point lights**

These two light types are most commonly used for simple interior light fixtures. Cone lights work well for lights with a limited illumination volume, such as recessed lights, desk lamps, or stage lights. Point lights should be used for all light fixtures, where light is emitted equally strong in all directions. For example, candles, hanging lamps and light bulbs can be represented by point lights. As with distant lights, the intensity of these lights can be specified as a simple percentage or through accurate units. Again, if the objective of the radiosity solution is to generate a visually pleasing image, it may be easier to set up the lighting using the simple intensity. When using accurate intensity, the exposure sensitivity of the rendering is set by default so that an interior scene with average illumination will give reasonable well exposed images. It may still be necessary, however, to use the **Exposure Correction** post processing feature to fine tune the final image.

**Projector lights**

If a projector light is used in a radiosity solution, it is processed like a cone light. That is, the light volume is conical in shape and the image is not projected nor do the colors of the image influence the color of the reflected light in the scene.

**Custom lights**

As described in section 6.6.6, custom lights can be used to represent light sources, where light is not emitted uniformly in all directions. Many manufacturers of light fixtures supply data files in IES, CIE or CIBSE format, which describe the intensity at a given direction of a light fixture. For accurate radiosity solutions, such light information can be used to set up custom lights. Although custom lights do not cause a significant slowdown in processing time, the visual difference between a custom light and a point light is often minimal. That is, for radiosity based renderings which aim at creating pleasing images, it is usually sufficient to place point lights instead of custom lights.
Dimming lights

In order to sufficiently illuminate a scene, often times, many light sources must be placed in it. To control the overall exposure of the scene, it may be necessary to increase or decrease the output of all light sources. One way to achieve this would be by changing the intensity setting for each individual light source, before generating the next radiosity solution. It is obvious, that this becomes very tedious for a scene with many lights. A more efficient way would be to gather the lights into logical groups, using the light grouping mechanism, as described in section 2.10. For example, all recessed lights operated by one wall switch can be grouped. In addition, light groups may be nested, allowing all groups to be placed in a single group. A light group can then specify an intensity overwrite parameter, which scales the intensity of each light in the group by a given factor.

Through the grouping mechanism, the illumination of, for example, all ceiling lights in a room can be controlled independently and the overall illumination may be controlled through the top level group. Such a setup is illustrated in Figure 6.8.6.4. Figures 6.8.6.5(a) and (b) show the Light Group Attributes dialog for a nested light group and the top level light group, both with the Scale Intensity overwrite option on.
6.8.7 Surface styles in radiosity based renderings

When the radiosity process computes the reflection of light off a surface, it is affected by two parameters contained in the definition of the surface style assigned to a surface: the color and the diffuse reflection factor.

Under certain illumination conditions, the color of a surface is reflected on neighboring surfaces. For example, a red table top in front of a white wall will reflect some of its red color on the wall when illuminated from above. This effect is known as color bleeding and is illustrated in Figure 6.8.7.1. The color is contained in the surface style definition assigned to a surface.

A certain amount of light is diffusely reflected from a surface. This amount is defined by the Diffuse Reflection factor, present in the standard Matte, Plastic, Chrome, Generic, Accurate Glass, Glossy, Accurate Metal, Mirror, Multilayer Paint and Environment reflection shaders. The Constant, Metal, and Simple Glass reflection shaders are assumed to have a diffuse reflection parameter of zero.

Radiosity assumes that the diffuse reflection is constant across a surface. That is, scaling the diffuse reflection via a reflection map is not taken into account during radiosity processing. As a general guideline, diffuse reflection values should be defined in a reasonable range to achieve realistic looking images. In the real world, diffuse reflection ranges from about 10% to 90% of the incident light. Diffuse reflection of 0%, where all light is absorbed, or 100%, where all light is reflected does not exist. Diffuse reflection values for some common surfaces are as follows:

<table>
<thead>
<tr>
<th>Surface</th>
<th>Diffuse Reflection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Painted Ceiling</td>
<td>60 - 90%</td>
</tr>
<tr>
<td>Floor</td>
<td>15 - 35%</td>
</tr>
<tr>
<td>Aluminum</td>
<td>60 - 70%</td>
</tr>
<tr>
<td>Walls</td>
<td>50 - 80%</td>
</tr>
<tr>
<td>Wood Surfaces</td>
<td>30 - 50%</td>
</tr>
</tbody>
</table>

As a general guideline, a high diffuse factor should be used for a surface style, unless the surface is highly glossy, in which case a high specular reflection factor should be specified.
In a radiosity model, rendering the effects of bump mapping are limited if the **Pure Radiosity** option is not selected. A rendering that uses the simple illumination model is able to simulate the surface irregularities, resulting from bump mapping. The bumps are created mostly from the diffuse reflection calculation at each pixel on the basis of an intentional disturbance of the surface normal direction at that pixel. These normal disturbances cannot be generated during a radiosity process, since radiosity simulates a process which accurately describes a physical phenomenon, whereas the normal disturbances are merely a trick to bend a surface at a pixel where in reality the surface is flat. In other words, the effects from the bump shaders used by a surface style will not be visible through diffuse reflection in radiosity based rendering. However, bumps will show up as a result of specular or mirrored reflections and refracted transparencies.

Figure 6.8.7.2 illustrates how bumps are displayed in radiosity based renderings. On the left is an image which is not based on a radiosity solution but was generated with the simple illumination. Bumps are visible on the wall surface, from diffuse reflection of the brick bumps. On the floor, the mirrored reflections are broken by the round bumps assigned to the floor’s surface style. The bumps in the area of the hotspots of the spheres are generated by specular illumination. On the right is the same scene, but rendered after a radiosity solution was generated. Note that the bumps on the walls, which were generated previously through diffuse reflection, have disappeared, but that the bumps on the reflective floor and on the spheres are still visible.

Again, this limitation only applies if the **Pure Radiosity** option is selected. For the other option, illumination from primary light sources is computed at rendering time on a per pixel basis, which enables bump mapping effects.

*Figure 6.8.7.2:* Displaying bumps (a) without and (b) with a radiosity solution.
6.8.8 Radiosity in form•Z Imager

The RenderZone Plus version of form•Z Imager can be used to generate radiosity based shaded render and RenderZone images. Since the Imager is not interactive, its primary use is for generating multiple high quality radiosity based images once the desired radiosity parameters have been established. For complete details on using the Imager, see section 3.12 of the form•Z User’s Manual.

The radiosity based renderings are selected through two items at the end of the Display Type pop up menu in the Imager Set window (Figure 6.8.8.1) and in the Default Image Type pop up menu in the Imager Set Up dialog (Figure 6.8.8.2).

Shaded Render Radiosity executes a shaded rendering based on radiosity, and RenderZone Radiosity executes a RenderZone rendering based on radiosity. For both of these renderings the radiosity calculations are performed prior to the rendering using the radiosity options stored in the project. When there are sequential radiosity based renderings from the same project in the Imager set list, the radiosity calculations are only performed once and then used for all the renderings for that project.
6.9 Radiosity tips

Radiosity is a complex process which typically requires fine tuning to produce desirable results. This section discusses a few techniques that will help you avoid some of the typical radiosity problems.

6.9.1 How to improve the radiosity solution

While “perfect” radiosity solutions are always possible by increasing the overall resolution of the radiosity mesh, this is typically not feasible, due to the processing times that extreme radiosity solutions would require. There are also techniques that can be employed to achieve desirable results without resorting to global resolution increases, as well as techniques for controlling the time required to complete a radiosity solution. Some of these techniques are discussed in the remainder of this section. However, they are only of concern when executing a radiosity solution under the Render Direct Illumination or Pure Radiosity mode. In Final Gather mode the radiosity mesh is never rendered and therefore a minimal mesh density is sufficient.

The radiosity solution takes too long

Despite all efforts to make radiosity as fast as possible, a radiosity solution may take a considerable amount of time to generate a satisfactory result, especially if a high image quality is required. This coupled with the iterative nature of the execution of the radiosity solution, where the calculations are repeated after adjustments are made to the parameters, may result in very lengthy computation times if not properly planned. The proper method here is to keep each stage of the iteration as simple as possible by only generating as much detail as necessary for evaluating a specific aspect of the radiosity solution.

When starting to generate a new radiosity solution, it is advisable to set the radiosity parameters in such a way that initially the solution progresses fast, but generates low quality images. While these images might not be usable as a final result, they are generally sufficient to tell whether the scene has been set up properly. For example, one will be able to examine whether the light sources are strong enough or whether the desired contrast in the scene has been achieved. As changes are made to the scene and the results become more and more satisfactory, it is possible to increase the quality of the radiosity solution, at the expense of a longer processing time.

One of the secrets to a fast radiosity solution is to keep the density of the mesh as low as possible and still generate a radiosity based rendering of a desired quality. A number of tricks can be applied to achieve this and are discussed in the following subsections.
Crisp shadows by manual meshing

Crisp shadow boundaries can always be achieved by increasing the resolution of the radiosity mesh. However, given that high resolution meshes require extensive time to compute, techniques other than simply increasing the resolution are more efficient. If a shadow boundary is a straight line, it helps to align the direction of the radiosity mesh with the direction of the shadow boundary. This can be achieved by applying form-Z’s Mesh tool to selected faces in such a way that the meshing direction and the direction of the shadow boundary match. Figure 6.9.1.1 shows such an example. Without applying the Mesh tool, the radiosity mesh has an arbitrary alignment and the shadow boundaries in Figure 6.9.1.1(a) show artifacts. In Figure 6.9.1.1(b), the floor on which the table stands has been subdivided with the Mesh tool. The direction of the mesh and the edge of the table align, which allows the radiosity mesh to follow the shadow boundary projected by the table top. In both cases the number of mesh polygons is about the same. Note that the second image shows less artifacts at the shadow boundaries.

Figure 6.9.1.1: Rendering shadows (a) without and (b) with aligning the radiosity mesh with the shadow boundary.
The next example is a case where we have no opportunity to align the direction of the mesh and the shadow boundaries since the latter has a curved shape. Consequently, other methods are applied.

Rough shadow boundaries are caused by a coarse radiosity mesh, combined with hard shadow edges. Such a situation is shown in Figure 6.9.1.2 (a) where a cone light is shining on a corner of a room. Increasing the radiosity mesh density or softening the shadow boundaries will reduce or eliminate the problem. Increasing the mesh density can be done on a global level by increasing the meshing parameters in the Radiosity Options dialog, or on a local level, by assigning meshing attributes to individual objects or faces by applying the Render Attributes tool. A denser mesh is illustrated in Figure 6.9.1.2 (b). Alternatively, the shadow boundaries may be softened without having to increase the mesh density. When using a cone light this can be done by assigning a smaller inner cone angle. Such a solution is shown in Figure 6.9.1.2 (c). Using area lights will also create softer shadows. More examples where rough shadow boundaries are avoided by proper selection of light type are discussed in the next section.

\textbf{Figure 6.9.1.2:} (a) Rough shadow boundaries; (b) denser meshing leading to crisper shadow edges; (c) softer shadows from a smaller inner cone angle.
Crisp shadows by proper choice of light type

Another way to keep the mesh density down is to avoid hard shadows whenever possible, which can be done by choosing the proper light type. Consider a table lamp, with a cylindrical lamp shade. One way to model this lamp would be to put a point light in the center of the lamp and let the lamp shade generate the shadow it casts on the scene. The shadow boundaries created by the top and bottom rim of the lamp shade would require a dense mesh to avoid artifacts. A rendering with such an approach is shown in Figure 6.9.1.3(a).

A better solution would be to put two cone lights inside the lamp, one shining straight up and the other shining straight down. Cone lights allow for a soft edge, by setting the inner cone angle a few degrees smaller than the outer cone angle. If the cone lights are lined up in such a way that they do not intersect the rims of the lamp shade, no shadows need to be generated. In addition the soft edge of the cone light will allow for a lower density mesh without visual artifacts. A rendering with this setup is shown in Figure 6.9.1.3(b).

Figure 6.9.1.3: Simulating soft shadows by using (a) a point light and (b) two cone lights.
A third method to avoid hard shadow edges would be to make use of area lights. Unlike all other light types, where light is emitted from a single point, area lights emit light from points randomly sampled across the faces of the area light object. As a result, the shadows caused by the light emitted from one point on a face may be softened by the light emitted by another point of the area light. Figure 6.9.1.4 shows the same lamp as before, except this time the lamp shade is converted into an area light. The image generated from the radiosity solution shows no visual artifacts, although the mesh density is the same as in the examples above.

Area lights may also be used efficiently to simulate the illumination of a room by an exterior window, if no direct sun light shines through the window. In this case, the area light represents the non directional light existing in the outside environment, which radiates non uniformly through the window into the room. Such an example is illustrated below in Figure 6.9.1.5.
Crisp shadows through direct illumination

In general, high quality images based on a radiosity solution require a very dense mesh in areas of sharp shadow boundaries to avoid unwanted artifacts. While there are several methods which can be employed to avoid sharp contrasts, it may be a better solution to divide the illumination task between radiosity and rendering. While radiosity is good at generating the subtle shading of secondary illumination, raytrace or z-buffer renderings deal well with crisp shadows. This hybrid approach can be used in form•Z by selecting the Render Direct Illumination mode in the Radiosity Options dialog. Figure 6.9.1.6(a) shows a scene with a low density radiosity mesh of about 10,000 polygons. If the radiosity solution is executed with this mesh and the Pure Radiosity mode is selected, visible artifacts appear as seen in Figure 6.9.1.6(b), but the processing time is short. Increasing the mesh density to about 60,000 polygons by choosing a higher quality setting for the adaptive meshing parameters (Figure 6.9.1.7(a)) yields a better image, as shown in Figure 6.9.1.7(b). However, at the same time, the radiosity solution had to run about three times longer to arrive at the same percentage of completion. In addition, shortage of memory resources may prevent the high density mesh from being created in the first place. The same high quality image can be created with a low density mesh, and by selecting the Render Direct Illumination mode. At the same time, the Adaptive Meshing parameters can be kept at a low quality setting or turned off completely. As the radiosity solution is processed, only secondary illumination is calculated. When rendered, the direct illumination, including shadows through raytracing or shadow maps, is calculated at rendering time. The mesh density in Figure 6.9.1.8(a) is the same as for the low density mesh in Figure 6.9.1.6(a) and it took the same time to complete the computation of the solution. However, when rendered, raytracing the shadows and calculating illumination values requires additional time. On the other hand, this extra time may be offset by the fact, that less polygons have to be rendered. In this example, the rendering time for Figure 6.9.1.8(b) was about the same as for Figure 6.9.1.7(b), where the high density mesh was used.

Figure 6.9.1.6: A scene using (a) low density mesh in Pure Radiosity mode and (b) the resulting rendering with visible artifacts.
Figure 6.9.1.7: The same scene with (a) high density mesh and (b) the resulting rendering with crisp shadows.

Figure 6.9.1.8: The same scene using (a) low density mesh and the **Render Direct Illumination** mode to create the (b) resulting rendering which is comparable to that seen in Figure 6.9.1.7(b).
6.9.2 Troubleshooting

Below is a list of problems which may arise and suggestions of how to deal with them.

Shadow or light leaks

A common problem of radiosity based renderings are shadow or light leaks. Such artifacts are illustrated in Figure 6.9.2.1(a). Note that at the area where the leg of the stand touches the floor, a dark shadow band is displayed.

When looking at the radiosity mesh of this scene, one can observe that the mesh polygons of the floor where the leg touches the floor are partially under the leg and partially visible. When light intensities are computed, one portion of those mesh polygons receives no light, whereas the other portion is illuminated. During the rendering of the scene, the light intensities are interpolated across the mesh polygons. Interpolating the dark color that is under the leg with the light color which is visible at the edge of the leg produces a shade that appears to be a leak.

The solution to this problem is to insert edges where objects touch. One way to achieve this in the example would be to union the leg with the floor. Alternatively, the Resolve Intersections option in the Radiosity Options dialog may be selected, which will automatically split intersecting objects. After this has been done, the mesh polygons are not partially obscured anymore and the resulting rendering does not show any more shadow leaks, as illustrated in Figure 6.9.2.1(b).

Figure 6.9.2.1: (a) Shadow leaks and (b) corrected leaks.
The radiosity mesh becomes too dense

It is possible that a radiosity solution very quickly generates a large number of mesh polygons. This in return will have a strong impact on the time it will take to arrive at an accurate solution. Dense areas of the radiosity mesh are necessary in regions which have sharp contrast in illumination, such as hard shadow boundaries. The answer to excessive meshing is to avoid over-meshing areas which do not exhibit sharp contrast, or which are too dark to see. A number of steps may be taken to achieve this. First the Threshold parameter for Pure Radiosity in the Radiosity Options dialog for global meshing or in the Radiosity Attributes dialog for object/face level meshing may be increased. Large values for this parameter will cause less subdivision. Figure 6.9.2.2(a) shows a scene with a dense mesh created with a Threshold parameter of 10%. Note that the mesh is dense even in areas where there is only little light contrast, such as in the center of the illuminated arcs. The same scene has been rendered based on a solution processed with a Threshold parameter of 25%, shown in Figure 6.9.2.2(b). Note that the Gouraud shaded renderings are virtually the same. However, the mesh of the second example contains only a fraction of the number of polygons of the first example.

A second solution to overmeshing would be to assign a denser mesh only to objects or faces which contain areas of sharp contrast, but to process the rest of the scene with a lower mesh. This can be done, by applying the Radiosity Attributes tool to those objects or faces and by setting stricter meshing parameter in the Radiosity Attributes dialog.

A third solution would be to generate a view dependent radiosity mesh. If renderings based on such a mesh are always created form the same view, areas in the scene which are not visible can be safely skipped from meshing. This can reduce the mesh density by a large amount, resulting in a significant speed gain.

Figure 6.9.2.2: (a) Overmeshing due to a low Threshold parameter. (b) Meshing corrected with an increased Threshold parameter.
The image appears over or underexposed

Radiosity models light distribution in an environment in a physically accurate manner. Given that the intensity of light sources may be specified in accurate units, such as Watts or Lumens, intensities recorded at mesh polygons represent a physical intensity unit. While the human eye is capable of adjusting to a wide range of light intensities, a display medium such as the computer screen is not. That is, while the intensities at the mesh polygons are physically accurate, when mapped directly on the screen, the inability of the screen to display very bright intensities or very dim intensities in a visible manner leads to over or underexposure of the image.

An image may also contain areas with both very bright or very dim areas. If the intensities are mapped on the screen in a linear fashion, the bright areas may appear overexposed, while the dim areas may appear underexposed. This can be corrected by mapping mesh intensities into the intensity range which can be handled by the computer screen. Selecting the Exposure Correction... button in the RenderZone Postprocess Options dialog invokes the Exposure Correction Options dialog, where the intensity range and mapping of physical intensities to screen intensities can be defined. An example of an image with a corrected exposure is shown in Figure 6.9.2.3.

Alternatively, it is possible to process a radiosity solution using the Simple intensity specification of the lights. While the resulting solution is not accurate in terms of calculated luminance values, it is generally easier to set up lighting conditions with simple intensities, which avoid under or overexposed images.

Figure 6.9.2.3: (a) Uncorrected and (b) corrected exposure of a radiosity based rendering.
The radiosity based image does not show any effects from bump mapping

Bump mapping is an effect which is created by the intentional disturbance of the surface normal direction during the shading calculation of the simple illumination model. If Pure Radiosity mode is selected, the accurate representation of diffuse reflection during the radiosity solution cannot take into account such surface disturbances. As a result bump mapping effects in a rendering are only created by specular reflections, mirrored reflections and refracted transparencies, which are still calculated on a per pixel basis during the rendering. However, if the Render Direct Illumination mode is selected, bump mapping effects will be generated as expected.

The progress bar does not advance during the execution of the radiosity solution

The execution of the radiosity solution is not a linear process. That is, the first 30% of the solution may be achieved very quickly. The amount of time it will take to advance from 30% to 40% may be much longer and even longer to get from 40% to 60%. As the solution approaches 100%, which it never will reach, more and more time will be spend to gain less and less accuracy. While it appears, that the progress bar does not advance, the computational effort is spent on calculating a minimal gain of the solution, which cannot be represented in the progress bar. This is especially the case, when the mesh density is high and the solution has advanced beyond an accuracy of 80%. In many cases, the visual gain of a solution which is more than 85% accurate is small, and the radiosity process may be interrupted manually by clicking in the Stop button of the progress bar.

The rendering based on a radiosity solution is all black

There are a number of reasons why this may occur. First, it is possible that the light sources are simply not strong enough to illuminate the scene. Recall that in the real world the intensity of light falls off with the square of the distance the light travels. That is, a point which is twice as far away from the light source than another point only receives a quarter of the intensity of the closer point. For example, if a candle is placed inside an auditorium, the walls of the space are simply too far away to receive any visible light energy. It is important to adjust the intensity of the light with respect to the size of the space which is illuminated. Unlike the simple illumination model, which allows for a constant, linear or square intensity falloff, the radiosity process always assumes square falloff of light intensities.

Another reason for non-illuminated scenes may be that light which shines from outside into a space through openings in the walls is obscured by objects in the openings. If a window is modeled with an object representing the glass surface, the object must have the shadow casting attribute turned off, even if the surface style used by the object has transparencies. Unlike the simple illumination model, radiosity currently assumes all shadows to be opaque.
6.10 Radiosity tutorials

There are three tutorials in this section. They give specific hint on how to create successful radiosity solutions, using daylight conditions and artificial lighting. Note, that these tutorials use the Render Direct Illumination and Pure Radiosity modes. While these modes are still supported, the primary use of radiosity is now Final Gather Support. A tutorial using radiosity for this purpose can be found in section 6.1.6.
6.10.1 A radiosity solution using sun light

The example shown in this section is meant to illustrate how to create a visually pleasing image, using only the sun as a light source. The example is divided into three subsections, which show how to progressively refine the radiosity solution, until a rendering can be generated that looks like the image in Figure 6.10.1.1.

Finding the correct light intensity

In this example we will be using the simple intensity for the radiosity solution. In form•Z, the default exposure sensitivity is set to give correctly exposed images when using accurate intensities for interior lighting. Since sun light is several orders of magnitude stronger than interior lighting, a radiosity solution generated with the default values for accurate intensity of distant lights usually gives overexposed renderings. In this example, we are not interested in accurate illumination values and it is therefore easier to use the simple intensity specification.

Open the file “Distant Light.fmz.” It contains the scene shown in Figure 6.10.1.1, except that the layers that contain all the interior decoration are turned off. To find the correct light intensity it may be necessary to produce a number of quick radiosity solutions. This can be done without the interior decoration, since the objects are rather small compared to the wall surfaces and therefore their contribution of reflected light is minimal. However, because they are modeled with a fairly high polygon count, removing them from the solution speeds up the process significantly.
Observe that there is only one light source present. Double click on it to open the **Light Parameters** dialog as shown in Figure 6.10.1.2. The simple light intensity is set to 100%. Close the dialog. At this point we are generating a fast radiosity solution to see how the light illuminates the interior.

- Set the display mode to **Shaded Render**.
- From the **Display** menu, select **Radiosity Options**...
- In the **Radiosity Options** dialog, shown in Figure 6.10.1.3, the parameters are set up to generate a low quality but fast radiosity solution in the **Pure Radiosity** mode. Note, that the **Terminate After** option is set to stop after 6000 iterations, instead of stopping after a certain accuracy is reached.

- Click **OK** and then select **Generate Radiosity Solution** and let the solution finish. The final preview rendering is shown in Figure 6.10.1.4. Note that the coarse radiosity mesh shows many imperfections, but is enough for telling that the light intensity was not set high enough to expose the scene sufficiently.
• Select **Exit Radiosity** from the **Display** menu.

• Open the **Light Parameters** dialog again and increase the simple intensity. The final image was created with a intensity of 300%. However you may experiment with different settings.

• Repeat the radiosity solution with the different intensity settings. The solution generated with 300% intensity is shown in Figure 6.10.1.5.

---

**Figure 6.10.1.4:** The final preview rendering.

**Figure 6.10.1.5:** The solution generated with 300% intensity.

---

**Refining the mesh density**

After finding the correct light intensity it is necessary to increase the density of the radiosity mesh to improve the quality of the shading. In the final pass, the radiosity solution will be generated with the **Render Direct Illumination** mode. This means, that the boundaries of shadows will be created through raytracing or shadow maps and will be crisp. However, the secondary illumination generated by the radiosity solution depends on the density of the radiosity mesh. It should be set to the coarsest level possible, which still yields acceptable shading quality. Again this may require a few fast radiosity solutions.

• Select **Radiosity Options** from the **Display** menu.
• In the **Radiosity Options** dialog, increase the **Patches** parameter to 90%.
Select **Wireframe** from the **Display** menu and then **Initialize Radiosity**. This will create the radiosity mesh without starting to generate the solution. In wireframe display it is easy to see how dense the mesh is. Repeat the process, choosing 75%. Figure 6.10.1.6 (a) shows the mesh with 90% for patch density and (b) with 75% patch density. In the final rendering 75% was used which proved to be sufficiently dense. Keep in mind, that creating an overly dense mesh may significantly affect the performance of the solution.

![Figure 6.10.1.6: The radiosity mesh rendered in wireframe with patch density set at (a) 90% and (b) 75%.

Creating the final solution

At this point, the illumination and mesh density have been determined and we can now proceed to generate the final radiosity solution.

- Turn on all the remaining layers.
- Select **Radiosity Options...** from the **Display** menu.
- In the **Radiosity Options** dialog, choose **Render Direct Illumination** from the **Mode** menu.
- Increase the **Minimum** and **Maximum** sampling parameters to 3 and 9 and choose **Better** from the **Sampling Quality** menu.
- Increase the number of iterations in the **Terminate** tab to 20000.
- Click **OK** and select the **Shaded Render** display mode.
- Select **Generate Radiosity Solution**.

When the **Render Direct Illumination** mode is selected, the preview renderings generated during the processing of the solution will not show any shadows, since those are created during the final rendering. In our example, the previews are generated every 30 seconds. The radiosity solution can be stopped before the 20000 iterations have been completed, based on whether the previews show an acceptable shading quality. After the radiosity solution is completed, select **RenderZone** from the **Display** menu. The rendered image is shown in Figure 6.10.1.1.
6.10.2 Simulating atmospheric light

This tutorial builds on the example in 6.10.1. It uses the same scene with a single distant light source. In the example in 6.10.1, the illumination of the space was generated exclusively by the sun light hitting the surfaces and the light reflecting off these surfaces. In a real environment direct sun light is not the only light entering through a window. Consider, for example, a north facing window. Although no direct sunlight enters, there is still a considerable amount of scattered light coming into the space. This light comes from the illuminated hemisphere and light that reflects off clouds, buildings, and the ground. It is also non directional, meaning that it is roughly equally strong in all directions. This light can be simulated in form-Z using the Atmospheric Light option in the Parameters tab of the Light Parameters dialog, when the Light is of type Distant. While it is very accurate, this option tends to increase the radiosity processing time significantly. In the following example, we shall be using a different, more efficient method for generating atmospheric light. We shall be using area lights, which are placed in the window openings as additional light sources.

Creating the area lights for a sunny day

Load the file “distant_atmospheric.fmz.” It contains the same scene as in 6.10.1, but it has additional light sources.

- In the Lights palette, the light group Window Panes contains a set of 8 area light sources. Turn the shining and visibility attribute of this light group on. As illustrated in Figure 6.10.2.1, the area lights cover the window openings and shine towards the room. Alternatively, you can create the area lights from scratch. The easiest way to do this is to pick the 8 faces of the window panes and turn them into surfaces with the 2D Surface tool, located in the Derivatives tool palette. One by one, each surface is then converted into an area light as described in section 6.6.5.

- Double click on the sun light to open the Light Parameters dialog. Decrease the intensity to 200%. This is necessary, since the added area lights will provide additional illumination. Also change the shadow type to Soft (Mapped) and change the shadow softness to 20% in the Shadows tab.

- Double click in the Window Panes light group to open the Light Group Attributes dialog. Turn on the Intensity overwrite option, select Scale By and enter a factor of 2.
Select the **Shaded Render** display mode and then **Generate Radiosity**. A rendered image of the fast solution is shown in Figure 6.10.2.2. As you can see, the diffuse lighting of the area lights adds additional soft shading to the interior surfaces. You may want to experiment with slightly different intensity settings to create different lighting effects.

Now the same mesh refinements as in 6.10.1 can be applied and a final solution can be generated. A rendering based on a high quality radiosity solution with the same settings as in 6.10.1 is shown in Figure 6.10.2.3.

*Figure 6.10.2.2:* A rendered image of the fast radiosity solution.

*Figure 6.10.2.3:* A rendering based on a high quality radiosity solution.

**Changing the light intensities for a cloudy day**

The atmospheric lighting conditions set up above were meant to simulate a sunny day. On a cloudy day, the intensity of the sun light would be much less, whereas the diffuse light from the atmosphere would be significantly stronger. In our example these conditions can be created by turning the sun light intensity to 100% and the intensity overwrite scale factor of the area lights to 4. Also change the shadow softness of the sun light to 50% to create even softer shadow boundaries. An image with these settings is shown in Figure 6.10.2.4.

*Figure 6.10.2.4:* A rendering simulating the atmospheric light of a cloudy day.
6.10.3 A radiosity solution using interior lighting

Unlike in the previous example, an interior scene can be set up with accurate light intensities without risking overexposing it. In the following example of an interior all the light fixtures are assumed to be standard light bulbs. The light intensities are therefore specified in watts. In addition, this example will show how light groups can be used to change the intensity of several light sources in one step. To start, load the sample file “interior.fmz”. It contains the scene, shown in Figure 6.10.3.1.

![An interior rendering using light intensities of standard light bulbs.](image)

**Figure 6.10.3.1:** An interior rendering using light intensities of standard light bulbs.

**Finding the correct light intensity**

As in the previous example, the first step is to finetune the intensity of all the lights so that the final image is properly exposed. In this case the sample file contains all the light sources present in the scene. The intensities of the lights are defined in accurate units and represent standard light bulbs. For example, the lights on the chandelier are defined as point lights with 25 watts energy output and 30% efficiency. There are 36 lights on the chandelier and 9 more cone lights representing recessed lighting. In order to speed up the process of determining the correct light intensities, the 36 lights of the chandelier are turned off and a single point light source with 36 times the intensity (900 Watts) is turned on instead. Since the chandelier’s lights are fairly close together, a single light source gives almost the same lighting characteristics, but yields much faster processing times. Also note that, as in the previous example, the interior decoration is initially not used, when working to determine the light intensities.
• Select the **Shaded Render** display mode.
• Select **Radiosity Options**... from the **Display** menu.

- In the **Radiosity Options** dialog, the parameters are set up to generate a low quality but fast radiosity solution. The **Pure Radiosity** mode is used for the fast illumination studies. Different from the previous example, the termination of the solution is now specified as an accuracy percentage. The dialog with these settings is shown in Figure 6.10.3.2.

- Close the dialog and select **Generate Radiosity Solution**.

After a few minutes, the solution is completed. The preview image is shown in Figure 6.10.3.3. At this point, you may want to change the intensities of the lights. For example, the spot lights on the walls may be too bright, but the chandelier in the center not bright enough. Instead of changing the intensity setting of each individual spot light through the **Light Parameters** dialog, you can use the light group named Spot Lights to achieve the same effect in one step.

---

**Figure 6.10.3.2:**
The **Radiosity Options** dialog:
(a) **Options** tab and
(b) **Termination** tab.

**Figure 6.10.3.3:** A preview image of the radiosity solution.
• Select Exit Radiosity to delete the current solution.

• Double click on the Spot Lights group item in the Lights palette. This opens the Light Group Attributes dialog as shown in Figure 6.10.3.4. Choose the Intensity overwrite option and select the Scale By button. Enter a scale factor, for example 0.8. This will dim all the lights in this group by 20%. Click OK to exit the dialog.

• To change the intensity of the chandelier, double click on the light named Chandelier Simple. This opens the Light Parameters dialog. Click on the Intensity tab (6.10.3.5). Increase the value for Watts.

• Repeat the radiosity solution.

The solution with the altered light intensities is shown in Figure 6.10.3.6.
Refining the mesh density

In order to improve the shading quality of the radiosity based rendering, it is necessary to increase the mesh density. As we will be using the **Render Direct Illumination** mode for the final radiosity solution, the mesh only needs to be dense enough to accommodate shading from the reflected light in the scene. In the previous example, we increased the **Patches** parameter in the **Radiosity Options** dialog to cause a finer subdivision of the faces in the scene. Another way to refine the mesh is to use the Mesh tool in the Meshes and Deform tool palette. The advantage of using this method is that we have precise control over the size and orientation of the mesh. In addition, only faces that require further subdivision can be meshed. This allows us to keep the **Patches** parameter in the **Radiosity Options** dialog at a low value. The disadvantage of this method is that we are altering the geometry of the scene permanently. It may be wise to make a backup copy of the project before applying the mesh tool.

- Select the Mesh tool and set the X, Y and Z mesh interval to 1'-0”.
- Pick the vertical wall faces in the front room and apply the meshing. You can control the direction of the meshing by selecting the appropriate segment when picking the face (see section 4.10.1 in the form-Z User’s Manual).
- Now change the mesh interval to 2'-0” and mesh the floor and ceiling faces.
- Shading artifacts tend to be most noticeable at wall corners. You may want to apply additional 6” meshing at the faces meeting at the ceiling as shown in Figure 6.10.3.7.
- Redo the radiosity solution. Note, that the shading quality has improved significantly as shown in Figure 6.10.3.8.

*Figure 6.10.3.7:* A rendering showing the radiosity mesh.  
*Figure 6.10.3.8:* A rendering showing the improved shading quality.
Refining the lights

In the beginning of this example we substituted a single point light for the 36 individual lights of the chandelier. We can now activate these lights and turn off the single light. To do this click in the visibility column of the light palette next to the Chandelier lights group and Chandelier Simple light. Double click on the Chandelier light group name to open the Light Group Attributes dialog. Select **Intensity, Scale By**, and enter 1.5 in the text field. This increases the intensity of all the chandelier lights by the same amount as for the Chandelier Simple light, without having to edit each light source independently. Now redo the radiosity solution.

As you can see in Figure 6.10.3.9, the shading is almost the same as in Figure 6.10.3.8, although the light is more spread out on the ceiling and floor since it is emitted from many smaller sources instead of a single source.

Creating the final solution

At this point, the illumination and mesh density have been determined and we can proceed to generate the final radiosity solution.

- Turn on all the remaining layers.
- Select **Radiosity Options...** from the **Display** menu.
- In the **Radiosity Options** dialog, choose **Render Direct Illumination** from the **Mode** menu.
  - You may want to increase the termination accuracy to a higher number, such as 85 or 90%.
  - Increase the **Minimum** and **Maximum** sampling parameters to 3 and 9 and choose **Better** from the **Sampling Quality** menu.
  - Click **OK** and select the **Shaded Render** display mode.
- Select **Generate Radiosity Solution**.

After the radiosity solution is completed, select **RenderZone** from the **Display** menu. The rendered image is shown in Figure 6.10.1.1. Even with a dense mesh it is sometimes difficult to achieve perfectly smooth shading. In this example we have defined surface styles for the walls, floor and ceiling which have a slight pattern. For example, the walls use the Mist color shader with two very similar colors. This creates a subtle cloudy pattern on the wall, which helps hide slight shading imperfections. In addition, wall surfaces in the real world are rarely perfectly flat. The Mist pattern helps simulate these imperfections that would be very difficult to model.
Appendix : Precaptured textures

**Organic**
- bubbles
- skin
- clover
- sky 1
- grass
- sky 2
- leather
- sponge

**Wood**
- bark
- pine
- beech
- sawdust
- mahogany
- maple
- walnut
Masonry patterns

- blocks
- cobbles
- English 1
- English 2
- Flemish
- rustic
- stretcher
- stretcher 2
- stone wall
- grey pavement
- red pavement

Paper

- chipboard
- crumpled
- marbled
- marbled 2
- mottled
- pebbled
- splatter
- wall paper
**Cloth**

- brushed
- denim 1
- denim 2
- dogtooth
- Hessian 1
- mesh 1
- mesh 2
- webbing

**Stone**

- conglomerate
- granite
- gravel
- limestone
- marble
- marble 2
- sandstone
- silt stone
Floor patterns

floor tile 1
floor tile 2
floor tile 3
floor tile 4
floor tile 5
floor tile 6
floor tile 7
floor board 1
floor board 2
grey tile
tile 1
tile 2
tile 3
tile 4
tile 5
Miscellaneous

basket  
cast  
galvanized  
hornblind  
knurl 1  
knurl 2  
neon  
pitted  
waffle  
wrought
7.0 Introduction

Animation is a sequence of images displayed one after another fast enough to give the impression of motion. Prior versions of form•Z offered walkthrough or fly-by type of animation, which simulates a viewer moving through or around a scene by animating the viewing parameters. form•Z can now animate objects and lights in addition to views (cameras). In this write-up, we shall be using the term entity to represent all the items that can be animated in form•Z, namely objects, lights, and cameras, whenever a text applies to all of them.

Motion is created by animating the position and orientation (rotation) of the entities. In addition, most parameters of an entity can be animated to change the appearance of the entity over time. Each parameter or piece of information about an entity that can be animated is called a track. Typically, an animated entity has a number of tracks. The change in value of a track during the animation is determined by one or more animation controllers associated with the track. The controllers determine how the track’s value changes over time. The controllers define the key values for a track at specific points in time and define the method for interpolating the time between the key values. The relationship between the animated entity, its tracks, and the tracks’ controllers are represented in form•Z in a hierarchical fashion. This hierarchy can be inspected in the Animation Score palette, shown in Figure 7.0.0.1.

As shown in the Figure, an animated scene may consist of any number of entities, which appear as folders with the usual right pointing triangle in front of them. Clicking on this triangle opens the folder and exposes the tracks of the entity. These are also folders that can be opened to show the controllers they contain (Figure 7.0.01(b)).
There are nine standard tracks that are used for most entities. These control the position (in X, Y and Z), rotation (in X, Y and Z), and scale (in X, Y and Z) of the entity. Additional tracks are available depending on the type of entity. For example, height may be a track of an animated object. Most parameters of objects, lights, and views that are user controllable can be animated. A corresponding track is available for each parameter that can be animated. For example, in an object of revolution, the revolution angle can be animated using the revolution track. The number of tracks available is dependent on the characteristics of the entity. Only parameters that can be animated have corresponding tracks (i.e. an object’s name is not animatable).

The default controller is the Bezier controller. All tracks are initially created with this controller. The Bezier controller uses a Bezier curve to determine the change in a track’s value between key values. This controller can be complemented or replaced with additional controllers such as the random or sine controller.

A frame of the animation that contains at least one track and one controller with a key value is called a keyframe or simply key. The program interpolates between the keyframes to derive in-between frames. All the frames together, when played at a proper speed, produce an animation or a movie.

Setting up and editing animations in form-Z is supported by a number of tools, dialogs, and palettes. In the Options menu, there are two menu commands: Animation..., and Animation Manager... that invoke dialogs when they are clicked. There are three more items in the Pal-ettes menu that open the Animation Time Line, Animation Score, and Animation Editor palettes. There is also a pair of tool rows next to the last row of the tool palette that contain 8 tools, namely Keyframe, Animate along Path, Animate Entities, Animation Group, Animate Deformation, Extract Animation, Reverse Animation, and Replace Animation.

The first step in building an animated scene is to create the entities to be animated (often in their expected starting position). The object axis is the base that is used for controlling the position, rotation, and scale of an animated object. That is, when the object is animated, the transformations applied to the object to create motion are applied relative to the object’s axes. The axis can be edited graphically using the Transform Object Axis tool (see section 4.3.7) and numerically using the Query Object tool (see section 4.22.1). For views, the animation is based on the eye point and/or the center of interest. For lights, the animation is based on the origin of the light. Direct, cone, and projector lights also can be controlled by their center of interest.

The next step is to tell form-Z that you want the entity to be animated (all entities are static until animated). The third step is to define the key values for the entity. The remainder of the process is to refine the key values and controller parameters until the entity is animated as desired. Note that it is possible to work on multiple entities simultaneously and the process often involves tuning the animation parameters for all entities as the scene evolves. The Animation Time Line palette can be used to preview the animation in either wireframe or interactive rendering modes during the construction process.
7.0.1 The animation process

There are four ways to convert a static entity into an animated entity. An animated entity is an entity that can change its position or shape over time. Once an entity is animated, it is listed in the Animation Score palette that lists all of the animated entities in the project.

The Keyframe tool

The first method is to use the Keyframe tool. When this tool is used on a static entity, it makes the entity animated and creates default tracks and default keys for the current state of the entity at the current time. A Bezier controller is created for each track with a single key representing the current state. At this point, as there is only a single keyframe, the entity will not actually change when the current time is changed in the Animation Time Line palette. The next step is to add additional keyframes to define the state of the entity at additional key points in time. Additional keyframes can be added using auto keyframing or manual keyframing methods.

With auto keyframing, form•Z automatically adds and updates keys when required as an entity is edited. That is, as changes are made using form•Z tools and dialogs, these changes are automatically reflected in the animation of the entity. Automatic keyframing is controlled by the Auto Keyframe icon in the Animation Time Line palette or through a key shortcut. Auto keyframing is on by default.

To add a key with the auto method, first set the current time to the desired time for the new key using the Animation Time Line palette. Next make the desired changes to the entity. As the changes are made, the necessary keys are added to the controllers to keep the animation on track with the changes. The process is repeated as many times as needed to produce the desired results. An example is shown in Figure 7.0.1.1.

When auto keyframing is disabled, manual keyframing can be used to add keys. To add a key with this method, first set the current time to the desired time for the key using the time line palette. Next, perform the necessary operations to the entities or edit their parameters to establish the desired state of the entity for the time. At this point the entity is said to be off track. That is, it is not in a state that is represented by the animation tracks. Once the desired state is achieved, click on the object using the Keyframe tool. This establishes a new keyframe at the desired time. If a keyframe already exists at the current time, its values are changed to the new values. The entity is no longer off track. The process is repeated as many times as needed to produce the desired results.

Figure 7.0.1.1: Auto key framing:
(a) A cube is animated using the Keyframe tool (current time = 0).
(b) Current time is set to 5 and the cube is moved to the desired location.
(c) Current time is set to 2.5 and the cube is again moved with the Move tool.
(d) The cube is moved to a new location for time of 2.5 seconds.
If an animated entity is off track and the current time is changed using the Animation Time Line palette, the **Entity Off Track** warning is issued as shown in Figure 7.0.1.3. This offers the opportunity to retain the state of the off track entities.

**Keep Entities Off Track**: Clicking on this button keeps the off track entities in their off track position and parameters. This allows for the current time to be changed without losing the changes to the entities. The entities can then be keyframed at the new time with the Keyframe tool.

**Keyframe Changed Entities**: Clicking on this button keyframes all of the off track entities at the current time. This is equivalent to selecting all of the off track objects with the Keyframe tool.

**Cancel**: If you click on this button leaves the current time unchanged.

**Do Not Keep Changes**: If you click on this button the changes made to the off track entities are discarded and the entity goes back to its position for the new time.

**Figure 7.0.1.2**: Manual key framing:
(a) A cube is animated with the Keyframe tool (current time = 0).
(b) Current time is set to 5 and the cube is moved to the desired location.
(c) The cube is clicked with the Keyframe tool to add a new key frame.
(d) Current time is set to 2.5 and the cube is moved again.
(e) The cube is clicked with the Keyframe tool to add a key frame.
(f) Cube is moved to a new location.
(g) The cube is clicked with the Keyframe tool to update the key frame.

**Figure 7.0.1.3**: **Entity Off Track** alert.
The Animate along Path tool

The second method is to use the Animate along Path tool. This tool makes an object animated such that its position and rotation are controlled by a path defined by a form·Z object. The tool creates default path tracks and default keyframes. A Bezier controller is created for each track with a keyframe at the start and a keyframe at the end of the path. The entity is animated from the start to the end of the path. The path object is not changed by the operation, hence if the path is edited, the animated entity follows the updated path. For a complete description of the Animate along Path tool, see section 7.2.2.

Figure 7.0.1.4: Animate along Path tool:
(a) The original cube and spline path.
(b) The cube is made animated by activating the Animate along Path tool and clicking on the cube and the spline path.
(c) The current time is changed to 5 seconds.
(d) The spline path is edited using the Edit Controls tool.
The Animate Entities tool

The third method is to use the Animate Entities tool, which uses a set of existing entities to create a single animated entity. To do this, prepick the objects you wish to animate, using the Pick tool, and then, with the Animate Entities tool active, click anywhere in the project window. Note that it only makes sense to pick objects that are copies of the same object. When dissimilar objects are picked, the first one will be the entity that will be animated. The others will only be used for their location and orientation.

Each of the selected entities represents a keyframe of the animation sequence. This tool provides similar functionality to the Animation From Keyframes menu command found in previous versions of form•Z. For a complete description of the Animate Entities tool, see section 7.2.3.

Figure 7.0.1.5: Using the Animate Entities tool:
(a) The original cube.
(b) The cube copied and positioned at desired locations.
(c) With the Pick tool, the cubes are preselected in the desired order and then, with the Animate Entities tool active, we click anywhere in the project window.

Animating through the Animation Score palette

The final method is to drag the name of an object, light, or view from the Objects, Lights, Views, or Surface Styles palette into the Animation Score palette. This is done by placing the mouse on the name in the desired palette, clicking on the name and holding the mouse button down to drag it to the desired position in the Animation Score palette. As with the first method, default tracks and controllers are created with single key values for the current state.
7.0.2 Animating parameters

In addition to animating an object's position, rotation, and scale, most entities' parameters can be animated as well. A corresponding track is available for each parameter that can be animated. By default, when an entity is made animated, tracks are created for the primary parameters of entities. These are the parameters that are commonly animated and are found in nature. There are a variety of ways to control which parameter tracks are created, as discussed in later sections.

Parameter tracks are also added by default when a keyframe is added (using auto keyframing or with the Keyframe tool) and a parameter is changed. That is, if a parameter has a different value at the time the key is inserted than the adjacent keys, the parameter track is added and the necessary keys are created. If an object is already animated, parameter tracks can also be added manually by selecting the **Add Tracks** button in the **Animation Manager** dialog or from the contextual menu of the Animation Scores palette. See section 7.1.4 for more details on this dialog. Recall that contextual menus are invoked by clicking on a palette while pressing **ctrl**, for a single-button mouse, or clicking with the right button when using a multi-button mouse.

An example of animating the height parameter of a cube is shown in Figure 7.0.2.1.

![Figure 7.0.2.1](image)

**Figure 7.0.2.1:**

Mapping the height of a cube with auto keyframing:

(a) The cube is made animated by using the Keyframe tool with current time = 0.
(b) Time is set to 5, cube is moved to a new location, and its height is changed using the Edit Controls tool.
(c) Current time is set to 4 seconds.
(d) The height is changed using the Edit Controls tool.
(e) Current time is set to 2.5 seconds.
Numeric control of animated information

The Query, Query Attributes, Query Parameters, View Parameters, and Light Parameters dialogs that contain the values for animatable information have a small bullet to their right. This bullet is also a menu that pops up when pressing the mouse cursor on it. The bullet graphic representation changes to reflect the status of the parameter, as follows:

- [ ] The parameter is animatable, but is currently not animated. If there is no graphic present, then the parameter is not animatable.

- [ ] The parameter is animated and its current value represents a key value at the current time.

- [ ] The parameter is animated and the current value does not represent a key value.

- [ ] The parameter is animated, but the track is currently locked so the parameter cannot be changed.

The bullet menu contains four items used to control the animation state of the parameter as follows:

Add Track: The track for this parameter is added and a key is added for the current value at the current time. This is only available if the parameter is not animated (the track does not exist).

Add Key: A key is added for the current value at the current time. This is only available if the parameter is animated and there is not already a key at the current time.

Remove Key: The current key associated with the value is deleted. This is only available if the parameter has a key at the current time.

Delete Track: The track corresponding to the parameter is removed and the parameter is no longer animated. This is only available if the parameter is animated and is not locked.

Figure 7.0.2.2: Animation parameter control in (a) the Query Object dialog and (b) the Extrusion/Convergence Edit dialog accessed using the Query Parameters tool on an extrusion.
7.0.3 Fine tuning, rendering, viewing, and exporting animations

The creation of a successful animation usually requires refinement of the animation parameters to achieve the desired results. Using the keyframing techniques described earlier allows for control of the critical values in the animation, however, it does not offer control over the transition between the key frames. This can be controlled using the Animation Editor palette. This palette offers complete control over the details of the track and controller parameters. It is discussed in section 7.2.1.

Rendering an animation

An animation is rendered by selecting **Generate Animation...** from the **Display** menu. Selecting this item invokes the **Animation Generation** dialog, where generation parameters can be set, including the type of rendering and the name of the file into which the animation will be stored. Clicking **Generate Animation...** starts the rendering process during which a progress bar dialog continuously displays the frame number that is currently rendered, the rendered image is displayed in the screen background and the rendered frames are entered in the Animation Time Line palette. After the process is completed the animation is saved in a file carrying the name entered in the **Animation Generation** dialog and the suffix “.fan”, which stands for “form•Z animation.” Note that this is a form•Z internal format that cannot be read by another application. To open the animation in another program it will need to be exported to another format.

Viewing an animation

An animation can be played by selecting the **Play Animation...** item in the **Display** menu, which first invokes the **Animation Playback** dialog for setting the playing parameters. When clicking on **OK**, the animation is run. The screen is cleared and only the animation window appears. If **Loop** was selected in the **Animation Playback** dialog, the animation plays continuously by looping back to its beginning, until it is interrupted by pressing **esc** or **command+period** (Macintosh) or **ctrl+period** (Windows).

An animation can also be viewed using the **View File** menu command. When the Open File dialog is invoked you can select the “.fan” file you wish to view.

Exporting an animation

To be able to take the animation to another application, it needs to be saved accordingly. From the **File** menu you select **Export Animation...**. This invokes an Open File dialog where the form•Z animation file to be exported is selected. A File Save dialog is then invoked for selection of the format and name of the file to be exported. Animations can be exported to formats including QuickTime or AVI (Windows) or as a set of independent image files (such as TIFF or Targa).

After a QuickTime (.mov) or AVI (.avi) file has been exported, it can also be played from within form•Z. This is done by selecting **View File...** from the **File** menu. When the Open File dialog is invoked you can select the animation file you wish to view.
7.0.4 The animation preferences

The animation features of form•Z can be turned on or off through an option in the Animation tab of the Preferences dialog, shown in Figure 7.0.4.1, invoked by clicking on the Preferences... item in the Edit menu.

**Enable Animation:** When this option is on, the animation features in form•Z are active and usable. They are disabled otherwise.

![Figure 7.0.4.1: The Animation tab of the Preferences dialog.](image-url)
7.1 The animation dialogs and palettes

There are two items in the **Options** menu that invoke dialogs relevant to the animation process. The **Animation...** item invokes the **Animation Options** dialog and the **Animation Manager...** item invokes the **Animation Manager** dialog. The latter can also be invoked from the Animation Score palette. The content of these dialogs are discussed in sections 7.1.1 and 7.1.4.

There are also three items in the **Palettes** menu that open palettes used to set up animations. The **Animation Time Line**, **Animation Score**, and **Animation Editor** items open palettes with the same names. The Animation Time Line palette is open by default when **form-Z** is launched. The other palettes are not. How they are used is discussed in sections 7.1.2, 7.1.3, and 7.1.6.
7.1.1 Animation options

General options that apply to all animation data for a project are found in the Animation Options dialog, shown in Figure 7.1.1.1. This dialog can be opened by clicking on the Animation... item of the Options menu. Its options are as follows:

**Temporal Settings:** This section controls the time and frame rate information for the animation.

- **Start Time:** This parameter indicates the animation’s starting time. The default is zero.
- **End Time:** This parameter indicates the animation’s end time. The default is 10 seconds.
- **Duration:** This parameter indicates the animation’s duration, from start to end.
- **Frames:** This field represents the total number of frames in the animation. This number equals Duration in seconds multiplied by Frames Per Second.

- **Frames Per Second:** This is a combined numeric field and pop up menu. Certain preset values can be selected from the pop up menu, or any value can be typed in the field. This value is the rate at which the frames are played when viewing an animation. The higher the frame rate, the smoother the animation appears when replayed. The default frame rate is 30, which is sufficient for the human eye to experience flicker free motion.

The above parameters are interrelated. That is, when one is changed, the others are also adjusted so that they all display the correct values.

- When **Start Time** or **End Time** are changed, **Durations** and **Frames** are adjusted.
- When **Duration** or **Frames Per Second** are changed, **End Time** and **Frames** are adjusted.
- When **Frames** is changed, **Duration** and **End Time** are adjusted.
It is important to note that, when animated entities exist in the project, changing the start time, end time or duration could leave these entities out of the time range of the animation. For example, an object that is animated starting at one second and ending at four seconds would be out of the animation range if the start time was changed to 5 seconds. When the **Animation Options** dialog is closed, and the start time, end time, or duration have been changed, a warning dialog is presented which offers the option for the existing animated entities to be adjusted to the new parameters so they are proportional to the previous settings. If the start and end times are changed but the duration remains the same, the time values for the animated entities are shifted. If the duration is changed, then the time values are scaled into the new duration.

**Time Display Format**: This pop up menu controls how time fields are displayed throughout the animation interfaces.

- **Seconds**: Time fields are displayed as seconds. This is the default option.
- **HH:MM:SS**: Time fields are displayed with 3 numbers separated by colons for hours, minutes, and seconds.
- **Frames**: Time fields are displayed as frames.

**Preview Playback**: This section controls how the animation playback occurs.

- **Loop Control**: This menu contains three looping options which control what happens when the end time is reached while playing an animation. This option can also be controlled using the corresponding options in the Animation Time Line palette.

  - **Loop**: The animation plays continually. When it reaches the end time, it continues on from the start time. This is the default option.
  - **Palindrome**: This choice also continually plays the animation, but upon reaching the end time, playback is reversed until the start time is reached again, at which point playback is reversed again and this cycle repeats.
  - **No Loop**: The animation is played to the end time and stops

- **Enable Frame Skipping**: When this option is on, the playback will stay “on time” independent of the underlying graphics hardware. If for whatever reason during playback the hardware takes too long to display a frame, this playback option will cause the playback to skip frames that technically should have already passed in time. Turning this option off makes sure that each frame is displayed and is useful for verifying the state of all frames of the animation. This option is on by default.

- **Range**: These fields affect the range of the playback time. These options can also be controlled using the corresponding graphic controls in the Animation Time Line palette.

  - **Start Time**: Time at which playback starts.
  - **End Time**: Time at which playback ends.
**Entity Preview**: These options determine how an animated entity (throughout its animated life span) is displayed in the modeling window. These options are useful for motion analysis in being able to visualize all at once the places the entity will be. Note that these options are check boxes and more than one can be on at the same time. With all of these options off, only the entity is drawn at its location for the current time. For examples of the different preview methods, see Figure 7.1.1.2.

**Show Keyframes**: When selected, the state of the entity at each of its keyframes is shown. This is on by default.

**Show At Interval n**: When selected, the state of the entity at each interval of n seconds is shown.

**Show n Frames Before Current At Interval m**: When selected, the state of the entity is shown n times at a specified time interval (m), before the current frame.

**Show n Frames After Current At Interval m**: When selected, the state of the entity is shown n times at a specified time interval (m), after the current frame.

**Show Trail**: When selected, the trail of the entity through time is shown. This is the path that the centroid of the entity follows during the course of the animation.

**Use Controller Cache**: With this option enabled, each value of each controller is computed once for each frame and stored in memory for fast playback. However, as the number of controllers grows, responsiveness in general may become slower due to high memory consumption on computers with limited availability of RAM. If this option is off, less memory is used by the animation system, since only the current frame is stored in memory. However, the controller values must be generated on the fly for each frame change, which increases the burden on the CPU.

**Use Object Cache**: With this option enabled, each object that is animated is computed and stored in memory for each frame of the animation. As the number of animated objects grows, the same memory performance issues described for the controller cache apply.

---

**Figure 7.1.1.2**: Previewing an animated entity using:
(a) **Show Keyframes**, (b) **Show At Interval n**, (c) **Show n Frames Before Current At Interval m**, (d) **Show n Frames After Current At Interval m**, and (e) **Show Trail**.
7.1.2 The Animation Time Line palette

The primary function of the Animation Time Line palette is to control the current animation time and to preview the animation sequence. The current animation time is the point in time that is reflected in the modeling window and is used in animation operations. Note that the Animation Time Line palette is only available in the modeling environment.

The Animation Options dialog (see previous section) can also be invoked from this palette by clicking in the palette while pressing options (Macintosh) or ctrl+shift (Windows).

The parts of the Animation Time Line palette are noted in Figure 7.1.2.1 and are as follows:

**Figure 7.1.2.1:** The Animation Time Line palette and its functions, for keyframe based time control.

The time line is a ruler (horizontal line with measurements) across the middle of the palette. It represents time, the positions of the keyframes in time, and the current time. By default the time line shows the entire length of the animation. The left edge is the start or zero time position. The right edge is the end of the animation and the length of the time line represents its duration. The ruler markers and labels on the top portion of the time line indicate the frame numbers. The lower marks and labels indicate time. The size of the marks and intervals depend on the duration of the animation and the size of the Animation Time Line palette, which can be resized.

This red symbol is the current time marquee that indicates where along the time line the present time is positioned. The position of the current time can be changed by clicking anywhere along the time line, which causes the current time marquee to move to the new location. The time marquee can also be dynamically changed by clicking and dragging it along the time line. When the current time is changed, the image in the active modeling window is refreshed to show the new view. Pressing the shift key when moving the current time will delay the redraw of the view, until the key is released. This is useful when complex scenes are animated. When the time line is zoomed (see below) and the complete time line does not fit and is not displayed in the palette, it will autoscroll when moving the current time marquee beyond its left or right edge.
This symbol is used to represent the **keyframes** and their position on the time line. These can be repositioned and can be copied, as follows:

- To select a keyframe, click on it. To select multiple keyframes, press **shift** while clicking.
- To **move** a keyframe, click on the diamond and drag it along the time line.
- To **copy** a keyframe, press **command** (Macintosh) or **ctrl** (Windows) while clicking on its diamond and dragging it to a new location. A copy of the keyframe is created at the new location.

These represent the **start** and **end** time for playing an animation on the time line. When the **Play** (▶) or **Play Reverse** (◀) buttons are selected, the animation is played between the start and end symbols. The start or end can be changed by clicking on the respective symbol and dragging it along the time line. Double clicking the start time flag sets the playback start time to the animation start time. Double clicking the end time flag sets the playback end time to the animation end time.

- Pressing **command** (Macintosh) or **ctrl** (Windows) while double clicking on the start time flag, sets the playback start time to the first keyframe.
- Pressing **command** (Macintosh) or **ctrl** (Windows) while double clicking on the end time flag, sets the playback end time to the last keyframe.

The eight icons in the lower left of the palette control the playback operations of the animation in the active window. They are similar to the controls found on a VCR or DVD player.

- **Start**: When this item is selected, the current time is moved to the start time.
- **Previous Keyframe**: When clicking on this button, the current time is moved back to the previous keyframe. If the current keyframe is the first frame, the current time jumps to the last keyframe of the animation.
- **Previous Frame**: When clicking on this button, the current time is moved back one frame. If the current frame is 0, the current frame jumps to the last frame of the animation. Holding the mouse button down keeps advancing frames.
- **Play Reverse**: When this item is selected, the animation is played in the active modeling window moving backward through time. The icon changes to the Stop icon (●). When this item is pressed again, the playback is stopped and the icon reverts to its original state.
- **Play**: Same as Play Reverse except that the animation is played forward.
- **Next Frame**: Same as Previous Frame except that the animation is advanced to the next frame.
- **Next Keyframe**: Same as Previous Keyframe except that the animation is advanced to the next keyframe.
- **End**: When this item is selected, the current time is moved to the end time.
**Loop Control:** This button toggles through the three available looping options used when playing an animation. Each click toggles to the next option and the icon changes to reflect the active option as follows:

- **Loop:** The animation continually plays. When it reaches the end time, it continues on from the start time. This is the default option.
- **Palindrome:** This choice also continually plays the animation, but upon reaching the end time, playback is reversed until the start time is reached again, at which point playback is reversed again and this cycle repeats.
- **No Loop:** The animation is played to the end time and then stops.

**Auto Keyframe.** This button turns on/off auto keyframing. If auto keyframing is on, then when an animated entity is edited, keyframes are automatically created for the tracks currently present on the entity. If off, the Keyframe tool or the Animation Editor palette can be used to add keyframes. Pressing `option` (Macintosh) or `ctrl+shift` (Windows) while clicking on this button opens the **Auto Keyframe Options** dialog, whose options are the same with those found in the **Keyframe Options** dialog, invoked from the Keyframe tool. They are discussed in section 7.2.1.

The box labeled **Time**, located next to the playback buttons, displays the current time numerically. The format is determined by the **Time Display Format** option set in the **Animation Options** dialog. The current time can be changed by entering the desired time in the text field.

The box labeled **Frame** displays the current frame based on the current time. As the current time moves, this number is updated to the frame number the current time marquee is on. The current frame can be changed by entering the desired time in the text field.

The four icons at the lower right portion of the Animation Time Line palette are buttons for zooming and panning the time line.

- **Zoom in:** Clicking on this icon zooms in the resolution of the time line by 50%. This enlarges the graphic detail of the time line ruler. The zoom is performed relative to the current time and the time marquee remains visible.
- **Zoom out:** Clicking on this icon zooms out by 200%, up to the limits of the duration of the animation. This has the reverse effect of the Zoom in operation.

The **scroll wheel** of a mouse can also be used to zoom the time line in and out.
Fit: Clicking on this icon fits the entire time line in the palette. Pressing `control` (Macintosh) or `ctrl + alt` (Windows) while clicking on this icon, will fit the time line to the playback range of the animation.

Pan: Clicking on this icon makes the pan tool active. When active, clicking and dragging in the time line scrolls the time line with the movement of the mouse. It can also be scrolled without selecting the icon by pressing `control` (Macintosh) or `ctrl + alt` (Windows) while clicking and dragging the time line or using the mouse scroll wheel while the mouse is positioned on the palette. This functionality is only available when the time line has been zoomed in. The *middle mouse button* can also be used to pan.

The Animation Time Line palette has the usual window controls:

- It can be **closed** by clicking on the close button at its upper left corner. When closed it can be invoked and opened by selecting the **Animation** item in the **Palettes** menu.
- It can be **hidden** by clicking on the hide button, which is second in its upper left corner.
- It can be **resized** by clicking and dragging the lower right corner. It can be resized to any width and one of two heights, which either includes or excludes the time line (see Figure 7.1.2.3).
7.1.3 The Animation Score palette

This palette provides an easy way to manage all the animated entities and how they are arranged hierarchically, along with all of their tracks and controllers. This palette is similar to the Objects, Lights, and Layers palettes and has similar functionality.

When you use any of the animation tools to animate entities and possibly group them, their names and their track names appear in the Animation Score palette. Another way to animate an entity is to drag its name from its original palette into the Animation Score palette. For example, dragging an object from the Objects palette into the Animation Score palette will make it “animated”, optionally adding default keyframes.

The Animation Score palette, shown in Figure 7.1.3.1, consists of seven columns, similar to those in other palettes.

- The first column (left to right) is the **selection** column. Clicking in it, in front of the name of an animation entity, selects/deselects the entity. It displays a check mark when the entity is selected and no mark when it is not selected.

- The second column is the **name** column. It displays the names of the graph entities, tracks, and controllers. These entries are folders and have a right pointing triangle in front of their names. Clicking on the triangle opens the folder and list its content.

- The third column is the **value** column. It displays the values of the tracks at the current frame.

- The fourth column is the **visibility** column. Clicking on the diamond marker makes the entity’s motion trails and keyframes visible in the model window, provided the same global settings in the Animation Options dialog are enabled. This is only applicable on a per entity basis, not for individual tracks, which means that changes made to any track are applied to all the tracks of the same entity.

- The fifth column is the **lock** column. It controls whether or not a track can be edited. If the lock symbol is displayed, the track is dimmed in the Animation Editor palette and cannot be keyframed.
• The sixth column is the **on/off track** column. Tracks with the bullet marker on are currently being applied to the entity and are having an effect on animating it. Tracks with this off are not being applied and hence have no effect on the animation.

• The seventh column is the **edit track** column. Tracks with the hammer icon shown are editable in the Animation Editor palette.

Clicking on the title bar of the Animation Score palette brings up the **Animation Manager** dialog, which can also be invoked from an item in the **Options** menu and is discussed in section 7.1.4.

Clicking on the name column in the blank space under the last item in the list creates an empty **animation group**. Animation groups can also be created using the Animation Group tool and are discussed in section 7.2.4. Dragging animated objects, whose names appear in the Animation Score palette, onto the name of the group, makes them members of the group.

Double clicking on an item listed in the Animation Score palette invokes a corresponding options dialog, if such a dialog exists. These dialogs are discussed in subsequent sections.
7.1.4 The Animation Manager dialog

This dialog, shown in Figure 7.1.4.1, provides management tools for working with animated entities. It can be invoked either by clicking on the title of the Name column of the Animation Score palette or from the Animation Manager... item in the Options menu. This dialog is similar to other palette supporting dialogs, and allows you to execute many of the same operations that can be executed directly through the Animation Score palette. It also offers some additional functionality.

Most of the operations available in the dialog are applied to the active listing of an animated entity, track, or controller. You make an entity, track, or controller active by clicking on it. Depending on what type of an item is active, some of the operations in the Animation Manager dialog may not apply and will be dimmed. When active, they may act differently depending on whether an entity, track, or controller is currently highlighted.

**New Group**: Clicking on this button creates a new animation group with no animation tracks.

**Add Tracks...**: This button is available only when an entity or group is highlighted. Clicking on it opens the Add Tracks dialog (Figure 7.1.4.2) that contains a list of all the available animation tracks for the currently active entity. Highlight one or more of the tracks from the list to add them to the active item.

**Add Controllers...**: This button is available only when a track is highlighted. Clicking on it opens the Add Controllers dialog (Figure 7.1.4.3) with a list of all the available controller functions for the currently active track. Highlight one or more of the controller functions from the list to add them with default values to the active track.

**Remove**: Clicking on this button will delete the active item. If the active item is an entity, it will lose its animation data. Note, this deletes the animation data only, not the original entity, which is still part of the model. If the active item is a track, that track and all its controllers will be removed from the entity. If the active item is a controller, that controller and any of its children will be removed.
**Options...**: Clicking on this button will open any additional options dialogs that may apply to the entity, track, or controller. These are the same dialogs that can be invoked by double clicking on an item listed in the Animation Score palette (see previous section) and are discussed in subsequent sections.

**Duplicate**: Clicking on this button creates a copy of the active entity, track, or controller. The copy is placed under the same parent. If a group is copied, all of its children are also copied.

**Top, Bottom**: Clicking on this button moves the active listing to the top or bottom of the list for the same grouping level.

**Sort**: Clicking on this button sorts the list of entities for the currently highlighted item and all its children groups. This does not sort tracks and controllers.

**Clear**: Clicking on this button deletes all animation data from all the entities in the project.

**Purge**: Clicking on this button deletes from the list all the empty entities, that is, entities that have no tracks and tracks that have no controllers.

**Default Group Name**: This string is used to create default names for animation groups. By default this string is “Animation Group.”

**When An Entity Changes Its Parent, Keep Current Location**: Since each parent has its own coordinate system, moving a child from one parent to another could affect the child's own coordinates. This setting affects animated entities' current position and orientation when an entity moves from one parent to another parent. If checked, the animated entity's apparent location stays the same, even though it moves to a new coordinate system, meaning that its coordinates are changed to match the same location. If this option is not checked, the animated entity's coordinates do not change, and therefore the entity tends to visibly shift as it moves to its new parent's coordinate system. However, its same coordinates are now just being interpreted relative to its new parent.

**When Adding An Entity To Animation Score**: These options affect entities that become animated as you move them into the Animation Score palette from another palette, or click in the Animation Score palette (when creating a new group).

- **Create Initial Tracks**: If checked, then tracks are initially created for the newly animated entity.
- **Default Tracks**: This adds only those tracks marked with a default status.
- **Transformation Tracks**: This adds only the standard transformation tracks.
- **User Selected Tracks**: This allows the user to select which tracks are added via a dialog.
The Add Tracks dialog

This dialog, shown in Figure 7.1.4.2, is used to add tracks to an animated entity. To do this, simply select the desired tracks from the list provided in the dialog and press OK. The selected tracks are added to the entity that was active in the Animation Manager dialog at the time the Add Tracks dialog was invoked. Depending on the type of the active entity to which the tracks are added, different types of tracks can apply. The Add Tracks dialog will only list the tracks that apply to the active entity.

Category: This pop up menu contains items that represent the categories of tracks that apply to the active entity. For example, if the active entity is an extruded object, the three items shown in Figure 7.1.4.2(b) are contained in the pop up menu. Different categories would be contained for animated lights, views, and morphed items. The tracks displayed in the Available Track Types list change according to the item selected from the Category pop up menu.

Both the content of the Category pop up menu and of the lists of tracks are different for different types of objects. Typically, the first two categories are the Transformations and the Object Attributes. The latter are attributes of a general character, such as Visibility, Surface Style, and Casts Shadows.

The third category depends on the type of object. That is, each controlled object type has its own set of tracks. For example, different types of objects would be a Sphere, Cube, Extrusion, Helix, etc. Each of these types has its own set of tracks.

For the animated views, the categories will be View Eye/COI, which contains tracks that pertain to the positions of the eye point and center of interest of views; View Attributes, which contains tracks for parameters such as visibility, hither, and yon; and a third category that depends on the view type that may be Perspective, Axonometric, etc. These contain tracks such as Focal Length and Zoom.

Figure 7.1.4.2: (a) The Add Tracks dialog and (b) the Category pop up menu for an extruded object.
Similarly, for the lights, the categories will be **Light Location/COI, Light Attributes**, and a light type category, which might be **Cone**, containing tracks such as **Inner Angle**.

For animated deformations, tracks that pertain to the current deformation are listed under the proper category. For example, a **Shear** deformation will have **Upper Scale** as one of its tracks.

For animated surface styles, categories relevant to the surface styles are contained in the pop up menu and there are categories for each type of **RenderZone Plus** shader.

**Available Track Types:** As already mentioned, this list contains the track types of the category currently displayed in the **Category** pop up menu. This is a list of all the tracks available to an entity, regardless of whether the entity already has a track of a particular type. Multiple track types can be selected by holding the **shift** key down, from the same or different categories, and they will be added to the entity at the same time.

**The Add Controllers dialog**

This dialog, shown in Figure 7.1.4.3, works in a fashion similar to the previously discussed **Add Tracks** dialog, except that it has no **Category** pop up menu. All the available controllers are listed under a single category. You select one or more controllers from its list and you then press **OK**. The selected controllers are added to the track that was active in the **Animation Manager** dialog at the time the **Add Controllers** dialog was invoked.

![Add Controllers dialog](image)

*Figure 7.1.4.3:*
The **Add Controllers** dialog.
7.1.5 Animation controller functions

The animation controller functions compute the values for every track of an animation. The default controller function for a track is the Bezier interpolation function. Other controller functions may be used in addition to or instead of the Bezier interpolation function. Animation controllers control how a track’s value changes over time, thus for each time value or frame of an animation there is a single value computed. Some controllers are based on a mathematical formula to compute their values, while others are based on an interpolation method between fixed values. A controller function in general can be built from any algorithm that results in computing a value given some input parameters (and usually time is also a parameter).

To add a controller to a track, open up the Animation Manager dialog. Select the track in the animation hierarchy for which you would like to add a new controller. Clicking on the triangle next to the track name will reveal what controllers are currently controlling the track’s value. By default there should be a Bezier controller. Clicking Add Controllers... will bring up a dialog from which to choose additional controllers for the track. Controllers may also be removed from a track.

Stacking controllers

Some controllers can be “stacked” on top of other controllers, meaning that one of their input values is the result from another controller and is used by the controller to compute a new result. The Sine and Noise controller may be stacked in this way with existing controllers. The effect of this can be seen in the following examples.
The Bezier controller

This controller provides the user with a sequence of control points consisting of time value pairs. These control points are strung together through Bezier interpolation, which provides smooth transitions between each pair of key control points. Lead control points offer control over how fast or slow a change in value occurs, as well as controlling sharp or smooth transitions at keyframe points. The options associated with this Bezier controller function are built right into the Animation Editor palette. Through the Animation Editor graph, you can easily manipulate the Bezier control points of all the animation tracks. For example, you can change the control points types there, and change the control points values graphically or numerically. See section 7.1.6 for more information on the Animation Editor palette. Figure 7.1.5.1 is an example of an object with 6 tracks, each controlled by a Bezier controller function.

Figure 7.1.5.1: The Animation Editor palette.
The sine controller

This controller changes its value over time in the shape of a sine function. You may want to be able to define the motion of an object with a motion based on a sine function. This type of motion would be useful for animating pendulum motion of swinging objects, or any sort of repeating oscillating motion. You can control its parameters like amplitude, period, and time and value offsets. Figure 7.1.5.2 shows an example of a clock’s pendulum with a Sine controller for its Y rotation track.

The Sine controller can be stacked on another controller, or used by itself. Stacking it on to another motion will result in the original motion becoming wavy. For example, the original motion of an object rising and falling vertically can have a Sine filter added on top of it to oscillate its height while it is rising and falling. See Figure 7.1.5.3 for an example. Note that the sine controller is drawn on the graph in its pure form and also where it is having an effect on the Position Z track.
The Sine Controller Options dialog, shown in Figure 7.1.5.5, can be invoked by clicking on Options... in the Animation Manager dialog, while a sine controller is highlighted in the list. Its options are as follows:

**Start Time**: If this option is on, its value represents the time at which to begin the sine function.

**End Time**: If this option is on, its value represents the time at which to end the sine function.

**Amplitude**: This value represents the magnitude to which the sine function reaches in value.

**Period**: This is the time of one complete oscillation.

**Time Offset**: This value represents the amount of time to shift the sine function horizontally.

**Value Offset**: This is the value to shift the sine function vertically.

The noise controller

This controller changes its value over time by producing a value randomly offset from the original value.

You may want to add an element of randomness or bumpiness to the motion of something you are animating. You can change the strength of the randomness, or its frequency. You may want to add randomness to a camera to simulate shakiness or the ground shaking. You may want to add noise to a flickering light. An example of adding noise to a moving object is shown in Figure 7.1.5.6. To simulate a bumpy terrain, noise was added for its rotation tracks and position track.

![Figure 7.1.5.5: The Sine Controller Options dialog.](image)

![Figure 7.1.5.6: The Animation Editor palette.](image)

![Figure 7.1.5.7: An animation with noise added to simulate bumpy terrain.](image)
The **Noise Controller Options** dialog, shown in Figure 7.1.5.8, can be invoked by clicking on **Options...** in the **Animation Manager** dialog, while a noise controller is highlighted in the list. Its options are as follows:

**Start Time, End Time:** As for the **Sine Controller Options** dialog.

**Strength:** The value in this field represents the magnitude of the largest possible disturbance for the noise. This field is displayed using the units of the track which the controller is applied to. If the track represents a distance (i.e. position), a distance value is shown. If the track represents an angle (i.e. rotation), an angle is shown.

**Frequency:** The value in this field represents how often (in time) a new disturbance occurs.

**Controllers may also have tracks**

Controller functions themselves may have parameters associated with them that may change with respect to time. This means that controllers can also have tracks for these parameters. These tracks in turn may be controlled by their own controllers. This nesting may go on indefinitely, though for reasonable results there usually wouldn't be a need to nest more than a couple of levels deep.
7.1.6 The Animation Editor palette

This palette provides ways for graphically editing the functions that describe the motion or value changes of a track. A basic way for representing the value changes of a track is through a Bezier controller function, which contains a sequence of keyframes that describe how the value of the track changes over time. The way one keyframe transitions to the next is called interpolation. Through this palette, one can fully describe how the interpolation between Bezier keyframes occurs. In addition, other controller functions can be added to describe the value changes of tracks with respect to time. These additional controller functions are not necessarily based on keyframes and they may be dictated by mathematical formulas or specific data.

The Bezier controlled keyframes can be edited in the same way one would manipulate control points of a curve. Editing these control points allows the user to specify how transitions happen between and beyond keyframes. One can speed up or slow down the way an entity moves toward a specified location, change how fast something is spinning, mimic the pull of gravity, or repeat something indefinitely. All these effects can be done by editing the control points of a track. In short, the position, velocity, and acceleration are easily editable for any type of track through the Bezier controller function.
Following are a few common notions about motion control in relation to keyframes:

- If the line between keyframes is flat horizontally, the value for that track is not changing. In other words, it is not “moving” and therefore has no velocity (no change in value).
- If the line through two keyframes is slanted and straight, the value is changing at a linear rate, or constant velocity.
- If the line through two keyframes is curved, the rate of change is no longer linear and the velocity is no longer constant, but is changing between the keyframes, due to some acceleration or deceleration. An example of this would be the motion of a tree falling due to gravity, which can be simulated with a quadratic curve.

The Animation Editor palette is divided into the following sections:

- On center right is the graph, which is the most significant section. Here curves for controller functions of tracks are displayed and edited.
- Below the graph are tools and commands that are used for window navigation and control point manipulation.
- On top right, the section labeled Key displays information about the currently selected and highlighted keyframe of the active controller function, such as numeric coordinates and control point types.
- The section labeled Track, on the left side, contains options of the active track, including its Before and After behavior.

**Graph**

The curves with control points shown in the graph represent the currently editable or visible tracks. The main area of the graph is used for display and editing.

The values associated with a curve can be inferred by the horizontal and vertical axes of the graph. The horizontal time line axis is displayed across the top and shows the time or frame at which a value occurs. The vertical axis on the left displays values, including the actual values of the active track. Each track may have a different range of values depending on its track type. For example, a rotation track will show angular values for the value axis, whereas a path track may show either distance or percentage values.

Editable tracks are shown in their full color. Tracks that are visible but locked so that no editing can occur on them, are shown in a dimmed color. The user has control over what tracks are visible and editable in the Animation Editor palette or through the hierarchical track list.

Some commands or options apply to the active track or controller function. The currently active item is highlighted in the hierarchical list. To make a track active, with the Pick tool of the Animation Editor palette selected, click on the item's name in the hierarchical track list or on a control point of an editable track.
Some commands or options apply to a single keyframe, which needs to be active. You make a keyframe active by clicking on it with the Pick tool. The currently active keyframe is indicated on the graph by a circle drawn around the control point (●). The active keyframe’s settings are shown in the **Key** section.

The keyframes shown on the graph are assigned different types of control points. Each keyframe control point may have **lead in** and **lead out** control points which are affected by the control point types. The control point types affect a keyframe’s tangency, whether or not it has lead in or lead out control points, or whether it has stepped behavior.

**Editing control points and curves**

Different editing actions can occur using one of the 15 tools found under the graph area of the Animation Editor palette. These actions generally belong in five categories, namely, picking and moving, inserting and deleting, zooming in and out, panning, and aligning. They are discussed below:

**Pick:** With this tool active click on a control point or a curve to select them. When using a two button mouse you click with the left button.
- Clicking on an unpicked control point unpicks any previously picked control points and picks the clicked control point and any other control points that are dependent on it.
- Clicking on a key control point picks the key and its lead control points.
- Clicking on a lead control point picks the opposing lead control point, if it is tangentially locked, or the other lead control point of a step.
- Pressing **shift** while clicking allows one to pick multiple control points, adding each newly clicked point to the selection.
- Clicking and dragging over an area of the graph rubber-bands a box and picks all the control points within the box.
- Clicking on a curve picks the two immediately surrounding key control points.
- Pressing **control** (Macintosh) or **ctrl+alt** (Windows) while clicking on the curve picks all the control points of the entire track.
- Clicking on empty space unpicks any previously picked control points.
- Whenever a lead control point is very near or on top of a key control point, clicking once on the two control points will select the lead control point; clicking again will select the key control point. If there are several overlapping control points, holding **command** (Macintosh) or **ctrl** (Windows) while clicking will allow one to toggle through all the potential picks near the click point.

**Pick and Move:** With this tool active click on a control point or curve and drag the mouse where you wish to move it. If more than one points are highlighted (picked), they all move. One can move control points of any editable track.
- Once the dragging has begun, holding **shift** restricts movement either horizontally or vertically, depending on the distance to the nearest axis of the original click location.
- One can also turn on snapping for both value and time to move control points to rounded positions.
**Pick and Scale:** If a curve is selected (which is equivalent to all its points being selected) and you pick the end control point while you press `control` (Macintosh) or `ctrl+alt` (Windows), you will scale all the points of the curve relative to the direction in which you drag the mouse.

**Arrow keys Up, Down, Left, and Right:** Nudge motion in respective direction: The arrow keys can be used to incrementally move selected control points one pixel in the respective direction. If snapping is on, the selected control point moves to the next snap location.

Note that what is displayed on the graph are tracks as functions of values with respect to time. Hence, the curve cannot have two values for the exact same time and loops or curves that turn back in time are meaningless. Therefore, it is not allowed to create such conditions by moving a control point to a position corresponding to an earlier time than the previous control point.

**Insert Control Point:** With this tool active, clicking on a curve will insert a new keyframe on the curve at the click location. Clicking away from all editable curves inserts a new keyframe for the active curve at the clicked location. If snapping is on, the newly inserted control point will be snapped to the nearest snap location. Holding `control` (Macintosh) or `ctrl+alt` (Windows) and clicking away from all editable curves inserts a new keyframe on all editable curves at the clicked location. Upon inserting once, the insert mode changes automatically to the pick mode. If you desire to stay in insert mode, hold `command` (Macintosh) or `ctrl` (Windows) while clicking.

**Delete Control Point:** Clicking on this tool deletes all the highlighted control points from the track curve they belong to. The same result is achieved if, instead of clicking on this tool, you press the `delete` key on the keyboard.

**Cutting, copying, and pasting control points**

Control points can also be cut, copied, and pasted using the respective menu items in the **Edit** menu. To cut or copy, pick any number of control points on the active track and then click on **Cut** or **Copy** in the **Edit** menu.

Next, you may want to paste these points on the same or another track. Before pasting, make active the track where you wish the points to be pasted, then click on **Paste** in the **Edit** menu. Before executing the pasting, the program will invoke the **Paste Control Points Options** dialog, shown in Figure 7.1.6.2, where you set how the points are to be pasted.

Different track types have different types of values. Hence, copying from one track type to another does not copy the raw values of the data like in the case of copying two similar track types. Instead, the relative placement of the control points to each other is copied, and placed onto the new track with respect to the values displayed in the second track’s value axes. Also integer track types cannot receive values copied from non-integer tracks.
When the first point is selected at the time you are copying and it becomes the first point in the track to which you are copying, then the **Before** settings are also copied to the new track. Likewise, if the last point is selected and it becomes the last point in the track to which you are copying, the **After** settings are also copied to the new track.

The top of the **Paste Control Points Options** dialog (Figure 7.1.6.2) contains feedback about what is about to be pasted. It describes what will be pasted and where.

**Values**: This option affects the values being pasted.

**Paste Relative**: When this option is on, the values are pasted with respect to the entity’s default value for that track. So, the Position X track’s control points of one entity could be copied and pasted on to another entity without the second entity having the exact same X values, but rather have the motion be the same relative to the second.

**Time**: These options affect the time values being pasted.

**Remove Control Points**: The options in this group determine whether existing control points will be removed and how.

- **None**: When this option is on, none of the existing keyframes is removed.

- **Only Those Within Time Span**: When this option is on, the control points on the pasting track that are within the time span of the copied control points are removed, prior to pasting the new ones.

- **All**: When this option is on, all existing control points are removed before pasting the new ones.

**Time Span**: The two options in this group define the time span of the pasted control points.

- **Keep Same**: When this option is on, the start and end times of the newly pasted control points remain the same.

- **New**: When this option is on, new start and end times are entered in the two numeric fields next to it. This allows you to scale the pasted control points to a desired span, or to just alter at what point in time they are pasted, or both.
Zooming, fitting, and panning the graph

There are three methods for zooming, which are analogous to those available in the main project window. They work as follows:

- **Zoom in and out incrementally**: To zoom in, with this tool active, click in the area of the graph. Where you click becomes the center of the new graph image and the graph is redrawn after it is enlarged incrementally by the current zoom percentage. To zoom out, press `option` (Macintosh) or `ctrl+shift` (Windows) and observe the sign on the zoom icon to change from "+" to "-". Then click as before. The graph image is reduced incrementally, centered about the click point.

- Pressing `control` (Macintosh) or `ctrl+alt` (Windows) switches to a zoom by frame operation. That is, clicking and dragging the mouse rubber bands a rectangle, which specifies the area to be zoomed. As soon as the mouse is released, this area is enlarged to fit the screen.

- **Zoom in, Zoom out**: Clicking on these tools the graph image is zoomed in or out incrementally, without adjusting the center of the image. The operations are repeated everytime you click on the icons.
  - Pressing `command` (Macintosh) or `ctrl` (Windows) zooms in/out in the time axis only.
  - Pressing `control` (Macintosh) or `ctrl+alt` (Windows) zooms in/out in the value axis only.

- **Fit Visible**: Clicking on this icon will zoom automatically to fit all the visible tracks in the viewing area of the graph.

- **Fit Animation**: Clicking on this icon will zoom automatically to fit all the visible tracks within their value axes and the entire animation start and end times in the horizontal axis.

- **Fit Picked**: Clicking on this icon will zoom automatically to fit all the currently picked control points in the graph area.

Use the **Lock** check buttons in the **Graph Options** dialog, discussed later in this section to force zooming in one axis only.

- **Panning**: With this tool active, clicking and dragging in the graph area will move the graph in the direction of the mouse movement. Panning can also be temporarily enabled when in Pick mode. With the Pick tool active, press `control` (Macintosh) or `ctrl+alt` (Windows) while you click on the value axis or the time line and drag them vertically or horizontally, respectively.

- **Center Current Frame**: Clicking on this tool will automatically pan so that the current frame is centered in the graph.

Use the **Lock** check buttons in the **Graph Options** dialog to force panning in one axis only.

The mouse scroll wheel can also be used to zoom in/out. Also, the middle mouse button can be used to pan the graph.
Aligning control points

There are four methods for aligning control points and/or the lead in/out values:

- **Linear**: This command will cause consecutively picked control points to be aligned in a straight line between the outermost selected control points (without causing the control point types to be changed to linear).

- **Flat**: This command will cause picked keyframes’ leads to become horizontal with or equal to their key points’ values.

- **Align Value**: This command will align all the picked keyframes horizontally. The currently active keyframe is used as the reference to which to align. Holding `shift` while clicking on this command will also align the leads to the same value.

- **Align Time**: This command will align picked keyframes vertically, in time. Hence, this is useful for aligning keyframes of different tracks.

You can also align control points by entering a value numerically. To do this, pick all the points you desire to align and then type a numeric value in either the `Time` or `Value` field, or both of the `Key` section (see next subsection). All picked points will take the same numeric value.

**Setting the current time**

If you click in the time line, the current time is set to the clicked frame. Also, in pick mode, if you hold `option` (Macintosh) or `ctrl+shift` (Windows) and click anywhere in the graph, the current time is set to the time of the frame closest to the click point.
The Key section

This section displays information related to the currently selected (active) keyframe. The active control point is noted on the graph with an additional outlining circle. There are several parameters within this section that affect the active keyframe and its leads.

**Current key fields:** Two alphanumeric fields (with the word **of** between them) are used to denote the index of the **active keyframe** and the **total number of keyframes** in the track controller. The value in the first field changes every time the active keyframe changes. Or a new value can be typed in this field to change the active keyframe. The value in the second field changes automatically when new keyframes are inserted or deleted.

**Previous/Next keys:** Clicking on these buttons changes the selected key to the previous or next key of the active track. If more than one keyframe is picked at the time these buttons are clicked, the active keyframe cycles through only the picked keys.

**Time:** This editable field represents the time of the current key. It changes every time the current key's position along the time axis is changed. Alternatively, the value in the field may be changed by typing another value, which will cause the current key to also move along the time axis.

**Value:** This editable field represents the value of the current key and behaves as the **Time**, except that it applies to the value axis.

**Control point types:** At the right end of the **Keys** section, there are three frames labeled **Lead In**, **Key**, and **Lead Out**. Each contains an icon that, when highlighted, represents the type of the current keyframe control point. Making a new control point active, the type icons change. Each icon is also a pop up menu, the items of which can be used to change the type of the active control point. That is, while a keyframe control point is active, go to the pop up menu that contains the current type and from it select another type. The type of a control point affects how the curve flows through that control point. Different types produce different animation effects. For example, to simulate a bounce or rapid change in direction, tangency at the respective keyframe should be off. The available types are as follows:

**Lead In:**
- **Curve In:** Lead in point controls how the curve arrives at the key point.
- **Linear In:** There is no lead in control point, so the line arrives linearly at the key point.
- **Step In:** The lead in point controls where the step occurs.

- **Lock:** This makes the currently selected keyframe’s lead in immovable with respect to the key. Hence, you can still move the key, and the control point moves along with it, but you cannot move the lead in separately. If the keyframe is locked tangentially, this also forces the lead out to be movable only in the tangent direction.
Key:

- **Tangency Locked**: The curve is tangent at the key point, but the lead in and out may have different distances from the key.

- **Tangency Locked, Equal**: The curve is tangent at the key point and the lead in and out will always have equal distances from the key.

- **No Tangency**: The key point does not enforce tangency, allowing for sharp corners.

- **Lock**: This makes the currently selected keyframe’s key control point to be immovable.

Lead Out:

- **Curve Out**: Lead out point controls how the curve leaves from the key point.

- **Linear Out**: There is no lead out control point, so the line leaves linearly from the key point.

- **Step Out**: The lead out point controls where the step occurs.

- **Lock**: This does the same thing as for the Lead In option.

Note that the first and last control point of a track only has (at most) one lead control point. Therefore only certain icons are selectable when one of these key control points is the currently selected one. The meaning of the locked tangency status of these terminal points depends on the ante or post settings for the track or whether or not the track is closed. So, one can lock the tangency of the first control point with the ante behavior, the last control point with the post behavior, or lock the first and last together with the closed check box. For example, if the first keyframe is locked tangently and the ante behavior is set to constant, then the lead out control point will be forced to be horizontal with the key point, making it tangent with the constant ante behavior. In the Repeat Normal case, tangency acts like the closed case in that the first and last control points are tangent to each other.
**The Track section**

The track hierarchy displays the list of all animated entities, their tracks, and controller functions. This is the same as the controls of the Score palette without the "Value" column. The currently highlighted item is the active one, and since certain operations apply to a single item, this is the item to which these settings are applied. To change the currently active item, either use the hierarchy list and select another editable item, or click an editable track's control point.

Through the hierarchy list, you can control which tracks and controllers are visible and editable in the graph by turning on and off the lock and hammer icons respectively.

**Before/After:** The before and after behavior of the active controller function can be changed in this section. The before and after behavior of a curve affects how an entity gets its values from a track prior to the start time of the controller, and after the end time (formally called “extrapolation”). For example, if an entity has its first keyframe for a Position X track at frame 100, the before behavior will define what the X location is before frame 100 is reached. The difference in the options for the before and after behavior is that one is applied prior to the first keyframe and one is applied after the last keyframe.

**Type:** The items in this pop up menu define the before or after behavior of end keyframes.

- **None:** When this item is selected, no behavior is defined, so the entity should use its original value. This is the default state.

- **Constant:** When this item is selected, a line extends horizontally from the first or last key point, meaning the value is unchanging from the first or last value.

- **Tangent:** When this item is selected, a line extends tangently from the first or last key point, meaning the value continues on changing linearly.

- **Repeat Normal:** When this item is selected, the entire track is repeated.

- **Repeat Mirror:** When this item is selected, the entire track’s keyframes are repeated in reverse order.

**Has A Bound:** When this option is on, the behavior stops at some predetermined time, specified in the following fields.

- **Start Time/End Time:** The former appears when **Before** is selected and the latter when **After** is selected. When this option is on, the value entered in the **Start Time** field determines the absolute time at which the behavior stops. The value entered in the **End Time** field determines the time at which the behavior starts.

- **Repetitions:** When this option is on, the value entered in its field determines the relative number of times that the behavior should continue. The length of a repetition is the span of the keyframes.
**Offset By:** When this option is on, the value entered in its field determines the offset of the next repetition.

**Match Beginning To End:** When this option is on, the values of the repeated first and last keyframes are aligned. This option is only meaningful for the **Repeat Normal** behavior.

**Connect Beginning And End:** When this option is checked, the first and the last control points of the track are forced to have the same value. Also, if either terminal point is tangently locked, then the first and last control points also share tangency.

**Track Color:** This box shows the color in which a track is shown in the graph. Clicking on the box invokes the standard Color Picker dialog, where the desired color can be set. Track colors are retained in the preferences and apply to all tracks of the same type (i.e. changing the color of the Position X track in one entity changes it for all entities).

**Show Velocity Curve:** When this option is checked, this track’s velocity curve is displayed in the graph in a dimmed color. The velocity curve is the first derivative curve of the value curve. It displays how fast the value is changing as well as speed ups and slow downs and changes in directions.

**Options...:** Clicking on this button invokes an options dialog for the currently highlighted item in the hierarchy list, if such a dialog exists. For example, an animation group has options to change its axes, a path track has options for the path it is following, a sine controller has options to define its functional values. When one of these entities is highlighted, clicking on this button invokes its dialog.

**Constant Velocity:** Clicking on this button adjusts the velocity of the highlighted entity such that its resulting velocity is constant along its current motion trail. Sometimes after keyframing an entity's position, one would like to easily and automatically make the entity move at a constant rate along its same trail. This button recomputes the position tracks' keyframes along the same trail to achieve this result. Alternatively, one could use the Extract tool to extract the motion trail object, delete the position and rotation tracks off the entity, and use the Animate along Path tool on the entity and the extracted path object, which will result in the entity moving with constant velocity by default.
The Graph Options... button

This button, located at the lower right portion of the Animation Editor palette, invokes the Graph Options dialog, shown in Figure 7.1.6.3. It contains options for controlling the look and feel of the graph and for specifying how editing is done.

Time Line Axis: The options in this group determine how the horizontal time line axis looks and works.

Lock: When this option is on, zooming and panning is restricted and the time line can not be changed.

Show Lines: With this option on, which is the default, the animation bounds and current frame line are displayed. They are not otherwise.

Snap: The options in this group determine the snapping that is in effect relative to the horizontal time axis.

Axis: When this is on, snapping occurs relative to time axis values, set in the following two fields.

Seconds: The value typed in this field determines the snapping interval in seconds.

Frames: The value entered in this field determines the snapping interval in frames.

Other Points’ Times: When this option is on, snapping also occurs relative to other visible tracks’ control points.

Value Axis: The options in this group determine how the vertical value axis looks and works.

Lock: Works as for Time Line Axis (see above).

Show Lines: When on, extra horizontal lines are shown across the whole graph.

Snap: The options in this section set snapping for the vertical axis.

Axis: When on, snapping occurs relative to the vertical axis values.

Match Axis: When this option is on, the snap values are automatically derived from the current graph’s increment values.

Figure 7.1.6.3: The Graph Options dialog.
**Custom:** When this option is on, the user can type desirable values to which to snap. Different values are entered for the different types of tracks, as follows:

- **Distance:** The value in this field sets the snap value for positional track types.
- **Angle:** The value in this field sets the snap value for angular track types.
- **Percent:** The value in this field sets the snap value for percentage track types.
- **None:** The value in this field sets the snap value for track types without a unit.
- **Integer:** The value in this field sets the snap value for integer track types.

**Other Points’ Values:** When this option is on, the control points of other visible tracks can be used as snapping values.

**View Path Distance As Percent:** With this option on, which is the default, a path’s value axis will be displayed as a percentage. It will be displayed as a distance along the path, otherwise.

**Auto Scroll:** When this option is on, when moving control points or rubber banding to frame pick multiple control points, the graph will automatically pan following the location of the mouse.

**Keyframe Movement:** The options in this group determine the behavior of the modeling window when control points are moved.

- **Update Window While Dragging:** When this option is on, the model window is updated continuously while control points are being actively moved. This option should be turned off if this kind of interactivity is not desired or the animated scene becomes too complex to work interactively.

- **Active Window Only:** When this option is on, only the active window of the project is kept updated.

- **All Windows:** With this option on, all project windows are kept updated.

- **Auto Adjust Leads:** When this is checked, the magnitude of the lead control points is automatically adjusted as key control points are moved.

- **Only When Necessary:** When this is checked, the leads magnitudes are only adjusted when necessary. For example, since a lead out can not pass the succeeding keypoint in time, the lead out magnitude will be shortened when dragging the key toward it.

- **Allow Dragging Past Neighbors:** With this option checked, key control points can be moved past their neighboring key control points, effectively swapping their order. Otherwise the neighbor key control points act as boundaries.
7.2 The animation tools

There are eight tools that relate to animation and are discussed in this section. They are:

- Keyframe
- Animate along Path
- Animate Entities
- Animation Group
- Animate Deformation
- Extract Animation
- Reverse Animation
- Replace Animation

In addition to these tools, a number of common modeling tools can be used to manipulate animated entities. These are also discussed towards the end of this section.
7.2.1 The Keyframe tool

The Keyframe tool makes static entities animated and creates additional keyframes for animated entities when using the manual keyframing method. If the entity is static, it is made animated by adding tracks, adding it to the animation score, and creating an initial keyframe for the current state at the current time. If it is animated, a keyframe of the entity in its current state is created at the current time. If a keyframe already exists at the current time, the values are updated to reflect the current state of the entity.

With the postpick method, set the current time to the desired time for the keyframe using the Animation Time Line palette, select the Keyframe tool, click on the entity (object, object point, light, view or animation group) to keyframe. With the prepick method, use the Pick tool to select the entities to be keyframed, set the current time to the desired time for the keyframe using the Animation Time Line palette, select the Keyframe tool and click anywhere in the graphics window.

To add additional keyframes using the manual keyframing method, first disable auto keyframing by de-selecting its icon from the Animation Time Line palette. Set the current time to the desired time for the next keyframe using the Animation Time Line palette. Use the form•Z tools and interfaces to change the entities' appearance to the desired form. Use the Keyframe tool to capture the new keyframe. Repeat this process as many times as necessary to achieve the desired animation.

The Keyframe Options dialog, shown in Figure 7.2.1.1, can be invoked directly from the tool. It contains options that control how the Keyframe tool is applied.

Create Initial Tracks: When this option is on, tracks are created when a non-animated entity is made animated. When this is not selected, no tracks are created, however the entity is still added to the Animation Score palette. This item is on by default. The following options are enabled when this option is on:

- **Default Tracks**: When this option is selected, only the default tracks are created for non-animated entities.

- **Transformation Tracks**: When this option is on, only position, rotation, and scale tracks are initially created for non-animated entities.

- **User Selected Tracks**: When this option is on, the Add Tracks dialog is invoked that allows the user to pick which tracks will be initially added. This dialog is the same with the one invoked from the Animation Manager dialog (see section 7.1.4).
**Separate Tracks For Views:** When this option is selected, separate tracks are created for the eye point and center of interest for non-animated views. This allows for views to be keyframed by their eye point and center of interest positions instead of the standard position and orientation tracks. This option is used when the tracks are initially created and only applied when the entire view is selected. If only the eye point or the center of interest is selected, then the corresponding tracks are created.

**Separate Tracks For Lights:** This item functions the same as the Create Separate Tracks For Views, except for lights.

**Existing Tracks:** This group of options controls how existing tracks are updated when the tool is applied to an entity that is already animated.

**Key All Existing:** When this option is on, a key is added to the default Bezier controllers for all of the entities' tracks.

**Key Changed Tracks Only:** When this option is on, keys are only added to the default Bezier controllers when the parameters have changed from the current evaluated values. This prevents the accumulation of unnecessary keys. This option is on by default.

**Add Missing Tracks:** When this option is on, if a default track does not exist for the entity (because it has been deleted or was never previously added), the track is created and default key values are added. Otherwise only those tracks present are keyed. This option is on by default.

**Maintain Adjacent Keys:** When this option is on, which is the default, keys are inserted in the adjacent keyframes to maintain the current value of the entity at the adjacent keyframe's times.

**Leads:** This group of options control how the default lead-in and lead-out values are established for the default Bezier controller of each track. These options have subtle effects on the motion of the entity at keyframe positions.

**Magnitude:** The value in this field, which is a percentage, specifies the distance of the leads from their keys. This value is relative to the distance between adjacent keyframes and may be limited by the proximity of adjacent keys, as a lead cannot be located before the previous key or beyond the next key.

**Keep Flat:** When this option is on, the leads of an inserted key are flat. That is, they have the same value as the key and they are horizontal. This results in sharper transitions. When this option is off, the leads are positioned along the line between adjacent keys. This results in smoother transitions. This option is off by default.

Figure 7.2.1.2 shows examples with different values for the **Magnitude** and **Keep Flat** options. On the left are the graphs of the Animation Editor palette that correspond to the animated objects shown on the right. The animated object is a cube shown in top view. The cube was animated by clicking on it with the Keyframe tool at time 0. The time was then moved to 2 and 4 seconds in the Animation Time Line palette, and the cube was moved to the shown positions, using the Move tool. The **Show At Interval** option is turned on in the **Animation Options** dialog and 0.25 has been entered in its field. This displays the in-between frames every 0.25 seconds, which makes it easier to follow the effects of the options we use. Also, observe the positions of the white circles (leads) in the Animation Editor palette graphs.
Figure 7.2.1.2: Keyframe animation using **Magnitude** and **Keep Flat** set to:
(a) 15% and off; (b) 85% and off; (c) 85% and on.
**Animating points, segments, outlines, and faces**

In addition to animating objects and their parameters, entities at lower topological levels, namely points, segments, outlines, and faces of both facetted and smooth objects can be animated, using the Keyframe tool, which is sensitive to the active topological level. That is, when the tool is used in either postpick or prepick mode and the Topological Level is set to Point, Segment, Outline, or Face, the selected object part becomes animated. When an object’s part becomes animated, the object itself also becomes animated; hence the object is automatically added to the list in the Animation Score palette, if it is not already included. The animated part appears nested in the object's group, as shown in Figure 7.2.1.3.

An example of an animated point is shown in Figure 7.2.1.4. Segments, outlines, and faces can be animated in a similar fashion.

**Figure 7.2.1.3:** The Animation Score palette listing animated point, segment, and face of three separate objects.

**Figure 7.2.1.4:** Animating a point:
(a) The pyramid and its apex are animated by setting time to 0, topological level to Point and, with the Keyframe tool active, clicking on the apex point.
(b) After moving the time to \( n \) seconds, the apex point is moved up vertically.
(c) Playing the animation and stopping it halfway. Shown are the keyframes and the animated object. (d) As in c with the keyframes turned off.
7.2.2 The Animate along Path tool

The Animate along Path tool makes static entities animated such that they follow a path defined by an object. When the tool is applied, a path track is added to the entity. This track defines the position of the entity along the path. By default the entities travel the entire course of the path from its start to its end. Two keyframes are added initially to the default Bezier controller of the path track. The first key has a value of 0% and indicates that the entity is at the start of the path. The second key has a value of 100% and indicates that the entity has moved the full path’s distance. The key values can be changed using the animation editor.

The path object can be any object. However, the type of the object has an important effect on how it is used as a path. The most common types of paths are splines, nurbs curves, or any open or closed wire objects. In these cases the object moves along the length of the line path. When any other type of object is used, the centroid of the object is used as the path. Since the centroid is a single point, it does not define a change in position or orientation. This is useful for views or lights that have center of interest or eye paths, where it is useful to have them reference or “look at” an object. The path object can itself be animated so that the entity would be following a moving object.

The path object needs to be constructed before using the Animate along Path tool. The tool can be used in the prepick or postpick method, however the path object is always selected in the postpick method. With the postpick method, select the Animate along Path tool, then click on the entity to animate and on the path object. With the prepick method, use the Pick tool to select the entities to be animated, activate the Animate along Path tool and click on the path object. When using the prepick method all of the selected entities are animated along the path object.

Entities animated along a path object are “linked” to the path. The path object remains a normal part of the project and can be edited as it could before the operation. When the path is changed, the animated entities will follow the edited shape. If the path object is deleted, the linked entities become static. The path object can be made invisible without affecting the animated entities.

Views can have a path track for the entire view or separate paths for the center of interest and the eye point. The type of tracks that are created depends on what part of the view is picked. If the entire view is picked, then a single track is created. If only the center of interest or the eye point is picked, then the corresponding track is created. To set a path for the center of interest and also for the eye point, the tool should be used once with the center of interest selected and one more time with the eye point selected.

Direct, cone, and projector lights that have an orientation direction work as the views do. That is, they can have path tracks for the entire light or separate paths for the center of interest and origin.
The **Animate Along Path Options** dialog, shown in Figure 7.2.2.1, is invoked directly from the tool and contains options that control its operation.

When making an entity follow a path, there are a few options that can be set for how the entity follows the path. These options can be changed later in the Animation Editor palette, through the **Options...** button.

**Align To Path:** When selected, in addition to following the path, the entity is aligned or orientated to the path. Aligning the local axes of the animated entity to the path object controls the orientation of the entity. There are three implicit axes along the path object. The tangent axis is the tangent to the path along the direction being traveled. For example in a car traveling along a road, the car is the animated object, the road is the path and the direction that the car is traveling is its tangent direction (straight ahead). The secondary axis is the perpendicular direction to the path along the direction being traveled. In the car example, this is the direction to the left of the car (perpendicular to the road's direction). The third axis is the Up direction and perpendicular to the plane defined by the first two axes. In the car example, this is the direction up to the sky (perpendicular to the road). The three menus **Tangent**, **Secondary** and **Up** specify which axis of the entity is aligned to the corresponding axis of the path. Note that these menus are all linked together and changing one may change another. Figure 7.2.2.2 shows a sampling of the possible orientations.
Figure 7.2.2.2: (a) Original objects for Animate along Path operations. **Align To Path: Tangent, Secondary, and Up** set to:
(b) +x, +y, +z; (c) -x, -y, +z; (d) +y, +x, -z; (e) +z, +x, +y.
**Keep On Top Of Path**: When the entity follows the path, it will try to keep its up axis pointing upward, even when climbing inclines. This is normally what is expected when following a path.

**Follow Path’s Curvature**: The up axis is aligned to the path’s curvature, which may tip and turn. The path’s curvature vectors are defined for every point along a curve. Each curvature vector points inward in the direction of the inscribing circle’s center at each point along the curve.

**Create Bank Track**: If checked, this option will add a special track to the path track that will cause it to automatically rotate about its tangent axis when it approaches turns in the path. The more a path curves the more the object will bank by leaning in to or away from the curve. This could simulate an aircraft as it leans in to a curve, or an object being thrown away from the curve. The bank track, by default, has one keyframe with a value of 50%. One can increase the amount of banking power the bank track has to affect the entity by editing the curve of the track in the Animation Editor. Positive values will cause the entity to “lean in” to the curve while negative values will cause the entity to “lean away” from the curve. Banking values of 0% mean the entity does not lean at turns in the curve.

Examples are shown in Figures 7.2.2.3, 7.2.2.4, and 7.2.2.5.

---

**Figure 7.2.2.3**: Animation along Path with
(a) **Keep On Top Of Path** on and (b) **Follow Path's Curvature** on.
Figure 7.2.2.4: Animations along path with **Align To Path: Keep On Top Of Path** and **Create Bank Track** set to (a) on and off, (b) on and on at 50%, (c) on and on at -50%, and (d) off and off (**Unaligned** on).

Figure 7.2.2.5: Animations along path using:
(a) **Align To Path: Keep On Top Of Path** on and **Create Bank Track** off.  (b) **Unaligned** and **Create Standard Path Track** on,
(c) **Unaligned** and **Create Separate Path Tracks**: X and Y on.
**Unaligned**: When this option is on, the entity follows the path, without being oriented to the path. Its location simply moves along the path.

**Create Standard Path Track**: When this option is on, a single track is added, which controls the distance along the path.

**Create Separate Path Tracks**: When this option is on, independent $X$, $Y$, and $Z$ distance tracks may be added, by turning on the respective item under it. This allows you to have different curves for each track.

**Start Position**: This group of options specifies the time at which the entity is positioned at the start of the path.

- **Animation Start Time**: When this option is on, the entity is positioned at the start of the path at the start time set in the **Animations Options** dialog.
- **Current Time**: When this option is on, the entity is positioned at the start of the path at current time.
- **Custom**: When this option is on, the entity is positioned at the start of the path at the time specified in its field.

**End Position**: This group of options specifies the time at which the entity is positioned at the end of the path. They are analogous to the options of the **Start Position** (see above).

**Animating points, segments, outlines, and faces**

As for the Keyframe tool, the Animate along Path tool can also be used to animate parts of objects and is sensitive to the active topological level. That is, when the tool is used in either the postpick or prepick mode at the Point, Segment, Outline, or Face level, the selected object part becomes animated. An example of a face being animated along a path is shown in Figure 7.2.2.6.

![Figure 7.2.2.6: Animations along path using Unaligned](image)

(a) Original object (cube) and an arc path. (b) With the Animate along Path tool, selecting the top face and the path, makes the cube animated and displays its two keyframes. (c) Playing and stopping the animation at about 1/3 of total time. (d) Playing and stopping the animation again at about 2/3 of total time.
7.2.3 The Animate Entities tool

This tool creates a single animated entity from a series of objects, views, lights, or animation groups. Each static entity in the series is used to define a keyframe of the animated entity.

When creating an animated view, the original views must be of the same type. Other parameters of the view can be different (spin for example). The different values will be represented in the keyframes. When creating an animated light, the original lights must be of the same type, but other parameters may be different (as with views). When creating an animated object and the picked objects are of different types, the first object’s type will be used. Plain objects must also be geometrically and topologically similar, but may have different attributes and parameters.

Before using the tool, create the desired entities and edit their positions and parameters. It is recommended that the entities be created by making copies to be sure that they are all of the same type. The tool is then applied using the prepick selection method. With the Pick tool active, select at least two entities (object, lights, views, or animated groups), then activate the Animate Entities tool and click anywhere in the graphics window.

No dialog can be invoked directly from this tool, however a dialog is presented after the click with the Animate Entities tool. Depending on what type of entities have been selected, one of four variations of a dialog is invoked, namely: the **Animate Object Options** dialog, when objects are selected; the **Animate View Options** dialog, when views are selected; the **Animate Light Options** dialog, when lights are selected; and the **Animate Group Options** dialog, when animation groups are selected. Some of the options in these dialogs are common and some are unique to specific dialogs, as follows:
The **Animate Object Options** dialog

This dialog is shown in Figure 7.2.3.1. Examples are shown in Figure 7.2.3.2.

**Origin:** This group contains the remaining options of the dialog, all of which relate to the object to be animated, which occurs through its origin.

**Time:** This group of options determines when in the animation period the entity will be animated.

**First Keyframe At:** The value entered in this field determines the time of the first keyframe.

**Last Keyframe At:** This value determines the time of the last keyframe.

**Close Path:** When this option is selected, the path object will be a closed curve, otherwise an extra keyframe will be added such that the animated entity returns to the first keyframe.

**Create Standard Tracks:** When this option is on, the standard transformation tracks for the object are created, instead of a path track.

**Create Path Track:** This option will create a path track (instead of X, Y, Z tracks), to control the entity’s position. The path will pass through all the originally selected objects.

**Path Type:** Path tracks can be of different types.

- **Spline:** When this option is selected, a spline object is created for the path track.
- **Line:** When this option is selected, a vector line object is created for the path track.
- **Point:** When this option is selected, a point is used for the path. In this case, a path track is not actually needed, so it will not be added to the entity by default. If desired, it can be added with the following option.

**Construct Point Object:** Select this option to explicitly add a new point object to the project and use it for the path track.

**Keyframe Placement:** This item determines how to place keyframes in time along the path.

- **Keyframes With Constant Velocity Along Path:** When this item is selected, keyframes will be placed for the path track such that the new entity moves at a constant velocity over the entire path.

- **Keyframes At Equal Time Intervals:** Keyframes will be spaced equally in time between the start and end time.
The **Animate View Options** and **Animate Light Options** dialogs

The **Animate View Options** dialog is shown in Figure 7.2.3.3 The **Animate Light Options** dialog is not shown as it is identical to the former except that one of its sections is labeled **Location** rather than **Eye** (see below).

**Create Standard Tracks**: When this option is on, the standard transformation tracks for the view or light are created, treating the entity as a whole, rather than separating the eye or the location and the center of interest.

**Time, First Keyframe At, Last Keyframe At, Close Path**: As for the **Animated Object Options** dialog.

![Figure 7.2.3.3: The Animated View Options dialog.](image)
Create Separate Tracks: When this option is on, separate tracks are created for the eye or location and the center of interest.

Eye/Location or Center Of Interest: The Animated View Options and Animated Light Options dialogs contain two main sections whose content is identical, except for their titles. They are labeled Eye and Center Of Interest for the views and Location and Center Of Interest for the lights. The options in both are as follows:

Time, First Keyframe At, Last Keyframe At: As for the Animated Object Options dialog.

Close Path: If Create Path Track is selected, the path object will be a closed curve, otherwise an extra keyframe will be added such that the animated entity returns to the first keyframe. Examples are shown in Figure 7.2.3.4.

Create Position Tracks: When this option is on, separate X, Y, Z tracks will be created for the new entity’s position, instead, instead of path tracks.

Create Path Track: When this option is on, one path track will be created to control the entity’s position. The path will pass through all the originally selected eye/location or center of interest points.

Path Type: Path tracks can be of different types.

Spline: When this option is on, a spline object is created for the path track.

Line: When this option is on, a vector line object is created for the path track.

Point: When this option is selected, a point is used for the path. In this case, a path track is not actually needed, so it will not be added to the entity by default. If desired, it can be added with the following option.

Construct Point Object: Select this option to explicitly add a new point object to the project and use it for the path track.

Keyframe Placement: The options in this group are as for the Animated Object Options dialog.

For newly animated objects, rotation tracks will also be added such that the entity’s rotation is keyframed in alignment with the previously positioned copies of the object.

Figure 7.2.3.4: (a) Original lights for applying the Animate Entities tool. (b) Create Path Track on; Location: Spline with Close Path off; Center Of Interest: Spline with Close Path on.
7.2.4 Animation Group Tool

Animation Group

This tool is used to create an animation group. Animation groups are useful for animating a collection of entities as a single transforming entity. They are also useful for hierarchically linking coordinate systems to make complex motions with simple parent-child relationships. An animation group has its own axis (origin and orientation) as does an individual object. The axis defines the local reference system for the group including its origin and rotation. All the children entities of a group are transformed within the group's coordinate system.

This tool can be used to form groups that include existing static or animated entities. It can also be used to create empty groups. Once groups are created, they can be managed using the Animation Score palette and the Animation Manager dialog. The transformation tools can be used to change the position, rotation, and scale of the group.

This tool is typically used with the prepick method. Use the Pick tool to select the entities to be grouped, select the Animation Group tool, and click in the project window. The click location is significant when the Click Point option is on in the Animation Group Options dialog (see below). The equivalent to the postpick method can be used to create an empty group. That is, with the Animation Group tool active (and no entities prepicked), clicking on the project window creates an empty animation group. Such groups are useful as entities can be moved into them at a later time.

The Animation Group Options dialog, shown in Figure 7.2.4.1, is invoked directly from the tool and contains options that affect the application of the Animation Group tool.

Group Axes From: This group of options determines how the local axes for the group are established.

  * Click Point: With this option on, the new group’s origin is at the click point and the active reference plane establishes the orientation.
  * World Axes: When this option is on, the new group’s axes are the same as the world axes.
  * Active Reference Plane: When this option is selected, the new group’s axes will be the same as the active reference plane’s axes.
  * First Picked: When this option is selected, the new group’s axes are the same as the local axis of the entity picked first.

![Animation Group Options dialog](image)
**Average:** When this option is selected, the new group’s origin will be the average location of all the picked entities’ origins. The orientation is derived from the first picked entity.

**Center Of Bounding Volume:** When this item is selected, the new group’s origin will be the center of the bounding volume of all the picked entities. The orientation is derived from the first picked entity.

Note that the last three options do not work when using the postpick method to create an empty group. If one of these options is selected and no entities are selected, the click point option is used.

Examples are shown in Figures 7.2.4.2 and 7.2.4.3.

![Figure 7.2.4.2](image1)

**Figure 7.2.4.2:** A ball is animated to bounce up and down, then it is grouped with a truck. The animation group is automatically animated and a position track is generated for it, with two keyframes that cause the group to travel. The truck and the ball travel together while the ball also bounces on the truck's trunk.

![Figure 7.2.4.3](image2)

**Figure 7.2.4.3:** The moon (a) moves around the earth (b) after being animated along the orbit path (c). These three objects are grouped and the group is animated along path e, which is the orbit of the earth around the sun.
Creating and querying animation groups

Clicking on the name column in the blank space under the last item in the list creates an empty animation group. Dragging animated objects, whose names appear in the Animation Score palette, onto the name of the group, makes them members of the group.

Double clicking on an animation group brings up its Query Animation Group dialog, shown in Figure 7.2.4.4. Its options are as follows:

**Name:** Editable name of the animation group.

**Parent:** Name of the parent group of this group.

**Children:** Displays the number of each type of child in the animation group.

**Group Transformation:** This section is for editing the origin and orientation of an animation group’s axes. When you edit the group’s origin and rotation, note that all the children within the group will transform with the group. If this is not desirable, use the Edit Group Axes... button described below.

- **Origin:** The X, Y, Z location of the group’s origin within its parent’s coordinate system.

- **Rotation:** The X, Y, Z rotation values of the group’s axes relative to its parent’s coordinate system.

**Edit Group Axes...:** This button invokes the Animation Group Axes dialog, shown in Figure 7.2.4.5. This dialog is similar to the Object Axes dialog invoked from the Query dialog.
Within the **Animation Group Axes** dialog you can edit an animation group’s axes, but keep all its children in their same apparent locations. That is, the group’s axes change, but the children’s axes are updated with respect the group’s new location.

**Origin, Rotation**: Same as for the **Query Animation Group** dialog, except that children are adjusted with respect to the new location and orientation so that their animated locations are the same as before.

**Reset**: Clicking on this button resets the rotation values to 0 degrees.

**Reset To**: This pop up menu resets an animation group’s axes relative to its children in one of the ways represented by its items, as follows:

- **World Origin**: The group’s axes are set to match its parent’s origin.

- **Active Reference Plane Origin**: The group’s axes are set to match the active reference plane’s origin.

- **First Child’s Origin**: The group’s axes are set to match the origin of the child listed first in the group.

- **Average All Children’s Origins**: The group’s origin is set to the average of all the children’s origins. The orientation is unchanged.

- **Center Of Bounding Volume Of All Children**: The group’s origin is set to the center of the bounding volume of all the children. The orientation is unchanged.

![Figure 7.2.4.5:](image)
7.2.5 Animation deformation

Animation deformations reshape an object or group of objects over time based on a deformation method. Animation deformations share the same methods found in the Point Disturb and Deform Object modeling deformation tools and the deformation control object they create. It is highly recommended that sections 4.11.3 and 4.11.4 be read prior to this section. The notable difference between modeling deformations and animation deformations is that the deformation information is retained in an animation node rather than in the object's deformation list. This allows for the deformation parameters to be animated and for multiple objects to share the same deformations. Animation deformations are a process that is applied in the animation evaluation and do not make permanent changes to the original object. That is, the original object's parameters are retained but the object appears deformed. This enables the animation of the object's parameters and a deformation simultaneously. For example, an extruded object can simultaneously have its height animated and have an animated twist applied.

Animation deformations appear as groups in the Animation Score palette as with other animated entities (objects, lights, groups etc.). Objects that are placed in a deformation group are deformed by its deformation parameters. Objects can be added to a deformation group through the Animation Deformation tool or by dragging the object into the deformation group in the Animation Score palette. The parameters that control the deformations vary, depending on the deformation method. The parameters are represented by corresponding tracks for each parameter.

Most deformations are based on a reference plane referred to as the **base plane** and an **axis** which is usually the line perpendicular to the base plane, through the origin of the plane. Many deformations are affected by a **deformation box** which defines the portion of the object to be deformed. The dimensions of the deformation box also establish the base for some deformation parameters.

An animation deformation can be applied to the objects in the deformation group individually or collectively. When applied individually, the deformation parameters are applied to each object using the object's coordinate system defined by its axes. That is, the X and Y directions of the local axis define the deformation base plane and the Z axis is the deformation axis. The deformation box is the extents (bounding box) of the object relative to the base plane and axis. This type of group is referred to as a **deformation object group**.

![Figure 7.2.5.1:](image) The Animation Score palette with bulge object and bulge cage deformation groups.
When applied collectively, the base plane, axis, and box for the deformation are defined explicitly in the deformation group. This is referred to as the deformation cage. The cage is defined when a deformation group is created. The parameters of the cage can be edited and animated. The explicit definition of the deformation cage makes this method useful even when applied to a single object, as it gives much more control over the deformation and allows effects that can not be created with the implicit method. This type of group is referred to as a deformation cage group.

Once created, deformation groups can be placed inside of each other to create nested deformations. Deformations can be re-grouped in the standard fashion by dragging a deformation in or out of another deformation group in the animation score palette. This allows for an object to be simultaneously animated with multiple deformations. Figures 7.2.5.2 and 7.2.5.3 show examples of nested Taper and Bulge deformations.

**Figure 7.2.5.2:** Animated bulge deformation applied to two extruded objects using (a) a deformation object group and (b) a deformation cage group.

**Figure 7.2.5.3:** Simultaneous taper and bulge deformations applied to an object: (a) Stages of the animation with the Bulge Anchor Scale X track increasing in value and the Taper Lower Scale Y track decreasing in value. (b) The palette showing the Taper nested inside the Bulge.
**Animate Deformation**

This tool creates a deformation group in the animation score. The type of deformation and other options are controlled by the settings in the **Animate Deformation Options** dialog, shown in Figure 7.2.5.4.

This tool may be applied in either prepick or postpick mode. In postpick mode, with the tool active, click on an object. The deformation group is added to the animation score and the selected object is placed into the deformation group. You can also click anywhere in the modeling window that does not contain a selectable object. This will create the deformation group without any objects in it.

When using the prepick mode, all of the multiple objects can be added to the deformation group. The multiple desired objects are selected first using the Pick tool, then, with the Animation Deformation tool active, click in the modeling window. The deformation group is created and all of the prepicked objects are added to the group.

The deformation groups are created with default values for all of the tracks. The objects in the group do not appear deformed until the track values are changed using the track editor or using the **Query Deformation Group** dialog (see below). This dialog can be accessed by double clicking on a deformation group in the Animation Score palette.

**Model Type:** These options are the same as in the dialogs for the Point Disturb and Deform Object modeling tools.

**Method:** This menu lists all of the currently available deformation methods. The method selected from this menu is used for the creation of the new deformation group. These methods are the same as found in the dialogs for the Point Disturb and Deform Object modeling tools.

**Create Object Group:** When this option is selected, the deformation group is created as an object group. That is, the deformation is applied to the objects in the deformation group individually as described above.
**Create Cage Group**: When this option is selected, the deformation group is created as a cage group. That is, the deformation is applied collectively to all the objects in the deformation group using a common axis. If no objects are selected, the origin for the cage is the click point. The extents of the cage are determined by the Min and Max values in X, Y and Z, as specified in this group.

**Extents From Selected Objects**: When this option is enabled, the combined extents (bounding box) of all of the selected objects is used as the cage. When this option is off or there are no objects selected, the above Min and Max parameters are used.

**Origin From Selected Objects**: When this option is enabled, the origin of the deformation group is established from the average of the origins of the selected objects. When this option is off, or no objects are selected, the origin point is the click point.

**Base Reference Plane**: This option allows you to specify the reference plane that will be used as the base plane for the deformation. These options are the same as for the Point Disturb and Deform Object modeling tools and deformation edit dialog. Note that if the Object (XY), Object (YX) or Object (XZ) options are selected and no objects are picked, then the XY, YZ or ZH option is used instead.

**Add All Tracks**: When selected, all available tracks for the active deformation method are added to the new deformation group.

**User Selected Tracks**: When this option is selected, the Add Tracks dialog is presented for the user to select which tracks of the deformation should be added. The tracks listed in the dialog will vary depending on the deformation method being used.

*Figure 7.2.5.5:* Deformation Cage group examples: (a) original pyramids, (b) Extents From Objects on and Origin From Objects on, (c) Extents From Objects on and Click Point (d) Extents From Objects off and click point.
The Query Deformation Group dialog

This dialog provides information about a deformation group and access to the group's parameters. This dialog is accessed by double clicking on a deformation in the Animation Score palette, or by selecting the Options... item from the Score palette's contextual menu, or from the Options... button in the Animation Manager dialog.

Figure 7.2.5.6:
The Query Deformation Group dialog: (a) for a deformation object group, (b) for a deformation cage group, (c) the Group And Box tab, and (d) the Model Type tab.
**Deformation Group Information**: This section of the dialog contains information about the deformation group.

**Name**: The name of the deformation group. The name can not be changed. The name is the name of the deformation method followed by either **Object** or **Cage** to distinguish the two types of deformation groups.

**Parent**: This is the name of the group in which the group being queried resides. If it is not in a group, then this field will read **None**.

**Children**: This group contains statistical information about the contents of the deformation group including the number of **Groups**, **Objects**, **Deformations**, and **Tracks**.

**Parameters**: This tab displays the parameters of the deformation. The contents of this area varies depending on the type of the active deformation, as each deformation has its own specific parameters. Most of the parameters can be edited relative to the deformation box (percentage) or in their real world dimensional coordinates. These parameters have a slider with a percentage text field to the right, followed by a numeric text field. The slider and text fields are all linked to the same parameter. This allows the percentage slider to be used when the actual dimension is less important and experimentation is desired. The dimensional text field can be used when accuracy is desired.

**Group And Box**: This tab is only available for deformation cage groups. It contains items that define the group transformation and box for the cage deformation. The contents of this tab are the same for all deformation types.

**Group Transformation**: This sections defines the axis used to define the coordinate system or reference plane for the deformation. These values are initially derived from the **Base Reference Plane** option in the **Animate Deformation Options** dialog. The fields are consenting with the same options in the Group Transformation section of the **Query Animation Group** dialog. (See section 7.2.4).

**Box**: This group contains the parameters for the deformation cage. The deformation box is defined by the **Min** and **Max** values in **X**, **Y** and **Z** dimensions. These values are relative to the coordinate system (reference plane) defined by the above Group Transformation.

**Model Type**: The options in this tab determine the model type for the objects in the deformation group. These options are the same as the **Model Type** section of the **Animate Deformation Options**. Smooth models often give better results but are computationally slower than facetted models. It is often efficient to develop an animation using the **Facetted** option for performance and then switch to the **Smooth** option for use in a high quality rendering.

**Display Resolution**: This tab displays the standard display parameters for smooth objects. The values in this tab are only used for smooth objects.

**Base Reference Plane**: This menu is only available for object groups. It displays the reference plane used for the object deformation. This item works the same as in the **Deformation Options** dialog.

*Animation • The animation tools*
7.2.6 The Extract Animation tool

**Extract Animation**

This tool is used to create static entities from animated entities (objects, lights, or views) by extracting states of the animated entity. These new static entities are added to the project databases for objects, lights, and views. For example, the motion trail of an animated entity can be extracted and made into a curve object. Additionally, snap shots of where the animated entity is at various frames of the animation could all be extracted and constructed as separate static entities.

To extract with the prepick method, use the Pick tool to select the desired animated entities. Then with the Extract Animation tool active, click anywhere in the graphics window. To extract with the postpick method, activate the Extract Animation tool and click on an animated entity.

The **Extract Animation Options** dialog, shown in Figure 7.2.6.1, is invoked directly from the tool and contains its options.

**Extract At Interval:** When this option is checked, a static entity is extracted from the animated entity according to the given time interval.

**Start Time:** The value in this field determines the time at which to begin extracting from the animated entity.

**End Time:** The value in this field determines the time at which to stop extracting from the animated entity.

**Interval:** The value in this field specifies the time interval between each extracted entity. For example, if 1 is entered, a static entity will be extracted for every frame of the animation. If 10 is entered, a static entity will be extracted every 10 frames, etc.

**Extract At Keyframes:** When this option is on, a static entity is extracted from the animated entity for each time value that a keyframe occurs for that entity.

**Extract Trail:** When this option is on, a new static curve object is created that precisely matches the animated entity’s motion trail.

Examples are shown in Figures 7.2.6.2 and 7.2.6.3.
**Figure 7.2.6.2:** The Extract Animation tool. (a) The original animation. Using (b) Extract At Interval, (c) Extract At Keyframes, and (d) Extract Trail.

**Figure 7.2.6.3:** The Extract Animation tool. (a) The original animation. Using Extract At Interval and shown in (b) wire frame and (c) hidden line.
7.2.7 The Reverse Animation tool

Reverse Animation

This tool is used to reverse in time all the animation parameters associated with an animated entity. The starting and end positions are swapped in time so that the animation starts at the end position and ends at the original start position. All the keyframes between the start and end are swapped proportionally to their time position between the start and end. An example is shown in Figure 7.2.7.1.

With the postpick method, select the Reverse Animation tool and click on the animated entity to reverse it. With the prepick method, use the Pick tool to select the entities to be reversed, select the Reverse Animation tool and click anywhere in the graphics window. Note that, when reversing an animated group, the entities within the group are not reversed unless they are explicitly picked as well. This allows for the group parameters to be reversed without affecting the entities in the group.

Figure 7.2.7.1: Reversing an animation:
(a) Original animation and (b) after it has been reversed.
7.2.8 Replace Animation

The Replace Animation tool allows the user to replace the animation of one or more entities with the animation of another entity. This is a complementary tool to the Replace tool which replaces objects. The entity that will be replacing is referred to as source. The entity being replaced is referred to as destination. Either the destination or the source may be non-animated (static) entities. This operation is particularly useful when multiple entities need to have the same or similar motion applied to them. The animation can be developed on one entity and then easily applied to other entities. The operation can be applied to objects, lights, views (cameras) and animation groups. When the replacement is applied to different entity types, only the common tracks are replaced.

This tool may be applied in either prepick or postpick mode. In postpick mode, with the tool active, you first click on the source and then on the destination. The operation is executed immediately and the source replaces the destination animation information. When using the prepick mode, more than one entity can be replaced with one operation. The multiple destination entities are selected first using the Pick tool, then, with the Replace Animation tool active, you click on the source entity. The animation of all prepicked entities are replaced with the animation of the source entity.

Figure 7.2.8.1:
The Replace Animation tool: (a) An animated object and a static geodesic spheroid. (b) The static state of the spheroid is replaced with the animation from the cube by selecting the cube and then the spheroid.
The Replace Animation tool is affected by settings selected from the **Replace Animation Options** dialog that is invoked directly from the tool. It is shown in Figure 7.2.8.2.

**Maintain Destination**: When this option is selected, the destination entity remains in its present location. When the option is disabled, the destination object is moved to the source entity's location. Figure 7.2.8.3 illustrates the effect of this option. When this option is enabled and the source entity follows a path, the path object is copied. The options inside this group affect the operation when this option is enabled.

**Figure 7.2.8.2**: The **Replace Animation Options** dialog.

**Figure 7.2.8.3**: Replacing animation from an animated cube to a static pyramid: (a) Original objects. With **Maintain Destination** (b) on and (c) off.

**Position**: When this option is selected, the destination entity's position is maintained but not its rotation. This keeps the animation trail parallel between the source and the destination, however, the destination entity may be rotated to maintain the animation alignment. This option is on by default.

**Position And Rotation**: When this option is selected, the destination entity's position and its rotation are maintained. This keeps the destination entity intact, however, the trail of the animation of the destination object may not be parallel to the source.
The animation tools

Relative To: This group determines what frame to use to align the replaced tracks. When First Frame is selected, which is the default, the location at the first frame of the source animation is used. When Current Frame is selected, the location at the current frame of the animation is used. Recall that the current frame is set in the Animation Time Line palette.

Replace All Tracks: When selected all of the tracks from the source that can be applied are transferred from the source to the destination. When this is disabled, only transformation tracks (position, rotation, and scale) are replaced. This option is on by default.

Ignore Static Tracks: When this option is selected, only tracks that change over time are copied. That is, tracks with only one or no keys, or tracks where all of the keys have the same value are not transferred. This option is off by default.
Replacing between different types

When the replacement is applied to different entity types, only the common parameters are replaced. For example when replacing the animation of a light with the animation of an extrusion, only the position information is retained. The eye point and center of interest for cameras and lights can be used as either a source or destination, as shown in Figure 7.2.8.6.

Figure 7.2.8.6:
Replacing an animation from an animated cube to a light: (a) Original entities. The animation of the object applied to (b) the entire light and (c) the light's origin point only.
7.2.9 Using common modeling tools on animated entities

There are two methods for using common form\textsuperscript{Z} modeling tools to operate on animated entities and their parameters at specific keyframes. The first method is to operate on the animated entity itself, at some point in time during the animation process. The second is to operate on a graphic representation of a keyframe.

Operating on an animated entity

When stopping the animation at a certain time in the Animation Time Line palette, the animated object freezes at that position and can be edited and changed. For example, it can be moved, rotated, or operated on in some other way. When this happens and Auto Keyframe is on, the changes are automatically applied to the animation and can be observed as soon as the animation is run. If Auto Keyframe is off, the changes are not applied automatically but can be captured using the Keyframe tool, after a modeling tool has been used. If changes are made to non-animated information, the entire animated entity is updated to reflect the change. All modeling tools that make changes to existing objects can be used with this method.

Examples of operating on an animated object are shown in Figure 7.2.9.1. Given the animated cube (shown in a), it is stopped at the middle frame and this is where it will be operated on. The first operation (shown in b) adjusts its height. When the animation is run again, the cube's height will shrink between the 2nd and 3rd keyframes and it will grow again to full height between the 3rd and 4th keyframes. In other words, this operation has a local effect. In contrast, the next two operations (shown in c and d) have a global effect.

**Figure 7.2.9.1:**
Using modeling tools on an animated extrusion with the current time set to the middle keyframe:
(a) Original animated object.
(b) Using the Edit Controls tool the height is changed, which affects this keyframe only.
(c) The Height track is removed and the height is edited again with the Edit Controls tool, which changes the height throughout the animation.
(d) A new edge is inserted with the Insert Segment tool and moved with the Move tool, which again has an effect throughout the animation.
Operating on keyframes

The second technique is to apply a modeling tool directly to the graphic representation of a keyframe. However, only certain modeling tools can have an effect, when applied to keyframes. These are discussed below.

This technique is only available when the graphic representations of keyframes are visible in the modeling window, which occurs when the **Show Keyframes** option is on in the **Animation Options** dialog and visibility is turned on in the 4th column of the Animation Score palette for the respective entity. By default these options are on. The following tools can be used on keyframes:

- **Pick**

  The Pick tool can be used to select a keyframe by clicking with the cursor positioned on the graphic representation of the keyframe in the modeling window.

  Picking a keyframe in the modeling window has the same effect as selecting all the key values at the time of the keyframe, in the Animation Editor palette. Clicking on an animation trail (the line between keyframes) selects the keyframes at both ends of the section of the trail where the mouse is positioned when the click occurs. Selected keyframes are indicated in the **Keyframe Highlight** color, which is controlled by the corresponding option in the **Project Colors** dialog, invoked from the **Options** menu. When two adjacent keyframes are selected and the animation trail is visible, the trail is also shown in the keyframe highlight color.

- **Move**, **Rotate**, **Independent Scale**, **Uniform Scale**, **Mirror**, **Transform**

  The geometric transformation tools Move, Rotate, Independent Scale, Uniform Scale, Mirror, and Transform all can be used to directly transform a keyframe. When the postpick method is used, with the desired transformation tool active, you click on the desired keyframe. With the prepick method, any number of keyframes can be picked using the Pick tool, followed by the selection of the Transformation tool with which you click anywhere in the project window. The transformation is applied to all the preselected keyframes. Examples are shown in Figure 7.2.9.2.

- **Delete**

  The Delete tool can be used to delete keyframes. When the postpick method is used, with the Delete tool active, click on the desired keyframe. When the prepick method is used, with the Pick tool, preselect any number of keyframes, then activate the Delete tool and click anywhere in the project window. All the preselected entities are deleted immediately. An example is shown in Figure 7.2.9.2.
Animation • The animation tools

The range of the Transform tool

Recall that the Transform tool can be used to apply different geometric transformations to an object. It can also be applied to keyframes but can have a different range of effects, depending on the status of two options: the Auto Keyframe button, which is set in the Animation Time Line palette, and the Transform Animation Tracks option, which is located in the Self Options tab of the Transform Options dialog.

By default, transformations on animated entities are applied to the entity in its current state for the current time. If the Auto Keyframe option is on, the transformation is applied to the keyframe at the current time. When Transform Animation Tracks is on, the transformation is applied to the entire animation sequence by transforming all of its key values. Examples are shown in Figure 7.2.9.3. Note that in order to preserve the transformation properly when a rotation is applied, the animated entity may be placed into a new group in the Animation Score palette. In this case the rotation is applied to the new group that contains the entity.
7.3 Animating color

Some animated entities can have color tracks to animate a color parameter. For example, you may want to change a scene’s mood by animating a light’s color, or animate a subtle color change like metal becoming red-hot, or illustrate differences in the brick color of a building.

There are two ways of animating a color. Since a color is actually made from three values, many options could apply for how two colors are transformed from one to another. form•Z offers a simple method and one that allows more user control.

The simple way is to add a single color track, and then specify a set of colors. form•Z will then automatically interpolate the three color components of each color from one color to the next. This is the easiest way and recommended for most purposes.

The way for more user control is to add three separate color tracks, one for each color component. The burden is then upon the user how the three components combine to make new colors. This way is only necessary if you wish to animate one color component differently than another component.
7.3.1 Animating surface styles

As with objects, lights, and views, the parameters of a surface style can be animated. Recall that a surface style may have a number of different representations. By default, in *form•Z RenderZone Plus*, there are two: the Simple and the RenderZone representation, each of which appears as a separate tab in the Surface Style Parameters dialog. Each of these tabs contains its own group of parameters. In addition to these default representations, plugins may install more representations, which are also grouped in separate tabs.

The default *form•Z* representations allow the user to animate most of their parameters, which are accessible at different levels. For example, the parameters of the Simple representation are shown directly in the tab for this representation. Other representations may require to open additional dialogs where different parameters may be accessed. The animatable parameters have the standard animation bullet menu displayed next to them, which works as discussed in section 7.0.2. The parameters of the RenderZone representation are further grouped by shader type and are usually accessible through shader options dialogs that need to be opened. Exceptions are shaders that have only a single color parameter, such as the Plain color shader. In these cases, the color box is shown directly in the RenderZone tab with a bullet menu next to it.

As with objects, lights, and views, you make a surface style animated by dragging its icon from the Surface Styles palette into the Animation Score palette. When this is done, all animatable parameters of the surface style are assigned a track. Unlike objects, lights, and views, a surface style does not have a graphic representation in the main project window. Thus the Keyframe tool can not be applied to surface styles.

A surface style is an attribute of objects and faces. This means that, if a surface style is animated, all the objects and faces that use that surface style will automatically show the animated effect.

The animatable parameters of a surface style can be further classified into three types and are discussed under these categories in the remainder of this section: color parameters, the Frame Time parameter of image map based shaders, and numeric parameters.
Animating surface style colors

The general method of animating colors is described in further detail in section 7.3.2. Within the context of surface styles, by animating color parameters one can, for example, make an object transition smoothly from one color to another. Such an effect can be achieved by the following steps:

• Assign a surface style with the **Plain Color** shader to an object.
• With the time line set to 0 in the Animation Time Line palette, double click on the surface style’s icon in the palette to invoke the **Surface Style Parameters** dialog. Click on the red bullet next to the **Plain Color** box and, from the menu that pops out, select **Add Track** (Figure 7.3.1.1). Click **OK** to exit the dialog.
• Move the time forward in the Animation Time Line palette.
• Enter the **Surface Style Parameters** dialog for the same surface style again. Click on the **Plain Color** box and choose a different color from the color picker dialog. Exit from the **Surface Style Parameters** dialog.

You can preview the animation (the object changing color) by setting time to 0 in the Animation Time Line palette and clicking on the **Play** button. You can also create an animation using the RenderZone display mode and see the object change color over.

---

*Figure 7.3.1.1:* Animating the color parameter of the **Plain Color** shader of a surface style.
Animating the frame time parameter of map based shaders

The **RenderZone Plus** representation of a surface style has three shaders, which may be based on texture maps. These are the **Color Map**, the **Transparency Map**, and the **Bump Map** shaders. In the options dialogs for each of these shaders is a parameter labeled **Frame Time**, as shown for the **Color Map Options** dialog in Figure 7.3.1.2. If the file selected for the map is a movie file, such as a QuickTime movie, the image used for the rendering is one of the frames of the movie. The **Frame Time** field allows the user to choose which frame of the movie to display. The time value is entered in Hours:Minutes:Seconds format in the text edit field.

One application of a movie based image map would be to animate a television screen (see tutorial example in section 7.5.7). The steps required are identical to the steps of animating a color. To play the movie in its original speed, the time value entered for a key frame should be the same as the time value of the key frame in the time line.

---

**Figure 7.3.1.2**: The **Color Map Options** dialog with a movie file loaded.
Animating numeric shader parameters

The remaining shader parameters represent different kinds of numerical values and can be animated, as described earlier. Given that there is a great number of ways in which shader parameters can be combined, one can create a huge number of animated effects. Some examples follow:

**An object becomes slowly invisible:**
Use the **Simple Transparency** shader and animate its **Transparency** parameter from 0 to 100% (Figure 7.3.1.3).

![Figure 7.3.1.3: Animating the Transparency parameter from 0 to 100%.](image)

**A light object blinks:**
Choose a bright **Plain** color and the **Matte** reflection shader. Animate the **Matte** shader’s **Glow** parameter from 0% to 100% and back, in a cyclical fashion.

**A relief emerges out of a surface:**
Choose a texture map with a relief as a **Bump Map** shader. Animate the **Amplitude** parameter from 0% to 100% (Figure 7.3.1.4).

![Figure 7.3.1.4: Animating the Amplitude parameter from 0 to 100%](image)

**A piece of metal rusts away:**
Choose the **Eroded** transparency shader. Animate the **Coverage** parameter from 0% to 100% (Figure 7.3.1.5).

![Figure 7.3.1.5: Animating the Coverage parameter from 0 to 100%](image)
Preview icons of animated surface styles

If a surface style has any animated parameters and the time line is changed, the preview icon in the Surface Styles palette shows a small yellow triangle (Figure 7.3.1.6). It indicates that the preview icon no longer represents the surface style parameters accurately. There are two ways to update the preview icon.

- First, the Surface Style Parameters dialog may be invoked by double clicking on the preview icon. This will render the preview in the dialog and, after it is closed, will show the new preview in the palette.

- Second, clicking inside the yellow triangle in the palette will also render a new preview. Note, that this may be instantaneous or may take a few seconds, depending on the complexity of the surface style.

Limitations

Certain parameters of a surface style can currently not be animated. They are:

- The parameters of the reflection map shaders for each of the standard reflection parameters (ambient, diffuse, specular, reflectivity, transmission and glow).

- The Noise parameters of shaders that use random noise.

- The Area Sampling parameter of shaders that offer area sampling.

- The environment related parameters of the Environment reflection shader.
7.3.2 Animated color options

The options of a color track can be accessed through the Animated Color Options dialog (Figure 7.3.2.1), which can be invoked by double clicking a color track in the Animation Score palette or by clicking the Options... button in the Animation Editor palette or in the Animation Manager dialog.

**Color Type:** There are two methods for defining color and one of them can be selected from this group of options, as follows:

- **RGB:** When selecting this option, Red, Green, and Blue will be used to define the color.
- **HSL:** With this option on, Hue, Saturation, and Lightness will be used to define the color.

**Tracks:** In this section you determine whether a color will be animated with a single or three tracks.

- **Use 3 Separate Children Tracks:** When this option is on, form•Z will use 3 children tracks to define this color. This means that you will have to combine 3 Bezier controllers to make the color. If the **Color Type** is RGB, separate Red, Green, and Blue tracks will define the color, in a range of 0 to 100%. If the **Color Type** is HSL, separate Hue, Saturation, and Lightness tracks will define the color, again in a range of 0 to 100%. Use the Animation Editor palette to animate the 3 color component tracks of colors.

- **Use Single Track:** When this option is on, form•Z will use a list of colors and interpolate between them. The colors list defines all the colors of the animated color. Use the **Add** button to add another color to the list. Use the **Delete** button to remove a color from the list. To change a color, double click on it, in the list. You may also drag colors in the list to change their order. When you use this method you have a single Bezier controller that controls how one color transitions to another. Use the Animation Editor palette to edit the Bezier curve and set how one color transitions to another.

When **Use Single Track** is on, form•Z will take into consideration the **Color Type** when transitions between the colors in the list. If RGB is selected, one color transitions to the next as if the 3 RGB values were being interpolated. This means that, if a pure red color were being changed to a pure green, the color between them would be a muddy color. If the **Color Type** were HSL, red would transition through orange, yellow and then become green.
7.3.3 Animated color example

To animate a light changing color from blue to green, add the light’s Color track to an existing light. Open the **Animation Editor** and fit the visible tracks. In the track list, highlight the Color track. Double click it to open its options. In the **Colors** list, double click the first color and change it to blue. Then add a green color. Select **OK**. Next, locate the only keyframe of the color track. Change its time to 0 if it is not already at 0. The value of the keyframe should be aligned with the blue color shown in the value axis. We want to animate to the green color shown in the value axis. In **Insert** mode, insert a new keyframe at frame 100 at approximately the vertical position of the green color. Now the color will change from blue to green between frame 0 and 100.
7.4 Generating and saving animations

After all the parameters for an animation have been set, the frames that compose the animation need to be rendered and stored in a form\textsuperscript{Z} animation file. The animation file can then be played back on the screen.
7.4.1 Generating animations

To render the frames of an animation, the Generate Animation... command is selected from the Display menu. It invokes the Animation Generation dialog, shown in Figure 7.4.1.1, which contains the parameters required for the rendering. After the desired options have been selected, the animation frames are rendered in sequence and added to the animation file. As each frame is rendered, the modeling window refreshes to show the image of the frame. The progress bar shows the number of frames that have been rendered and the Animation Time Line palette shows the time and frame number, as frames are rendered.

**Display Type**: This pop up menu determines which type of rendering is used for the animation. The options currently set in the dialog for the selected rendering type are used for each frame of the animation. Pressing Options... next to the menu invokes the Options dialog for the currently selected rendering type.

One of the rendering types available is Interactive Shaded, which usually is performed by video hardware. Consequently, form-Z’s ability to capture the image from the hardware depends on the card and its driver. For best results, make sure that none of the form-Z palettes or any other window obscures the form-Z window being rendered.

Despite advanced antialiasing techniques in Shaded Render and RenderZone images, an animation based on these display types may show artifacts which appear as flickers. These are usually most noticeable where a pattern on the screen is of high contrast and small scale or along edges which are almost, but not exactly, vertical or horizontal. For example, when moving along a sequence of closely placed columns, the vertical edges of the columns may flicker. The following three options are designed to reduce or eliminate these artifacts. They can be used individually, or in combination with each other. Note, that these options are available for all rendering types, not only Shaded Render and RenderZone.

![Animation Generation dialog](image)

*Figure 7.4.1.1:*
The Animation Generation dialog with Animation File selected from the Store Frames In menu.
Super Sampling: When this option is selected, the entire frame is super sampled. That is, for each pixel on the screen additional samples are rendered, which leads to improved antialiasing. How many samples per pixel are taken depends on the option selected from the menu. Low renders a grid of 2x2, Medium 3x3 and High 4x4 samples. Note that super sampling will lead to increased rendering times for pixel based rendering modes like Shaded Render and RenderZone. For example, if High is used, 16 additional samples are rendered per pixel. This may increase the rendering time up to 16 times.

Motion Blur: When this option is selected, a number of subframes are generated and averaged with the original frame. This captures motions that occur faster than the frame rate and blurs them to achieve a more natural look. The more subframes one specifies, the higher quality the blurring looks, but the longer the animation takes to generate.

Blur Filter: Areas of high contrast in animation tend to flicker when played back. To avoid the flicker, they can be further softened by blurring each frame. Although each individual image will lose some of its crispness, the blur is much less noticeable in an animation. How much blur is necessary to reduce flickering depends greatly on the animation. In some instances, blur may not be necessary at all, whereas other scenes may need a high degree of blur. In general, if an animation contains patterns with high contrast and small scale, blur will be necessary to remove flickering artifacts. How much blur is applied depends on the option selected from the menu. Note that unlike Super Sampling and Motion Blur, Blur Filter does not significantly increase rendering time.

Frames To Render: The options in this group specify which frames of the animation will be rendered at this time. All the frames will have to be rendered for the final animation, however, they do not need to be rendered all at once. Rendering only some strategically chosen frames allows us to preview the animation and make adjustments as necessary, before spending the time to render all the frames.

All Frames: When this option is on (default), all of the frames in the animation are rendered.

Frame Range: When this option is selected, a range of frames can be specified to be rendered. This is done with the following parameters.

Start, End: The values entered in these numeric fields specify the beginning and end frames to be rendered.

Current: Clicking on this button sets Start and End to the current frame, as marked in the Animation palette.

Every n Frames: The value in this field specifies an interval for skipping frames between the Start and End.

Frame List: When this option is selected, a variable list of frames can be specified. This list can include single frame numbers separated by commas or a range of frames indicated by a dash and also separated by commas. For example “10, 31, 35-45” would render frames 10, 31, and 35 through 45.
**Generate On Local Machine:** When this item is selected, the animation is generated in a designated file on the local machine.

**Store Frames In:** This pop up menu contains two items, which allows the user to determine whether all rendered frames are stored in an animation file or whether each frame is saved to a separate image. Depending on which item is selected, a different set of options are displayed below it in the dialog.

  **Animation File:** When this item is selected, all the frames are stored in an animation file designed for movie playback. The area below the menu contains information about the name and location of the current animation file as well as options about how to save the animation file. By default, the animation file takes the name of the project with a “.fan” extension at its end. The default location is in the same directory as the project file.

  **Format:** On Macintosh, this menu contains: form•Z Animation and QuickTime™ Movie. On Windows, this menu contains: form•Z Animation, AVI, and QuickTime™ Movie. Each generates the respective animation format.

  **Compression:** This menu determines what type of compression, if any, is applied to the animation file.

    - **Off:** When selected, no compression is applied.
    - **Deflate:** This item is only enabled when the **Format** is set to form•Z Animation. When selected, the **Deflate** algorithm is used to produce smaller .fan files.
    - **QuickTime:** This item is only enabled when **QuickTime** is selected from the **Format** menu. When selected, the QuickTime compression engine is used to compress the movie file. The **Options...** button to the right is used to access the **QuickTime Compression** options.
    - **AVI:** This item is only available on Windows and is only enabled when **AVI** is selected from the **Format** menu. When selected, the AVI compression engine is used to compress the AVI file. The **Options...** button to the right enables access to the **AVI Compression** options.

  **Keep Existing Frames:** When this option is selected, any frames that are already rendered and are in the animation file from a previous generation are kept in the file. Any new frames that are generated replace any previous renderings of the same frame that may exist in the file. When off, which is the default, all rendered frames are cleared at the start of generation. If some critical parameter of the animation is changed (such as the dimensions), then the older frames in the file are automatically cleared.

  **Do Not Re-Render:** When this option is on, frames that exist in a file are not re-rendered even if they are specified in the **Frames To Render** section. This keeps existing frames from being overwritten in the file. This option, off by default, is most useful when some frames of an animation have been rendered for previewing and you want to render the remaining frames. Use this option along with the **All Frames** option to complete the animation.
Figure 7.4.1.2: The Frame Info dialog.

New File...: Clicking on this button invokes a standard Save dialog where a location for the file, other than the default, can be selected and a new file name can be entered. After exiting the Save dialog, the name and location of the animation file are displayed.

Choose File...: This button is an alternative to New File... and allows you to open an existing file with a previously saved animation for saving the new or revised animation.

Frame Info: This button invokes the Frame Info dialog, shown in Figure 7.4.1.2. It displays the status of the frames already rendered and of those that remain.

Playback When Finished: When this option is on, the animation is automatically played back, after the generation is complete. When this option is off, the animation can be played back as discussed in the next section.

If a long animation is created with a large image resolution, it is quite possible that the file size of the animation file will exceed a limit of 2 gigabytes. In such a case, the Format chosen should be form•Z Animation, which is able to split the entire animation over several files. If, during the animation generation, the initial animation file is about to reach the 2 gigabyte limit, a new, “overflow” animation file is created with the same name and “-2” appended. This new file will then be used. If it also reaches this limit, a third file will be created, etc. Each such overflow file is independent from the others and can be played back by itself. Care should be taken when re-rendering new frames to an existing set of animation files.

If the Keep Existing Frames option is off, all overflow files derived from the original animation file are deleted together with the original file. If the Keep Existing Frames option is on, new frames are always appended to the last of the overflow files. If the rendered frames are not in sequence with the frames in the last overflow file, the animation won’t play back the frames in the proper order. For example, consider three animation files, one original and two overflow files, with the first two containing 10,000 and the third containing 5,000 frames. If the user would choose to re-render frames 1-1000 with the Keep Existing Frames on, these frames would not be overwritten in the first animation file, but would be appended to the last animation file. One scenario where the Keep Existing Frames option can be used with overlap files, is when an animation is cancelled by the user and needs to be resumed at the cancelled point. The animation can be continued by using the Frame Range option and setting the Start to the frame, where the animation was previously cancelled. To find the last rendered frame, the last overflow animation file should be played with the View File... command, where the frame counter can be seen in the playback window. Keep in mind, that the overflow feature is designed to allow for the final production of a large animation without exceeding the 2 gigabyte file size limit. It is not meant to support flexible frame re-rendering.
Separate Image Files: When this item is selected from the Store Frames In menu, each frame of the animation is saved as a separate image file. The options below the menu change and contain information about the name, format, and location of the image files.

Format: This menu contains all installed image formats, that support pixel images. The Options... button next to the menu invokes the respective Options dialog associated with the selected format.

Base Name: The name entered in this field determines the file name of the image files. A 5 digit counter is appended to the base name, which identifies which frame is saved in the file. For example, if Base Name is set to “my_animation”, the 100th frame saved to a TIFF file would create a file called “my_animation-00100.tif”. Note, that any files that already exist with the same name, potentially from a previous animation generation, are overwritten without warning.

Location: This field shows the directory where the image files will be saved.

Choose Location...: Clicking on this button invokes the standard Choose A Folder dialog, where the location for the image files can be changed.

Figure 7.4.1.3: Part of the Animation Generation dialog with Separate Image Files selected from the Store Frames In menu.
Generate On Network: When this option is selected, the animation is generated on the network. When the Generate Animation... button is clicked with this option on, the Animation Network Render dialog (Figure 7.4.1.4) is invoked. Its options are as follows:

Job Name: The name of the rendering job is entered in this field. This identifies the job in the job queue.

Server: This pop-up menu identifies the rendering server that the job will be sent to. The Settings... button invokes the Network Settings dialog, discussed in section 3.6.8 of the form•Z User's Manual.

Priority: This slider determines the priority assigned to the job. If the priority selected exceeds the user’s priority, the job priority will be capped to the user priority.

Keep Project On Server: When this option is selected, rendering content files (form•Z project files, textures, etc.) are kept on the server for future renderings. This option is only available when the Registered User option is selected for the current server, as set in the Server Settings dialog.

Project: This pop-up menu identifies the name of the project to use for the rendering job. The project name is used to identify all of the rendering content files for the project. This name will be used in the directory name on the server machine where the files are stored.

Add...: This button can be used to add a new project name to the list. The Add Project dialog is invoked for entering the name of the project.

Remove...: Clicking on this button removes the current project from the project list. This will also delete any associated jobs which use this project. This will remove the project and its jobs from the server as well.

File Retrieval: This group of options is used to determine how the completed renderings and animations are retrieved from the server.
Manual: When this option is selected, completed renderings and animations are retrieved manually by the user. Manual retrieval is done by selecting the Retrieve... button in the Network Status dialog.

Automatic: When this option is selected, completed renderings and animations are automatically retrieved when they are completed. Note that automatic retrieval only happens while the form•Z application is running. The remainder of the options in this group are the same as the corresponding options in the Image Autosave Options dialog as documented in Section 6.1.1 of the form•Z RenderZone Plus User’s Manual.

Display Image After Retrieval: When this option is selected, the completed rendering will automatically be displayed, once retrieved from the Server.

Leave File On The Server: When this option is selected the completed rendering or animation is left on the server after retrieval. This option allows for future manual re-retrieval.

Store Frames In: This pop-up menu determines how the rendered animation will be stored.

Animation File: When this item is selected, the animation is saved as a FAN file.

Separate Image Files: When this item is selected, each frame of the animation is saved as a separate image file. The format of the image files is set by the next menu, which appears when this item is selected.

Format: This menu contains all installed image formats that support pixel images. The Options... button next to it invokes the Options dialog associated with the selected format.
### 7.4.2 Exporting animations

An exportable movie format can be produced from the form•Z animation format (.fan) by executing the **Export Animation...** command in the File menu. This converts the form•Z animation into a common animation file or into a set of independent image files. The supported animation files are QuickTime Movie (available in Macintosh and Windows) or AVI file (Windows only). To be able to export into a QT Movie format, QuickTime v. 6.0 or later must be used on both the Macintosh and Windows. It is recommended that you use the latest version of QuickTime available. Please see www.QuickTime.com for the latest QuickTime versions. The image formats are all of the image based formats that could be used to save a static image (TIFF, Targa, etc.).

When selecting the **Export Animation...** command, a standard File Open dialog is presented from where the form•Z animation file (.fan) to be exported is selected. Once the file is identified, double click on it or click on the Open button. The standard File Save dialog appears for the naming of the exported file, for identifying the location where it will be stored, and for selecting the export file format. Once the desired options have been set, press Save to export the movie. A progress bar shows the progress as each frame of the animation is converted and exported.

Available export options are QuickTime or AVI, plus the image formats. When exporting to QuickTime, clicking on the Save button invokes the standard QuickTime compression dialog, shown in Figure 7.4.2.1. Given that QuickTime supports a variety of compression types that can be expanded with third party plug-ins, the appearance of this dialog may vary from computer to computer.

When exporting an animation to an image format, the frames of the animation are saved to individual files. The name of each file contains the frame number following the name entered in the Save dialog. The **Animation Frame Export Options** dialog (Figure 7.4.2.2) is automatically invoked after the Save dialog when exporting to an image file. This dialog allows for the selection of the frames from the animation which should be exported. The options in this dialog function as the same options in the **Animation Generation** dialog (see earlier in this section); except that they control which frames should be exported rather than rendered. If a frame is specified that does not exist in the animation file (.fan), it is skipped.
7.5 Tutorials

This section contains seven tutorials that attempt to cover the different variations of animation with relatively simple examples. The best way to read the material in this chapter is in the order it is provided. That is, the theoretical sections (7.1 through 7.4) should be read first, followed by the tutorials in this section, in the order presented. While they all offer step-by-step instructions, the early tutorials are more detailed than the latter ones. That is, some theoretical explanations are given in the first couple tutorials, but are not repeated in subsequent tutorials.

While in all tutorials we expect the reader to be in a position to generate the required models, even though we do not give any modeling instructions, we also provide files with the initial models required by the tutorials. They are organized in folders carrying the same indices and names similar to the titles of the tutorial subsections. For each tutorial there is a file carrying the label "...Base," another one carrying the label "...Complete," a third that is a .fan file (form•Z animation file), and a fourth that is a QuickTime movie.

The Base file contains the initial model required to start the tutorial. The Complete file is for comparison, after you complete the tutorial, especially if you have a difficulty with some portion of the tutorial. The .fan file contains a rendered animation and is intended to show you roughly how yours will look after completion. A QuickTime movie is easily generated from the .fan file and can be played independently of form•Z.

The form•Z project files are all set up with an image size of 640 x 480 pixels. The .fan and QuickTime files were created at 320 x 240 to meet CD size constaints. The image size can be changed in the Image Options dialog, accessed from the Image Options item at the bottom of the Display menu.

Even though we provide you the "answers" and basically expect your animations to look more or less like ours, we would also like to encourage experimentation. Once you grasp the basics, you should feel free to experiment and explore beyond what we suggest. Actually, the more capable you are to deviate from what we give you the more knowledgeable you will have become of the form•Z animation techniques.


7.5.1 Swinging door tutorial

This tutorial will cover the creation and animation of a door swinging open. This is the first of a sequence of animation tutorials that are covered in this section and covers more details than the subsequent tutorials do. That is, some explanations about what is the meaning of some of the operations you will be asked to apply are included as well as some term definitions, rather than simply giving you instruction about what to do. This way this is more of an introductory session than the following tutorials are.

Simple animations, like the opening door that you will be doing shortly, can be accomplished easily using the Keyframe tool, while additional methods are also available and will be covered by subsequent tutorials. In all cases, you first need to set up the model you will be animating. For this tutorial, either open the sample file "Swinging Door Base", found on your program CD, or build a simple door yourself. If you do, change its name to "door_geometry" so that the illustrations we show agree with those you will be observing on your screen. The door in our file is as shown in Figure 7.5.1.1, which is admittedly a bit elaborate. If you build it yourself, it will probably be a lot simpler. In spite of this, all the animation values and settings you will be generating for this tutorial will look exactly the same regardless of which door you use. As a matter of fact, you can also do what is standard practice among animation professionals: set your animation with the simplest door possible and then transfer these settings to a more elaborate design of a door.

You will first prepare your door to rotate as you want it for this tutorial.

From the Rotate tool invoke the Rotate Options dialog and in it, under Axis Through, select Object Origin.

Select the Transform Object Axes tool and click on the door.

Notice that the door highlights in red and a transform widget appears at the center of the door where the axis currently is placed, as shown in Figure 7.5.1.1.

- Turn on Point Snap.
- Click at the center of the transform widget and move it down to the front, left corner of the door, as shown in Figure 7.5.1.2.
- Click on the project window (away from any selectable entity) to exit the Transform Object Axes tool edit mode.

Your door is now ready to start the animation process.
With the Keyframe tool active, click on the door.

A blue diamond is displayed at the left end of the Animation Time Line palette. This represents a key and indicates that the door is now animated.

- From the Palettes menu, open the Animation Score palette. Notice the "door_geometry" object is listed. Click on the arrow before the name to expose its contents. Observe that several attributes are listed, as shown in Figure 7.5.1.3. Recall that these attributes are called tracks and are generated by default when applying the Keyframe tool.

- In the Animation Time Line palette, move the current time forward to 2 seconds.
- Make sure the Auto Keyframe ( ) button is on.

With the Rotate tool rotate the door 90° clockwise about the Z (vertical) axis, as shown in Figure 7.5.1.4.

A clarification about the sign of the rotation is in order. form-Z gives you the option to specify whether clockwise or counter-clockwise is considered a positive angle. This is done in the Project Working Units dialog (Angle Options tab) invoked from the Options menu. By default, positive is counter-clockwise. Thus, assuming you did not change the default setting, the rotation you just did was negative 90 degrees.

Because Auto Keyframe is on, any changes made to an animated object are saved at the current time. In this case, another keyframe is automatically made to save the rotated new position of the door.

To test this, grab the current time marker in the Animation Time Line palette and drag it left and right. Notice the door swings open as the animation moves forward in time.

- From the Options menu, select Animation... to invoke the Animation Options dialog. In it, under Entity Preview, make sure Show Keyframes is off. Click OK to close the dialog.

- In the Animation Time Line palette, hit the Play button and observe the door open and then stop. It stops as soon as it reaches 2 seconds and stays open as time continues to move all the way to 10 seconds, at the right end of the time line. Then it cycles back to 0 seconds and starts again. At any time, hit the Stop button.

Note that, had you not turned off Show Keyframes, the image of the door would be shown dimmed at the positions of the two keyframes. Now that this option is off, the keyframes are not shown.

Animation • Tutorials
Animation Editor palette

Next, you will adjust the animation curve (the graphic representation of the track for Rotation Z) to make the motion of the door more realistic. Currently, the motion looks mechanical as it opens at a very constant speed. With a little adjustment you will make the door swing more naturally. You can also make the door appear to slam open or perhaps to appear as though it is blowing in the wind. You will do all these adjustments working in the Animation Editor palette.

- From the Palettes menu open the Animation Editor palette. The door object name and the tracks it contains should be listed in the upper left area of the palette and a graph should be displayed to the right. If the tracks are not listed, click on the triangle in front of the door name to expose them.

- Under the graph, there is a row of buttons. Click on the Fit Visible button ( ), which is the third from the left. The graph is adjusted to occupy the graphic area, as shown in Figure 7.5.1.5. The graph consists of a single straight line corresponding to the Rotation Z track. It is oriented diagonally, which represents the constant speed with which the door opens, as you observed earlier.

The remainder of the tracks (Position X, Y, and Z, Rotation X and Y) are currently graphed in the window as single dots on the left vertical axis, which means they only exist at time 0 and they have not been animated. You may actually make their graphs disappear by toggling off the little hammer symbols displayed next to them, in the fourth column of the list.

![Figure 7.5.1.5: The Animation Editor palette.](image)
Click on the **Rotation Z** track to highlight it.

The graph of **Rotation Z** in the Animation Editor palette shows all the keyframes. How the data is interpolated between keyframes is controlled by the **leads** attached to each keyframe. The leads are drawn as a line with a circle at their end and are coming out of both sides of a keyframe. The lead on the left of a keyframe is called the **lead in** while the lead on the right is called the **lead out**.

For the **Rotation Z** of the door, experiment with the positions of the leads to see how they affect the motion of the door. For example, grab the lead in of the last keyframe and drag it up so that it is some distance vertically above the keyframe. This will make the animation curve turn almost vertical as it is drawn into that second keyframe. Also drag the lead of the first keyframe to the 0 line, roughly as shown in Figure 7.5.1.6. To confirm the effects of these changes, close the Animation Editor palette and play the animation.

Now that you have played the animation, you must have noticed that, while a diagonal orientation of the graph represents even speeds, portions with more or less vertical orientations represent fast speeds and portions with horizontal orientations represent slow speeds. When reading the graph of a track, to figure out the behavior it represents, compare the distance traveled on the vertical axis relative to time, the distance traveled on the horizontal time. The longer the vertical distance, the highest the speed.

Next, you will make the lead into the second keyframe flat. This will gently ease the motion into that position.

1. Open the Animation Editor palette and highlight the second keyframe of **Rotation Z**.
2. Click the **Flat** button.

   Note that the lead in is positioned directly left of the keyframe. The lead handle can be stretched further left to make the lead in even flatter (Figure 7.5.1.7).

   In the Animation Time Line palette, click **Play**. The door starts to swing out and gently comes to a stop. Flat leads will give objects a smooth and fluid motion.
Adding keyframes

Next, you will make the door appear to be abruptly pushed open, slam into the wall, and rebound slightly. You will do this, by adjusting the value in the existing keyframe and by adding one more keyframe.

First, adjust the existing keyframe to make the door slam into the wall. Logically, the door would be at roughly 180 degrees when it strikes the wall. You will make the adjustment to the rotation angle directly in the Animation Editor palette.

  • Open the Animation Editor palette and click on the Zoom out ( ) button (second from left) as many times as necessary until the -180° mark is displayed on the vertical axis.
  • Now drag the second keyframe vertically to the -180° line. The time remains the same, which is 2 seconds. This will cause the door to open by 180° and slam against the wall.

You will next adjust the lead out of the first keyframe to imitate a sudden opening of the door. This can be done by making the lead out for the first keyframe linear.

  • Click on the first keyframe. Click on the Lead Out icon and, from the menu that pops up, select Linear. This is the middle icon that shows a line with no lead.

Notice that the lead has disappeared from the first keyframe. The lead out for the first keyframe is basically pointing at the end of the lead in for the second keyframe. Drag around the lead in for the second keyframe to observe the behavior of the graph.

To adjust the lead in for the second keyframe to be linear, use the same technique as above.

  • Highlight the second keyframe, click on the Lead In icon and select Linear (middle icon) from the pop up menu. Click on the Fit Visible button and observe the perfectly straight line connecting the two keyframes, as shown in Figure 7.5.1.8.

Play the animation. With the second keyframe set to a value of -180° and a time of 2 seconds, the door appears to open rather slowly. Thus you will make some more adjustments, next.
• In the Animation Editor palette click on the second keyframe.
• Change the **Time** field to 0.5 and observe how it changes the graph (see Figure 7.5.1.9).
• Play the animation. It appears as if the door swings open and then sticks to the wall.

Next, give the door a little rebound, so that it bounces off the wall. This requires the addition of one more keyframe.

• In the Animation Editor palette, activate the **Insert** button and click on the space after the second keyframe, to add one more keyframe.
• Highlight the second keyframe, click on the **Lead Out** icon, and select **Linear**. This makes a linear lead out from the second keyframe.
• Set the **Value** of the third keyframe to -165° and the **Time** to 0.7 seconds.
• Adjust the lead in of the third keyframe to be flat. To do this select the lead in and then click on the **Flat** button.

The graph should now appear as the one shown in Figure 7.5.1.10.

• Play the animation and observe the motion of the door.

The sharp point formed by the line going into and out of the second keyframe is what gives the motion that abrupt reversal in direction. Since the angle of the outgoing line from the second key is a little less steep, the speed of the door on the rebound will be a little slower. The speed decreases from there until coming to a soft stop. The gentle end is due to a flat lead in on the last keyframe.
A door swinging in the breeze

You will next make the door appear to swing in the breeze. While there are many ways this can be done, a simple method will be used in this tutorial.

- Select the first keyframe in the Animation Editor palette and change the **Value** field to -15.
- Make the lead out of the first keyframe curved. You do this by highlighting the first keyframe, then clicking on the **Lead Out** pop up menu, and selecting **Curved**, which is the top icon of the pop out menu. This causes a lead to appear.
- Click on the **Flat** button under the graph to make the lead flat, then drag that lead further right to make its curve still flatter as it exits the keyframe.

- Change the last keyframe to a **Time** of 4 seconds and a **Value** of -15. It should have a curving lead in. Make it **Flat**. It should be roughly a mirror image of the first keyframe.
- Set the second keyframe to a **Value** of -60 at a **Time** of 2 seconds. Also make this keyframe have a curved lead in and lead out that are **Flat**.

The graph should now look like the graph in Figure 7.5.1.11. If not exactly, make additional adjustments by manually moving some of the leads as necessary to roughly match the graph in the Figure.

- Click **Play** in the Animation Time Line palette to review the animation.

The door should gently swing partially open and gently swing back.

To make this door constantly flap in the breeze, the animation can be made to repeat.

- In the Animation Editor palette, make sure **Rotate Z** is highlighted. Click the button labeled **After** and, from the **Type** pop up menu, select **Repeat Normal**.
- Click on the **Fit Animation** button and observe how the graph repeats itself and covers the complete span of 10 seconds, as shown in Figure 7.5.1.12.
- Play the animation.
When you played the animation, you noticed that the door swings continuously. After 10 seconds it returns to time 0 and starts all over again, thus going continuously until you hit **Stop**.

You must have observed that when it cycles back to 0 from 10 seconds, the motion is not smooth. It jumps from the open to the close position, rather than closing smoothly, as it does the other times. This is because the 4 seconds do not divide exactly the 10 seconds. To correct this we need to either extend the 10 seconds (say to 12) or to reduce the 4 seconds (say to 2.5). In this tutorial we shall do the latter.

- In the Animation Editor palette, highlight the second keyframe and in the **Time** field enter 1.25.

- Next, highlight the third keyframe and in the **Time** field enter 2.5.

Note that we could also click and drag these keyframes to the suggested time positions, but it is more accurate to just type the desired values.

- Click on the **Fit Visible** ( ) button. Your graph should be as shown in Figure 7.5.1.13.

- Click on the **Fit Animation** ( ) button. Your graph should be as in Figure 7.5.1.14. Note how the position at time 0 matches exactly the position at time 10, which will allow the door swing to transition smoothly. Compare this with the graph in Figure 7.5.1.12.

- Play the animation.
7.5.2 Bouncing ball tutorial

In this tutorial, you will create a bouncing ball. You will first generate a model of a sphere and then you will animate it.

- Make a primitive sphere of radius 6" with its center at the origin of the reference plane. The best way to make such a sphere is by using the one-click sphere method (upper left icon in the **Sphere Options** dialog, where you can also type 6" in each of the radius fields).
- Once the proper options are set in the dialog, with Grid Snap on and the Sphere tool active, click on the origin of the reference plane.
- From the **Display** menu, set scale to 2" = 1', or whatever scale allows you to see the sphere well.
- Double click on the Grid Snap icon to invoke the **Grid Snap Options** dialog and in it adjust the **Grid Module** to 6" for all directions. Turn on the Perpendicular switch and, with the Move tool, move the sphere vertically 6", so that the bottom of the sphere is on the XY plane.

You will next move the local origin and axes of the sphere to its lower end, which currently sits on the origin of the reference plane.

With the Transform Object Axes tool click on the sphere.

*This displays the local axes and origin of the sphere.*

- Click on the axis origin and type 0,0,0 in the Prompts palette to move the object axis to the origin, as shown in Figure 7.5.2.1. Click away from the sphere to exit the edit mode.

You will next add some basic animation to the sphere.

Activate the Keyframe tool and click on the sphere.

While nothing very visible happens, the sphere has been animated and a keyframe symbol appears in the Animation Timeline palette. Also, animation information has been entered in some of the other animation supporting palettes and dialogs, as you will see soon.

- From the **Palettes** menu, open the Animation Score palette. Click on its "Name" header to invoke the **Animation Manager** dialog. Note that one object is listed in its list area on the left.

  - Click on the little triangle in front of the object name to expose its content. It contains a number of default tracks, which are not needed for this example, therefore they will be deleted.
  - To delete a track, highlight it and then click on the **Remove** button. Delete all the tracks except for **Position Z**. An efficient way of doing this is to highlight the last item, click **Remove**, which deletes it and moves the highlight to the track before it. Repeat this until you reach **Position Z**, which you do not want to delete. Delete whatever else remains at this point.
• Next, in the **Options** menu, click on **Animation...** to invoke the **Animation Options** dialog. Change the **Time Display Format** to **Frames**. This changes how time will be displayed. Click **OK** to return to the main window. Note that some users prefer to work in time units of seconds, others prefer to work using frames.

• In the Animation Time Line palette, in the **Frame** field, located at the lower portion, enter 28. Make sure the Auto Keyframe button ( ) is on.

• With the Perpendicular switch still on, move the sphere vertically 4'.

• From the **Palettes** menu, open the Animation Editor palette and click on the Fit Visible ( ) button. The palette should now appear as shown in Figure 7.5.2.2.

Notice that there is a blue line in the graph that represents the Z position of the sphere. There are also two dots, one at each end of the blue line. These dots represent the keyframes of this track. The line between the two dots shows how the track is interpolated between the keyframes. With only two keyframes, the line is straight.

On the line, before the blue dots, there are two white dots. These are at the ends of the leads, which are controls of the keyframes. You will soon see that these white dots do not have to be on the blue line and that by moving them in different positions you can control the type and intensity of the motion at the respective keyframe.

![Figure 7.5.2.2: The Animation Editor palette.](image-url)
You can add more keyframes to tracks by activating the Insert tool ( ) and clicking in the graph where we want the new keyframe to be added. You will do such an insertion next.

- Highlight Position Z in the list on the left side of the palette.

  Click on the Zoom Out tool as many times as necessary until you can see the 58 frame mark.

  Activate the Insert tool and click in the graph approximately at the point that has a 0 value (vertical axis) and a 56 frame (horizontal axis). You are assisted by a "hint" display that shows you the position you are at before you actually click to make the insertion. As soon as you click, you notice that a new keyframe or dot is added where you clicked and the blue line is now connecting this keyframe to the first two.

  Click on the Fit Visible button again. The palette should now appear as in Figure 7.5.2.3.

If you have difficulty inserting a keyframe at exactly the desired position, you can still adjust the position after the insertion has been made. While the keyframe is highlighted, its Time and Value are displayed in two numeric fields at the upper middle portion of the palette. If they do not show 56 and 0, you can type these numbers in.
You will next play the animation up to this point.

- Minimize the Animation Editor palette.
- In the Animation Time Line palette, with the mouse, move the current time symbol to 0.
- Click Play. As the time symbol in the palette moves from the 0 to the 56 frame, the sphere bounces up and down once and stops, while the time symbol continues to move to the right. When it reaches the end of the palette it goes back to the begining and the sphere bounces once again.

If we want the sphere to bounce continuously, we can do it through the Animation Editor palette.

- Open the Animation Editor palette and highlight the **Position Z** track. Click the **After** button (left side of the palette). From the pop up menu under it select **Repeat Normal**.

Click on the Fit Animation button. The palette should now appear as in Figure 7.5.2.4.

In the graph area of the palette, you will see that the original line has been repeated as many times as necessary to fill the time span available for this animation. The repetitions are shown in a light blue line.

Minimize the Animation Editor palette and from the Animation Time Line palette play the animation again. Observe how it plays continuously.

![Figure 7.5.2.4: The Animation Editor palette.](image)
As you were playing the animation and the ball was bouncing up and down continuously, you must have noticed that its motion appeared unnatural and mechanical. In order to make the sphere bounce more naturally, we need to adjust the way in which the position of the sphere is interpolated between the keyframes.

- Reopen the Animation Editor palette.

  Click on the Fit Visible button again.

The palette should appear as in Figure 7.5.2.5.

Look at the middle keyframe. Notice the line with the white circles at its ends, coming out both sides of the keyframe. These are the leads that control how the curve enters and leaves the keyframe. You will next manipulate the leads to change how the sphere bounces.

- On the first keyframe, drag the lead so that it is vertically above the keyframe. Do the same with the lead on the last keyframe. The graph in the palette should now look as in Figure 7.5.2.6.

  Play the animation. The previous adjustments cause the sphere to transition in and out of the first and last keyframes quicker, which gives the appearance of an abrupt change in direction.

- Back to the Animation Editor palette, extend the leads of the middle keyframe away from the keyframe, as shown in Figure 7.5.2.7. This makes the line formed at the middle keyframe look like an upside-down U. The middle keyframe corresponds to the upper end of the motion of the sphere and the change we just made will make the sphere appear that it is hanging in the air a moment longer.

- From the Animation Time Line palette, play the animation and observe the differences the changes you just made are causing.
Advanced motion: squash and stretch

In order to make the ball appear as though it interacts with the ground during the bounce, the sphere should be squashed as it bounces, as shown in Figure 7.5.2.8. Before we squash the sphere, however, we need to make the sphere contact the ground longer than it does currently.

- On the Animation Editor palette, with the Insert Keyframe tool active, click on time 60 (horizontal axis) and value 0 (vertical axis).

  Adjust the Lead In to be linear. You do this by pressing on the icon labeled Lead In (upper portion of the palette) and, when a menu pops up, you select the second item. The graph should now be as shown in Figure 7.5.2.9.

- From the Animation Time Line palette, play the animation. Notice that the sphere seems to hesitate or pause when it is in contact with the ground. This may look unnatural now, but will look right once the sphere is contracting and expanding during this time.

- In the Time Line palette, set current time to 0.

  From the Options menu, open the Animation Manager dialog. In it highlight the name of the sphere and click on the Add Tracks... button. This invokes the Add Tracks dialog, which contains a list of tracks that can be added to the active animated object (the sphere).

  Select the Scale Z track and click on OK to close the dialog. If you now click on the triangle before the sphere name to expose its content, you see that it contains two tracks: Position Z and Scale Z.

  Open the Animation Editor palette.

  Go to the list of tracks on the left and click on the little hammer, in the last column of the Position Z row. This erases the hammer symbol and the display of the graph for Position Z.

  For Scale Z, there should be a keyframe at time 0. Set this first keyframe to a value of 0.7, as shown in Figure 7.5.2.10.
• Insert the following keyframes at times and values: 20 and 1; 56 and 1; 58 and 0.5; 60 and 1.
  • Highlight the Scale Z track, click on After and from the menu under it select Repeat Normal.
  • Click on the Fit Visible button.

If you are setting up an animation that is meant to cycle many times, you will want the first keyframe to have the same value as the last keyframe. This will make a smooth transition between cycles.

• In the Animation Editor palette, check the Connect Beginning And End option. When you do, you notice that the last keyframe changes to the same value as the first, which is 0.7.

The graph for the Scale Z track, in the Animation Editor palette should now look as in Figure 7.5.2.11.

• In the Animation Time Line palette, click play. The sphere should now look as though it is being compressed when it hits the ground. We can exaggerate the rebound by making the sphere to actually expand beyond its normal shape when it touches the ground.

• Reopen the Animation Editor palette. In it, adjust the lead of the first keyframe so that it is pointing vertically, as shown in Figure 7.5.2.12.
  • Do the same with the Lead In for the second keyframe.
  • Set the Lead Out of the second key and the Lead In of the third key to be linear. Recall that you do this by selecting the second item from the respective pop up menu.

• From the Animation Time Line palette, play the animation.

You notice now that, as the sphere leaves the ground, it stretches up. It returns to its normal shape as it approaches the top of its flight.

This can be taken a step farther and given some lateral scale that is inversely proportional to the vertical scale. In other words, as the sphere is compressed vertically, it should expand horizontally.
• Set current time to 0.
• In the Animation Manager dialog, add two more tracks for Scale X and Scale Y.
• In the Animation Editor palette, move the first keyframe of Scale X up to a value of 1.3 at frame 0. This is the point at which the sphere is squashed down at maximum compression.
• Add a second key at frame 6 with value 0.6. This will cause the sphere to get thinner as it is rebounding up into the air, reinforcing its vertical motion.
• Make two keys at frames 16 and 56 of value 1. The sphere is at a normal size during its flight.
• Add a final key at frame 60 of value 1.3 to complete the cycle, as shown in Figure 7.5.2.13.

Scale Y will be identical to Scale X and will thus be constructed by copying and pasting all the keyframes from one track to the other.

• Select all the keyframes on Scale X. To do this, press ctrl (Macintosh) or shift (Windows) and click on the line for Scale X. This will select the entire animation curve. Select Copy from the Edit menu.

• Highlight Scale Y in the Animation Editor palette and select Paste from the Edit menu. As soon as you do, the Paste Control Points Options dialog appears (Figure 7.5.2.14).

• Change Remove Control Points to All. This will delete all existing control points in the time span we specify. Leave Time Span at the default Keep Same. Click Ok.

Now, Scale X and Scale Y should be the same. Highlight each, click on After and from the menu under it select Repeat Normal.

Click play in the Animation Time Line palette to see the results of the animation.
7.5.3 Animating surface styles

In this tutorial, you will continue the previous one with the bouncing ball and you will assign colors to the ball and the floor, which change as the ball bounces.

There are two ways in which the color of an object can change over time. Remembering that colors in form•Z are handled with surface styles, one way of changing color is to switch from one surface style to another at appropriate times. The other way is to animate the individual parameters of the surface style itself. This tutorial will cover both methods of changing an object color.

You can continue using your own file, that is, the one you saved in the previous tutorial session, or you can open the one we provide and continue working with this. It is called "Surface Styles Base." Here is what you will find in the new file or, if you continue your own file, here is what you should add, before you proceed: (1) A roughly 40' x 40' rectangle (surface object) in the middle of the XY reference plane representing the floor on which the ball is bouncing, as shown in Figure 7.5.3.1. (2) Four surface styles labeled green, yellow, red, and floor. The first three have plain colors corresponding to their names. The fourth is the Checker color shader with its default settings and uses two shades of blue to color the tiles of the checker pattern. The floor surface style is currently assigned to the floor rectangle and the ball is colored green.

You may produce a (static) RenderZone rendering to confirm that this is in fact the case, which will look something like that in Figure 7.5.3.2. The rendering in the Figure is a perspective and uses two Distant lights at their default intensity, with Shadows on and Per Light. Needless to say that, since this tutorial is not about rendering, you should render it according to your preferences.
Animating the color of the ball

- Select **Interactive Shaded** and, from the Animation Time Line palette, play the animation to confirm that it is still in place.

You notice that Open GL, which is behind the **Interactive Shaded** mode of rendering, does not render procedural textures and the floor appears as a flat blue. Do not be concerned about it. You will render the animation with **RenderZone** at the end of the session.

Your next task is to make the bouncing sphere change color at each bounce.

- In the Animation Time Line palette, set current time to 0.
- Open the **Animation Manager** and in its list highlight the sphere name.
- Click the **Add Tracks**... button and, when the **Add Tracks** dialog is invoked, from the **Category** pop up menu, select **Object Attributes**.
  - Highlight **Surface Style** and click **OK** to exit **Add Tracks**.
  - Click **OK** on the **Animation Manager** dialog to exit it.

- In the Animation Time Line palette, set current time to 60 frames.
- In the Surface Styles palette, set the active color to **yellow** and, with the Color tool active, click on the sphere.
  - In the Animation Time Line palette, with the mouse, move the current time between roughly 1 and 3 seconds. Observe that the sphere is not only moving in space, but its color is also changing from green to yellow.

- In the Animation Time Line palette, set current time to 120 frames.
  - Set current color to **red** and, with the Color tool active, click on the sphere.
  - Open the Animation Editor palette and highlight the sphere's **Surface Style** track.
  - Use the Insert tool to add another keyframe at frame 180 and value **red**.
  - Click the **After** tab and from the **Type** menu select **Repeat Normal**.

The Animation Editor palette and its graph should now look as in Figure 7.5.3.3. From the Animation Time Line palette, play the animation and observe how the ball changes color as it bounces.

![Figure 7.5.3.3: The Surface Style graph in the Animation Editor palette.](image)
Animating the color of the floor

Next, the color of the floor object will be set up to change smoothly over time. This will be done by animating shader parameters, which is the second method available for changing color of animated entities.

Before you proceed a little warning. You will only be able to see the effects of your manipulations after you render your animation, at the end of the session. That is, you will not be able to preview your settings while they are still in progress, as you were able to do with the previous animations. The reason is, of course, that you will be animating parameters of procedural textures, which are only available in RenderZone and not in Open GL.

- In the Animation Time Line palette, set the current time to frame 60.
- In the Surface Styles palette, double-click on the Floor shader to invoke the Surface Style Parameters dialog. In it, you notice that Checker is displayed for Color shade. Click the Options... button next to it, to invoke the Checker Options dialog, shown in Figure 7.5.3.4.
  - Click on the red dot next to the Color 2 box. From the menu that pops up, select Add Track.
  - Click OK to return to the Surface Style Parameters dialog and click OK again to return to the modeling window.
  - Open the Animation Editor palette and in it highlight the Color 2 parameter of the animated Surface Style.
  - Use the Insert tool to add a key for Color 2 at frame 70 with the same value as the first keyframe. Close the Animation Editor palette.
  - In the Animation Time Line palette, set the current time to frame 120.
  - Double-click on the floor surface style again and, when the Surface Style Parameters dialog is invoked, click on Options... to invoke the Checker Options dialog.
  - Click on the box for Color 2 and when the Color Picker dialog appears, change the color to yellow, or to some color you may prefer, other than blue.
  - Click OK, OK, and OK to return to the modeling window.
  - Open the Animation Editor palette and use the Insert tool to add another keyframe at frame 170 with the same value as the original value of Color 2.
  - Add another keyframe at time 180, as shown in Figure 7.5.3.5.
  - Click the After tab and for Type select Repeat Normal.
  - Click the Before tab and for Type select Repeat Normal.

![Checker Options dialog](image.png)

*Figure 7.5.3.4:* The Checker Options dialog.
Before proceeding with the final stage of this animation, which is the rendering, observe the displayed Animation Editor palette and how the actual colors are displayed with the vertical value axis.

You could have actually changed the color from the Animation Editor palette, rather than going to the Checkered Options dialog, which you did a bit ago. If you double click on a color box of the value axis you will invoke the Color Picker dialog, where the color can be changed.

Rendering the animation

- From the Display menu, select Generate Animation....

- When the Animation Generation dialog is invoked, set the desired rendering and other parameters. One of the settings in the dialog is about which rendering mode you will use. You can experiment with different ones, but for this tutorial select RenderZone, whose options should already be set from the still rendering you did at the beginning of this session. If not, you can do it now. Once your settings are in place, click the Generate Animation... button (not the OK button).

form•Z proceeds immediately and executes the rendering one frame at a time. You are kept informed about the total number of frames to be rendered and which frame is currently being rendered. How fast the rendering will proceed depends, of course, on the usual factors, such as number of frames, the rendering effects, and the size of the frames. For this example, you may actually want to reduce the size of your frames, which is done from the Image Options dialog, invoked from the Display menu.

When the rendering is completed, the animation is saved according to the settings in the Animation Generation dialog. The defaults will be to save all the frames in one file and in the form•Z animation format (.fan). The .fan file can be played back by selecting the Play Animation... item in the Display menu.

Once you have a .fan file you can also generate a Quick Time movie, as follows:

- From the File menu select Export Animation... and, when the Open file dialog appears, find the .fan file, set the place and the name of the QR movie, and click on Open.

- When the Animation Frame Export Options dialog appears, click OK; when the Save file dialog appears, make sure Quick Time Movie is selected for File Format and click OK; when the Compression Settings dialog appears, click OK.

After a few minutes, a QT movie is generated and stored where you told it. Play it.
7.5.4 Animating lights and cameras

Lights and cameras can be animated just like any object. However, they typically have more parameters that can be animated. For example, in the case of lights, their intensity can be animated, but also their location and their center of interest can be animated separately. The same is true for animated views, which are also called cameras.

This tutorial will cover mostly parameters of lights and cameras that are unique to these types of entities. A light and a camera will be animated in a way such that their location and center of interest move independently of each other. These entities will be applied to a variation of the scene with the bouncing ball that you worked with in the previous two sections.

Again, you may continue with the file you saved at the end of section 7.5.1 (called "Bouncing Ball") or you may open and use a new file that we have set up for this exercise. It is called "Lights and Cameras Base." In addition to the bouncing ball, following is what is new in this file:

(1) A 10' high brick wall that delineates part of a room. (2) A gray surface for the floor. (3) Three surface styles: a yellow for the ball, rough gray for the floor, and a procedural brick texture for the wall. (4) Two lights, one of which is a cone light that will be animated. (5) Two saved views, one of which is a perspective camera that will also be animated. These entities are shown in Figure 7.5.4.1.

* Next, in the Views palette, select the preset view called "main view" and from the **Display** menu select **RenderZone**. Your rendered image should be as in Figure 7.5.4.2.

You can observe the cone light that is shining on the scene and is currently stationary. The view of the scene is also stationary. You will first animate the camera to slowly pan by the scene, while it remains focused on the bouncing ball.
Animating the camera

- In the Views palette, verify Visibility of "main view" is on.
- From the Views menu, select Top.
- Double click on the Fit All icon (Window tool at the bottom of your screen) to invoke the Zoom Options dialog. In it, under Fit All turn on Include Lights and Include Views/Cameras.
- Click on the Fit All tool. All the visible entities of your project, including lights and cameras, are now displayed on the screen, as shown in Figure 7.5.4.3(a).
- In the Lights palette, in the visibility column (this is the one with the diamonds) click next to the cone light, to ghost it or to even make it completely invisible.
- Make sure the time is at 0 in the Animation Time Line palette.
- With the Keyframe tool active, click on the camera.
- Move the current time ahead to 5 seconds.
- Make sure the AutoKey option is on.
- With the Move tool active, click on the eyepoint end of the view and drag it up to another location in the scene, as shown in Figure 7.5.4.3(b).
- In the Views palette, make "main view" active and, in the Animation Time Line palette, click on the Start button to set the time to 0.
- Click the Play button. Observe how the whole scene rotates as it follows the motion of the animated camera.

You will next adjust one of the parameters of the camera.

- Open the Animation Editor palette.
- Highlight the Focal Length track of the animated "main view."
- You may want to turn off the hammers for all the other items except for focal length. This will hide them and make it easier to work with focal length.
- With focal length highlighted, use the Insert tool to add a keyframe at 5 seconds.
- Adjust the value of this keyframe to 70.
- Click on the Fit Visible icon. The graph in the Animation Editor palette should be as shown in Figure 7.5.4.4.

Close the Animation Editor, play the animation, and note how the camera is zooming in.
**Animating the cone light**

You will next animate part of the cone light so that, while its location remains stationary, its focal point (or center of interest) follows the motion of the ball. You will actually do this by copying the animation setting for the ball to the center of interest track of the light.

- Stop the animation if it is still running.
- In the Animation Time Line palette, set the current time to 0.
- In the Lights palette, make the cone light visible again, if you turned it off earlier.
- Activate the Keyframe tool and click on the cone light. Note that, to pick a cone light you click on its line of sight and not on the representations of the cones, which are not pickable.
- Open the Animation Editor palette and in it turn on the hammer (last column to the right) for the Position Z track of the bouncing ball. This will make the track visible.
- Select the entire Position Z track by clicking on its graph while pressing `ctrl` (Macintosh) or `ctrl + alt` (Windows).
- Once the line is selected, from the **Edit** menu, select **Copy**. Or you can press `cmd+c` on the Macintosh or `ctrl+c` on Windows.
  - Highlight the Center of Interest Z track for the cone light.
  - From the **Edit** menu select **Paste** or use the key commands `cmd+v` (Macintosh) or `ctrl+v` (Windows).
  - When the **Paste Options** dialog is invoked, click **OK** to accept the default settings.
  - While the Center of Interest Z track is still highlighted, click the **After** tab and from the **Type** pop up menu select **Repeat Normal**.
  - The Animation Editor palette should now be as shown in Figure 7.5.4.5. Close it.
  - In the Animation Time Line palette, click Play.

![Figure 7.5.4.5: The Animation Editor palette after copying the motions of the ball to the center of interest of the cone light.](image)

The center of interest of the cone light should now be constantly moving and is aiming at the bouncing ball.
You will next animate one of the parameters of the cone light so that its lighting transforms from a concentrated light effect to a more diffused lighting.

- In the Animation Time Line palette, set the current time to 0.
- From the **Options** menu, open the **Animation Manager** dialog.
- In it, highlight the cone light and click **Add Tracks**...
  - In the **Add Tracks** dialog that is invoked, from the **Category** menu, select **Cone**.
  - From the list of **Available Track Types** that is displayed select **Outer Angle**.
  - Click **OK** to exit **Add Tracks** and click **OK** again to exit the **Animation Manager** dialog.

- In the Animation Editor palette, highlight **Outer Angle**.
- Activate the Insert tool and add a key 5 seconds with a value of 50.
- Click on the **Fit Visible** icon.

The graph in the Animation Editor palette should look as in Figure 7.5.4.6.

While "main view" is active in the Views palette, from the Animation Time Line palette, play the animation to preview how the cone light behaves. You may also observe your animated entities from a distance. To do this, select the "axo view" in the Views palette and play the animation again.

You can finally render your animation to watch the results of the tutorial. Before you do, make sure you select "main view" again. Then, from the **Display** menu, select **Generate Animation**... and watch the frames being rendered one at a time.
7.5.5 Animating parts of objects

One can animate individual points, segments, outlines, or faces of objects. In this tutorial, you will animate points. In the first part, which will illustrate the use of the Keyframe tool, a flat roof will be reshaped into a gable roof and then a hipped roof. In the second part, which will illustrate the use of the Animate along Path tool for animating parts of objects, the apex point of a pyramid will be moving about a circle.

Keyframing points

First build a block with a segment that runs through the middle of the top face. The size of the block does not really matter. After you draw it, use the Insert Segment tool (found in the Line Edit tool palette) and Midpoint snap.

The points that are a part of the top of the object need to be keyframed and then animated.

- Set Topological Level to Point and with the Pick tool select the two points at the end of the segment you inserted (shown with a bullet).
- With time set to 0 in the Animation Time Line palette, activate the Keyframe tool and click in the project window.
- In the Animation Time Line palette, move the time to 2 seconds.
- The two points should still be selected.
- Turn on the Perpendicular Switch and, with the Move tool active, drag the two points upwards 5’.
- Turn the Perpendicular Switch back off.

If you play the animation, the block is now forming a gable roof as it approaches the 2 seconds mark. You will next move the points some more to form a hipped roof.

- Set time to 4 seconds. Topological Level should still be at Point.
- Set the Directional Snap to Ortho, with the Move tool click on one of the ends of the gable roof, and drag it inwards about 7’.
- Repeat the same on the opposite end of the roof.

The block with the hipped roof now has non-planar surfaces on the ends. To correct this, use the Insert Segment tool to insert segments between points 1 and 2, and 3 and 4 (the points marked with bullets).

Rewind the animation and click Play to watch how your block changes shape dynamically and becomes first a gable and then a hipped roof.
Animating point along a path

Open a new project so that you can start fresh.

- Draw a pyramid roughly as shown in Figure 7.5.5.2(a). Then, working on Top view, draw a circle that preferably touches the points of the pyramid (even though it is not an absolutely necessary condition).
- With the Break tool, click on the circle where marked with a bullet (Figure 7.5.5.2(b)). This turns the circle into an open arc.
- Working in Front view, move the circle from the base of the pyramid to the height of its apex, as shown in Figure 7.5.5.2(c).

You now have the objects you need to animate a point along a path.

- With Topological Level set to Point and the Animate along Path tool active, click on the apex of the pyramid and then on the circle.

The apex point of the pyramid jumps to the point where the circle was broken and this shape of the pyramid becomes a keyframe (Figure 7.5.5.3(b)).

- From the Animation Time Line palette play the animation and observe how the apex moves around the circle continuously. Two instances of this motion are shown in Figures 7.5.5.3(c) and (d).

In a similar fashion segments, outlines, and faces can be animated along paths, except that the entities at these other topological levels are sensitive to the alignment options that are set in the Animate Along Path Options dialog.

You should experiment on your own with animations of parts of object at topological levels other than point.
7.5.6 Animating deformations or dancing board

Objects can be made to alter their shapes over time, using the Animate Deformation tool. This tutorial will demonstrate some of the features of the Animate Deformation tool, while creating an animation of a board doing a backflip.

Open the "Deformations Base.fmz" file that can be found in the tutorials folder on your program CD. After you open it, play the animation. You observe that a board is already animated to do a backflip. However its shape does not change, which you will do next. Before you do, you may also want to take a look at the Animation Editor palette to see what tracks have been animated. If the tracks are not visible, turn on the little hammers in the fourth column. Your palette should look as in Figure 7.5.6.1 and the model of the board in the file as in 7.5.6.2.

You will first create the animated deform entities that will be used in the animation.

- Activate the Animate Deformation tool and in the Tool Options palette (or the Animate Deformation Options dialog), set Model Type to Facetted, drag the Density slider all the way to the right, from the Method pop up menu select Bulge, and from the Base Reference Plane menu select Object (XY).
- With the Animate Deformation tool active, click in the project window. This creates an animated deformation entity in the project.
From the **Method** menu, select **Radial Bend** and with the Animate Deformation tool click in the project window once more.

This creates one more animated deformation entity, which is a radial bend entity. The previous one was a bulge entity. You will next observe these entities in the Animation Score palette.

Open the Animation Score palette. It should be as in Figure 7.5.6.3(a). **Bulge(Object)** and **Radial Bend(Object)** are displayed at the end of its list.

For these two deformations to work and affect the object they need to be nested, which you will do next.

- In the Animation Score palette, drag the board object into the Radial Bend entity.
- Next, drag the Radial Bend into the Bulge entity.

After you open the Bulge and Radial Bend entities, the list in the Animation Score palette should be as in Figure 7.5.6.3(b).

The bulge deform will be used to exaggerate the squashing of the board as it warms up before attempting a backflip.

- Open the Animation Editor palette and in it highlight the **Anchor Scale Y** track under **Bulge**.
- Now use the **Insert** tool to add the following keyframes, where the numbers are for **Time** and **Value**, in this order:
  
  1.5 and 35; 3 and 0; 3.7 and 0; 4.3 and 60; 4.9 and 0.

- Play the animation to see the results of the bulge deformation.

Two frames are shown in Figure 7.5.6.4. The track with the new keyframes in the Animation Editor palette are shown in Figure 7.5.6.5.
Figure 7.5.6.5:
The Animation Editor palette after new keyframes were added to the Anchor Scale Y track.
(a) The Anchor Scale Y track only and (b) all the tracks.
The Radial Bend deformation will be used next to make the board flex as it flies through the air.

- Open the Animation Editor palette and in it highlight the **Angle** track of the **Radial Bend**.
- With the **Insert** tool add the following two keyframes: 4.5 and 0; 5 and -25.
- Play the animation and observe how the board bends backwards as it begins its jump.

Some frames of the new deformations are shown in Figure 7.5.6.6 and the graph of the **Angle** track in Figure 7.5.6.7.

![Animation Editor](image)

**Figure 7.5.6.6:**
Frames of the new animation.

**Figure 7.5.6.7:**
The Animation Editor palette after new keyframes were added to the **Angle** track.
You will next make the board bend forward as it is flipping.

- Open the Animation Editor palette and highlight the **Angle** track of the **Radial Bend**.
- Add the following keyframes:
  5.7 and 115; 5.9 and 115; 6.4 and 30.

- Play the animation. The board bends in the air and lands on the ground slightly bent.

Some frames of the new deformations are shown in Figure 7.5.6.8 and the graph of the **Angle** track in Figure 7.5.6.9.
Lastly, you will make the board stand straight after landing, take a bow, and stand straight again.

- Open the Animation Editor palette and in it highlight the **Angle** track of the **Radial Bend**.
- With the **Insert** tool add the following keyframes:
  6.7 and 30; 7.2 and 0; 8 and 0; 8.6 and 90; and 9.2 and 0.
- Play the animation.

Some frames of the new deformations are shown in Figure 7.5.6.10 and the graph of the **Angle** track in Figure 7.5.6.11.

Concluding, we recommend that you experiment further with your own settings.

Figure 7.5.6.10:
Frames of the new animation.

Figure 7.5.6.11:
The Animation Editor palette after more keyframes were added to the **Angle** track.
7.5.7 A movie within a movie

Movie files can be used as texture maps for any of the four shaders of a surface style, namely Color, Reflection, Transparency, or Bump. The practical implication of this is that prerecorded movies can be included in an animation and run within it. This tutorial will show you how to load a movie file as a color map. It will also show you how to set up the rate at which the movie will play.

- Open the file "Color Maps Base," which contains the models of the two television sets, shown in Figure 7.5.7.1.
- In the Animation Time Line palette set current time to 0.
- In the Surface Style Parameters dialog, double click on the "tv tube" surface style and when the dialog appears, from the Color menu, select Color Map. Then click on the Options... button next to it.
- In the Color Map Options dialog, click on the Load... button.
- When the Open File dialog appears, select "spaceman.mov," the Quicktime file supplied with the tutorial.

Actually, you can select any .mov file you may prefer, including one you may have rendered for one of the tutorials you did earlier.
- In the Frame Time field of the still open Color Map Options dialog, enter 0, then press on the red dot next to it and, from the menu that pops up, select Add Track.
- Make a note of the Duration of the movie, shown a couple of lines above Frame Time. Then click OK to exit the Color Map Options dialog, and click OK again to exit the Surface Style Parameters dialog.

![Figure 7.5.7.2: The Color Map Options dialog after "spaceman" was selected.](image-url)
• In the Animation Editor palette, find and highlight the track called "Frame Time."
• In the graphic area, add another keyframe with the **Insert** tool.
  Set its value to 3 and time to 3.
• Make the leads of the animation track **Linear** and close the Animation Editor palette.
  The graph in the Animation Editor palette should look as in Figure 7.5.7.3.

• From the **Display** menu, invoke the **Interactive Shaded Options** dialog and in it turn on **Textures**, if not already on.

As you exit the dialog, your model should be rendered in interactive shaded mode and the space-man should appear on both TVs.

• From the Animation Time Line palette, play the animation. You should see the movie playing to both TV screens. Since the movie is a "texture" that is associated with a surface style, it will play on all the surfaces where the surface style is used.

---

*Figure 7.5.7.3: The Animation Editor palette with the graph for "Frame Time."*
You will next apply some motion to one of the TVs, so that the movie plays while the TV is rotating.

- Select all the objects that make up the model of the plasma screen.
- Set current time to 0.
- In the **Animation Group Options** dialog turn on **Center Of Bounding Volume**.
- With the Animation Group tool active, click on the project window.
- Open the **Animation Manager** dialog and in it highlight the animation group just created.
- Click the **Add Tracks** button and add a **Rotation Z** track.
- In the Animation Editor palette, highlight the **Position Z** track and, with the Insert tool, add a keyframe at 2 seconds and a value of 45° higher than the first keyframe.
- Add a third keyframe at 4 seconds with a value equal to the first keyframe.
- Select the entire motion track, which you do by clicking on it while pressing `ctrl` (Macintosh) or `ctrl+alt` (Windows).
- Make all the leads of this track flat by clicking the **Flat** button.

The graph should now be as shown in Figure 7.5.7.4.

The plasma screen TV will now rotate back and forth and play the movie at the same time. Play the animation as before to verify this.

Finally you may render the animation by selecting **Generate Animation...** from the **Display** menu and using **RenderZone** for rendering mode.

![Animation Editor](image)

**Figure 7.5.7.4:** The Animation Editor palette with the graph for the **Position Z** track.
absolute coordinates: Coordinates measured relative to a constant origin, which is the origin of the world space for both the **Cartesian** and **polar coordinate systems** (see respective entries). For an alternative method of measuring coordinates, see **relative coordinates**.

accurate glow: The effect of light shining in a foggy environment simulated in an accurate manner. This is achieved by placing small particles in the light's volume and computing the illumination of these particles. Glow can be generated for cone, point, and projector lights. Available in **form-Z RenderZone Plus**. See also **simple glow**.

accurate intensity: Light intensity defined by physically accurate units. Can be used optionally in **form-Z RenderZone Plus** when illuminating a scene. See also **basic intensity**.

adaptive meshing: A method employed during radiosity processing, which subdivides the **radiosity mesh** based on light and shadow contrast to achieve smoother illumination. Available in **form-Z RenderZone Plus**.

additive weight: A metaform parameter that adds the volume of one metaform to another.

align: An operation that rearranges a set of objects/elements in a way such that their maxima, minima, or middle points are placed on a straight line. This operation is typically combined with the **distribute** operation (see respective entry).

alpha channel: One of the background options available in **form-Z RenderZone Plus**. It is a method by which the percentage of background that is visible at each transparent or antialiased pixel is calculated and saved with the image. This facilitates compositing such images with different backgrounds in image processing applications.

ambient light: A type of light that illuminates all the surfaces equally and which is always present. Its intensity and color are user defined parameters.

ambient occlusion: (or AO for short) A global illumination technique available in **form-Z RenderZone Plus**, that performs a better distribution of the otherwise constant ambient light. For a given point to be rendered, the ambient light is decreased, based on close by objects, that may occlude the point. This leads to an improved lighting from ambient light. Ambient Occlusion is the simplest GI technique. Its effects are still not very accurate, but it is easy to control and can be computed fast.

amplitude: A parameter of the bump shaders that controls the three dimensionality of the bumps. Positive values produce raised bumps and negative values produce indented bumps. Only available in **RenderZone Plus**. See also **bumps** and **bump shader**.

analytic primitives: The basic analytic geometry shapes, namely cube, cone, cylinder, sphere, and torus.

angular dimensions: They are used to measure and post the size of an angle in degrees. They are drawn with an arc shaped dimension line. They appear similar to the **arc dimensions** (see entry), but they measure angles rather than lengths of arcs.

animation: A sequence of images displayed one after another fast enough to give the impression of motion.

antialias: A method by which diagonal edges in rendered images, which would normally appear as staircases, are corrected and made to appear smooth. The antialias method actually produces a visual illusion by replacing the color of the pixels in the area of an edge with gradual mixtures of the colors on each side of the edge.

antialiasing: See **antialias**.

arc: A parametric entity consisting of a single curved segment and no points. It can be a complete or a partial circle. In drafting the arc is distinct from the polyarc, which is an arc approximated by a circular vector line. Similarly in modeling, the arc is initially generated as a parametric object. It can subsequently be transformed to a polygonal entity.
**arc dimensions**: They are used to measure the length of arcs, and can only be applied to open arc elements. Their dimension line is drawn in the shape of an arc. They appear similar to the *angular dimensions* (see respective entry), but they measure the length rather than the angle of an arc.

**area element**: A special type of a drafting element that is used to select and manipulate other drafting elements. It is always a closed shape and the portions that are inside versus outside its boundary line are significant for most of the operations with which it is used. Only one area element may exist at any given time. The area does not appear when a drafting drawing is plotted. • A drafting tool (modifier) that generates an area element when a closed shape is drawn. • One of the topological level modifiers in drafting that makes it possible to select an area element.

**area light**: One of the light types available in *form•Z RenderZone Plus*. It is defined by the geometry of an object and emits light diffusely off the object’s faces.

**area sampling**: A method for determining the color of a pixel when an image map, which has a resolution larger than the rendered image, is applied. It assigns the average value of the image pixels that are in the area of the screen pixel to be rendered. For another method, see *point sampling*. Area sampling usually produces better image quality than point sampling, which is faster. These options are only available in *form•Z RenderZone Plus*.

**Arrow tool**: See *pick*.

**aspect ratio**: The ratio of the width to the height of a camera’s image.

**associative** (or associated): An attribute of the *dimensions, leader lines*, and *hatch pattern* elements (see respective entries), which establishes a link to the points used to create the element. Associative elements are automatically updated when the positions of the points with which they are associated change.

**Attach**: A *form•Z* tool used to execute attachments. See also *attachment*.

**attachment**: The operation by which one object (or some part of an object) is moved and positioned relative to another object (or some part of another object). Attachments can be applied relative to points, segments, or faces.

**atmospheric light**: An effect of distant lights, which can be used during radiosity processing to simulate illumination through the atmosphere. Available in *form•Z RenderZone Plus*.

**automatic topological level**: An intelligent topological level that is context sensitive. That is, it switches according to the entity type close to which the mouse cursor is.

**auto scrolling**: The automatic panning of the window and its displays when the cursor reaches the edge of the window.

**AVI** (Audio Video Interface): A Microsoft API and file format for storing audio and video data including animations. The AVI API is available only on Windows computers. AVI files can be created and played through QuickTime.

**axes of Cartesian space**: The three axes, labeled X, Y and Z, of the Cartesian coordinate system, which in *form•Z* modeling is used as the 3D *world space* (see respective entry). In drafting, a 2D Cartesian space is used, whose axes are X and Y. The Cartesian or world axes are also called orthogonal axes. All Cartesian axes intersect at the same point, which is called the *origin* (see entry) and has coordinate values (0,0,0) or (0,0) in 3D and 2D spaces, respectively.

**axes of object**: See *object axes*.

**axes of reference plane**: See *reference plane*.

**axis of revolution**: The axis about which a profile is revolved to derive an *object of revolution* (see also respective entry).

**axonometric**: One of the 3D view types available in *form•Z*. It is derived by projecting the 3D coordinates of objects to the surface of the screen. With this view type, lines which are parallel in 3D space remain parallel, but their true lengths are not preserved.
background: The color, pattern, or image used to render the parts of an image that are not covered by objects in the modeling scene. Background types available in form•Z RenderZone Plus are: plain color, graduated color, sky, procedural textures, precaptured textures (images), environment, and alpha channel. See respective entries.

basic intensity: Light intensity defined by a simple percentage factor and used by all rendering types when illuminating a scene. A RenderZone rendering may optimally use accurate intensity (see also respective entry). A rendering based on a radiosity solution (see also respective entry) always uses accurate intensity.

beveling: A variation of the fit fillet operation (see respective entry) that flattens the edges of solids or the points of vector lines (polylines). For solids see rounding. For vector lines, it is one of the line editing operations available in both modeling and drafting. It replaces the vertex of a line with a segment, whose size is determined by the user.

Bezier: A type of a spline curve. In form•Z, a number of Bezier variations are available, including the complete Bezier that can be generated at any degree, the broken and continuous Bezier that consist of 2nd (quadratic) or 3rd (cubic) degree pieces. See also controlled curves.

Bezier bend: One of the deformation operations that proportionally resizes the middle region of a meshed object.

bi-cubic Bezier patch: See patches.

blend: A parameter of the bump shader that controls the transition of a bump to the surface it is on. Small values produce sharper transitions and higher values produce smoother transitions. Only available in form•Z RenderZone Plus. See also bump and bump shader.

blending surfaces: See rounding.

bolt: See screw.

Boolean operations: They are the union, intersection, difference, and split operations, which can be applied to solids, closed 2D modeling shapes lying on the same plane, and closed drafting elements. See respective entries.

boundary representation (b-rep): See faceted.

break: A form•Z line editing operation, available in both modeling and drafting, by which a vector line (polyline) is broken at the point where the mouse is clicked.

B-splines: A type of a spline curve whose control parameters include knots. It is a NURBS with unit weights. It can be generated at any degree. See also controlled curves and splines.

bulge: One of the deformation operations that proportionally resizes the middle region of a meshed object.

bullet marker: A little round icon used in palettes to represent the snapability status of entities.

bump: A three dimensional effect that is added to a surface through rendering. Only available in form•Z RenderZone Plus.

bump map: An image map used as a filter for producing bump effects on surfaces rendered in form•Z RenderZone Plus.

bump shader: It creates small surface disturbances that produce the effect of sculpted surfaces. Bumps may be produced by either procedural or precaptured texture maps. See also shader.

cage: A low resolution object that may be used as a temporary substitute for a high resolution object, during the process of setting up scenes for still rendering or animations. In form•Z it is also a tool that derives low resolution approximations of high resolution objects.

calibration: The act of spatially synchronizing devices such as a digitizer with the 3D Space.
camera: Viewing parameters that are used to look at a scene. They include the viewer position, viewing location, image size, and other type dependent parameters.

cap: An operation that generates a surface from a closed sequence of edges.

Cartesian coordinate system: An orthogonal coordinate system based on three axes labeled X, Y, and Z. It is the coordinate system used in form•Z for storing the geometry of its entities. For a different coordinate system, also available in form•Z for input purposes, see polar coordinate system.

circle: A closed shape all points of which are at the same distance from its center. In form•Z modeling, the circle is a polygon with a high number of segments (default 24). In drafting, the circle can be a polygon (called polyarc) or a parametric arc. See polyarc and arc.

c-curve: See controlled curve.

center-edge transparency: A type of transparency available in form•Z RenderZone Plus that is calculated on the basis of the normal of a surface. When applied to curved surfaces, such as spheres, it highlights their curvature by applying different transparencies to central surfaces, whose normals are almost parallel to the direction of the light, and different transparencies to perimeter surfaces, whose normal approaches being perpendicular to the light.

center of interest: The target point of the line of sight (see entry). The point where we focus our eyes when looking at a scene. • The path the camera’s center of interest follows during an animation.

centroid of object: A point in the center of an object, which may be calculated as the center of gravity of a solid, as the average of its coordinates, or as the center of its bounding box.

client: Used with network rendering to represent the computers over which a rendering is distributed.

c-meshes: See controlled meshes.

c-objects: See controlled objects.

color shader: It computes the color pattern of a surface, which may be a plain color selected form the Color Picker, a procedural texture (wrapped or solid), or a precaptured texture (image). See also shader. Only available in form•Z RenderZone Plus.

color temperature: A parameter of photometric light intensity, which determines the output color of a light source from a given temperature at which the light burns. Available in form•Z RenderZone Plus.

composite curve: A line sequence that may consist of parametric splines, arcs, and/or straight lines. The complete curve is parametric and, when edited, the different types of components behave differently.

compound element: A composite drafting element where more than one elements are grouped together as a single element. Compound elements may be created by composing other previously created elements, or they may result from a drafting operation, such as a Boolean operation. • A drafting tool (operator) that is used to create compound elements. • A drafting topological level modifier that makes it possible to select a compound element.

cone light: One of the light types available in form•Z RenderZone Plus. It emits rays from a given point in the direction defined by a conic shape that is associated with it. It is also called spot light.

cone of vision: A graphic representation of the viewing parameters. Also, a menu item and environment where the viewing parameters can be set.

connect: A form•Z line editing operation, available in both modeling and drafting, by which two separate open vector lines (polylines) are connected into one by generating a segment between two end points of the vector lines.

constant falloff: See falloff.

contextual palette: A pallete whose content adjusts to the entity that is clicked when the palette is invoked.

continuous copy: A form•Z tool (modifier) that causes the geometric transformations to make copies of the object/element.
controller: A function, such as a Bezier spline, that represents the animation sequence of a track.

copy: A menu item and operation that places copies of the selected objects/elements into the Clipboard.

continuous line: See *composite curve*.

contour line: A line used to represent topography. It represents the section of a land form with a horizontal plane at a certain elevation. Contour lines are typically derived from spot elevations measured on the land. Contour lines are by definition always closed, except that they may appear open when trimmed to the borders of a site or region. Contour lines do not intersect, except in the rare cases of overhangs. Contour lines are typically derived at constant increments called inter contour intervals. All the points of a contour line are at the same elevation, which typically measures the distance from the surface of the sea. While contour lines are commonly used to describe land forms, they can also be used to describe any 3D shape. In form-Z, contour lines can be drawn as any vector line, or they can be imported. form-Z also offers a tool with which contour lines can be derived from any solid and in any orientation.

controlled curve (c-curve): A curved line generated from a vector line (polyline), which functions as a control line. When stored internally, c-curves carry their controls and can be subsequently edited and changed. In form-Z, c-curves can be generated using any of the available spline formulas (Bezier, B-splines, NURBS, etc.).

controlled meshes (c-meshes): Rectangular meshes generated from control lines, to which spline formulas are applied first. These are controlled objects that are stored with their control parameters and can subsequently be edited and changed.

controlled objects (c-objects): Objects that are stored with the control parameters used for their generation. Such objects can subsequently be edited and changed, while they remain c-objects. See also *parametric*.

controlled rounding (c-rounding): A variation of the rounding operation, where the rounding parameters are stored with the rounded object and can later be edited and changed through a preview window.

convergence: An operation, (also called extrusion to point) by which, given a shape, its centroid is calculated and placed at a distance from the original shape, called the apex. Then triangular side faces are generated that connect the apex with each segment of the original shape. A convergence derived from a rectangle produces a pyramid; a convergence derived from a circle produces a cone.

converting: Changing one type of object and/or model to another, such as changing a smooth to a facetted model, changing a revolved to a plain object, etc. The operation is executed with the Convert tool.

Coons patch: See *patches*.

covering: Fitting a surface to a closed wire, which is typically non-planar. The operation is executed with the Cover tool.

cubic curve: See *quick cubic*.

cubic falloff: See *falloff*.

cubic texture mapping: See *texture mapping types*.

current time: A point along the time line of an animation that corresponds to the current moment of the animation sequence.

cursor: An icon displayed on the screen at the position of the mouse. Cursor icons vary, depending on the active preferences and the operation being executed. See also *extended cursor*.

customizing: The act of setting the features of a program to a user's personal preferences.

custom heights: Height values defined by a user and entered in the Heights menu from where they can subsequently be selected and used to execute operations. See also *heights*. 
custom light: One of the light types available in form•Z RenderZone Plus. It is similar to a point light, except that the distribution of the light intensity in a given direction is defined by a distribution curve. These curves are usually supplied by light fixture manufacturers in the form of a photometric data file (IES, CIE or CIBSE format) and indicate a particular distribution pattern for a specific light fixture.

cutting plane: The plane used by the Section tool to cut an object and derive a 2D or 3D section. See also section.

cylindrical texture mapping: See texture mapping types.

decal: A texture map that is superimposed on plain color or another texture. It typically has a transparent background and it appears as a label or graffiti when rendered. Available only in form•Z RenderZone Plus, where up to a maximum of 32 decals per object can be placed. Decals are defined through surface styles and can overlap. See also surface styles.

defaults: The values assigned by the system at start up to parameters that affect the execution of operations, as well as other settings of the system. In form•Z, all these settings can be subsequently changed by a user, who can also save his/her own defaults as preferences.

deform: A patterned transformation of the geometry (points) of a meshed object. Typical deformations include twists, bends, tappers, bulges, etc.

delete: A form•Z tool that can be used to eliminate from memory complete objects/elements, or parts of objects/elements. See also deletions.

deletions: Eliminating geometric or topological entities, such as points, segments, outlines, faces, or holes, from the structure of an object. Complete objects or elements can also be deleted using the Delete tool or an equivalent key (the delete key on the Macintosh or backspace on Windows).

depth blur: An artifact in optical photography, that causes areas in a photo that are in front of and behind the focal point of the lens to appear blurred. The effect can be artificially added to a RenderZone rendering through a post process effect.

depth cue: One of the depth effects available in form•Z RenderZone Plus which renders objects closer to the viewer crisper and brighter and objects further back phased out.

depth effects: Also called foreground effects, they are rendering effects that highlight the different distances from the viewer at which the objects in a modeling scene are positioned. Depth effects available in form•Z RenderZone Plus are depth cue, fog, snow, and rain. See respective entries.

derivative objects: Objects that are derived from previously created objects or their parts.

deselecting: The opposite of selecting. That is, cancelling the selection of an entity.

developable surface: A surface that is curved in one direction and can be unfolded.

diamond: A 4-sided polygon.

diamond marker: A little diamond icon used in palettes to represent the visibility status of entities.

difference: One of the Boolean operations. When applied to solids, it creates an object from the volume which is in one of the objects but not in the other. That is, it cuts out the volume at which the two objects overlap. When applied to 2D shapes, the results are analogous.

diffuse reflection: A reflection parameter available in form•Z RenderZone Plus. It determines the percentage of light that illuminates a surface, which is reflected uniformly in all directions.

digitizer: A device that allows graphic input by tracing shapes through a stylus or similar devices. The digitizer can be 2D or 3D.
**dimension**: A composite drafting element that displays the distances between selected points or other measurements. It consists of a *dimension line*, *witness lines*, *terminators*, and *text* (see respective entries). In *form•Z* dimensions may or may not be *associative* (see entry) and can be one of four types: *linear*, *angular*, *radius/diameter*, or *arc dimensions*. See respective entries.

**dimension line**: A line drawn relative to a pair of points whose distance is measured by a dimension element. It is a straight line for the *linear* and *radius/diameter dimensions* or it takes an arc shape for the *angular* and *arc dimensions* (see respective entries). It may be a continues line or broken into two parts, depending on the position of the text.

**dimple**: A pattern consisting of a grid of spheres protruding from a surface. It is generated by one of the procedural bump textures available in *form•Z RenderZone Plus*.

**DirectDraw**: A Microsoft API that works with video cards and drivers to deliver accelerated 2D drawing of images on the screen for real-time animation playback. It is a component of the Windows operating system.

**direction**: The order in which sequences of connected points (as in faces and vector lines) are traced. Also a *form•Z* tool that can be used to manipulate the directions of objects.

**direct illumination**: Light applied to a surface during rendering, that travels in a straight line from the light source to a surface point. Direct illumination can be calculated reasonably fast and produces crisp shadows. All fast rendering modes employ direct illumination at the expense of lower lighting realism. High end rendering modes also use *indirect illumination* for increased realism.

**directional light**: See *distant light*.

**directional snaps**: A set of window tools that turn on or off snapping to specific orientations, such as horizontal, vertical, or a particular slope. See also *snapping*.

**dismantle group**: One of the *form•Z* grouping operations available in modeling. For the selected group, it destroys all the grouping links, at all levels of grouping.

**displacement**: A transformation applied to the geometry of a meshed surface, using a gray scale image that is mapped onto the surface. The intensity of the displacement is determined by the level (darkness) of the gray shade at different portions of the image.

**display resolution**: The density of an image when displayed on a computer screen or printed on paper. It manifests itself in a different fashion for different types of displays. For shaded renderings, it is measured by the number of pixels per unit length. For smooth objects, when they are displayed by approximating their surfaces with facets and or iso lines, the term refers to the density of the facets and the iso lines.

**display scale**: The size, measured relative to the original size, in which 3D views and 2D projection views in modeling, or 2D drawings in drafting are depicted on the screen or paper.

**distant light**: Sometimes also called directional light, it emits parallel rays, as if the light source is positioned far away. A special distant light is the *sun* (see entry), which is the only light affecting Quick Paint and Surface Render.

**distribute**: An operation that rearranges a set of objects/elements at equal distances from each other, where distances may be measured from their maxima, minima, or middle points. This operation is typically combined with the *align* operation (see respective entry).

**dodecahedron**: A symmetrical spherical object and one of the Platonic solids. It consists of twelve equal pentagons. See also *Platonic solids*.

**dome lights**: Light types, that emit light in a non uniform direction globally throughout a scene. The *environment* light type and the *atmospheric light* option of the distant light are dome lights available in *form•Z RenderZone Plus*. They work especially well in a final gather based rendering to create realistic illumination.
**doubly curved**: A surface that is curved in two directions, which makes it impossible to unfold it as a flat surface.

**draft angle**: A slight inclination of the sides of a solid in order to facilitate removing it from its cast, when the model is used for prototyping.

**drafting space**: One of the two types of spaces available in drafting. It is the space where drafting drawings can be created. The other type is **layout space** (see respective entry).

**drawing scale**: See **display scale**.

**drop controls**: An operation by which the control parameters of a parametric or controlled object are eliminated, which transforms the object into a plain object.

**duplicate**: A menu item and operation that makes a copy of a selected object or element, possibly slightly disposed from the original position of the entity.

**environment cube**: An imaginary cube on which one or more images are mapped and are then reflected on reflective surfaces. Both procedural and precaptured textures can be used with the environment cube. Available in form•Z RenderZone Plus.

**environment light**: One of the light types available in form•Z RenderZone Plus. It emits light from an image map that is wrapped around the scene. The light intensity is determined by the color and brightness of the pixels in the map. HDRI (high definition range image) maps may be used to simulate real world illumination.

**environment map**: An image mapped on an imaginary environment bounding volume, which can be a cube or a sphere. This image is then reflected on the surfaces of reflective objects. Only available in form•Z RenderZone Plus.

**environment sphere**: An imaginary sphere that surrounds a modeling scene, on which images are mapped so that they may be subsequently reflected on reflective surfaces. Both procedural and precaptured textures can be mapped on the environment sphere. Available in form•Z RenderZone Plus.

**explode**: form•Z tools in both modeling and drafting used to decompose a symbol to the entities it was constructed from. In drafting, it is also used to transform elements from one level to another, such as transforming a rectangle to a plain polyline, or a dimension element to its component parts.

**export**: See **import/export**.

**exposure correction**: A method in form•Z RenderZone Plus to correct the brightness and contrast in a RenderZone rendering. It is executed in a post process, which does not require for the rendering to be recalculated.

**extend**: A form•Z modeling tool used to extend a segment to the point where it intersects a plane in 3D space. A form•Z drafting tool used to extend a segment to its point of intersection with another segment.

**extended cursor**: A cursor whose two or three branches, corresponding to the coordinate axes of 3D or 2D views, extend possibly to the borders of the screen. In form•Z, it can be optionally turned on and used in the place of the regular cursor.
extract from group: One of the form•Z grouping operations available in modeling. It removes an object from the group where it was previously placed.

extracting: Retrieving the control lines/shapes of parametric objects and making independent objects from them. The operation is executed with the Extract tool.

extrusion: It is the operation by which, given a 2D shape, a copy is made at a certain distance from the original shape and in a position parallel to the original shape or to the reference plane, and then faces (called side faces) are generated between pairs of corresponding segments. This is more specifically called parallel extrusion to distinguish it from the extrusion to point or convergence (see entry). The term is also used to describe a solid derived by applying an extrusion operation. The latter is also called extruded object or extruded solid.

extrusion to point: See convergence.

eye path: The path the camera’s location follows during an animation.

face: This is the entity that is used to describe the surfaces of facetted objects. It consists of one or more outlines. One of these outlines always has a positive direction and delineates the outer perimeter of the face. All additional outlines have negative directions, are contained within the outer outline, and represent holes. The face is a modeling topological level positioned between the outline and the object.

facet: See face.

facetted: Also known as polygonal or boundary representation, it is a method by which surfaces and solid volumes are represented by a number of bounded flat surfaces called facets or faces (see respective entry). This representation approximates curved surfaces with flat facets, and is resolution sensitive.

facett model: A type of model available in form•Z, by which an object is represented by a closed set of bounded flat surfaces.

fallop: A parameter of lights that determines how the intensity of a light decreases as the distance of the light source increases. It affects all light types except distant. In form•Z RenderZone Plus three options are available: Constant maintains a constant intensity regardless of how far the light source is, which essentially produces no falloff effect. Linear decreases the intensity linearly with distance; that is if a surface is twice as far from the light source as another surface, it receives half the light. Square decreases the intensity exponentially with distance; that is if a surface is twice as far from the light source as another surface, it receives one quarter of the light. This parameter is also used in two transparency shaders that produce glowing effects, namely center-edge, and neon (see respective entries), where an additional option is available. Cubic decreases the intensity by a cubic exponent, as the distance of a surface from the light increases.

filleting: See fit fillet.

final gather: A global illumination technique available in form•Z RenderZone Plus, that computes a one bounce light reflection in a scene prior to a rendering. The reflected light is stored as sample points in a cache. The sample points are used to gather light at rendering time to provide more accurate illumination from reflected light. Final gather may also use a radiosity solution to provide illumination from multiple light bounces.

first point: The point at which the tracing of the connected sequences of points begins. The positions of first points are significant when lining up vector lines, such as when generating c-meshes.

fit fillet: The operation that rounds the edges of solids or the points of vector lines (polylines). For solids see rounding. For vector lines, it is one of the line editing operations available in both modeling and drafting. It replaces the vertex of a line with an arc, whose size is set by the user. A variation of this operation is beveling (see respective entry).
flat rendering: A rendering method that renders each face of an object opaque and in a uniform shade. That is, it applies no rendering effects, which makes it a fast rendering method, but also one that produces lower quality.

flat texture mapping: See texture mapping types.

flat textures: See procedural textures.

fly by: See walkthrough.

fog: One of the depth effects available in form•Z RenderZone Plus that simulates the effect of natural fog by reducing the crispness and brightness of objects in regions where fog is present.

foreground: See depth effects.

frame: A single image of an animation sequence. A portion of a form•Z window that is divided from the other portion of the window and shows the same project from a different view and scale. See window frames.

frames per second: The rate at which the frames of an animation are drawn on the screen. The faster the rate, the smoother the animation appears. 24 f.p.s. is common for film applications and 30 f.p.s is common for video applications.

Fresnel filtering: This is a filtering algorithm that models specular reflections in a physically accurate manner. It specifies how the specularly reflected fraction of light varies with the angle of incidence on a given surface of a given material. This fraction attains a maximum value at surface grazing incidence and it generally falls as the angle of incidence approaches the surface normal. In form•Z RenderZone Plus this filtering algorithm is used with four reflection shaders, namely accurate metal, mirror, accurate and simple glass.

global illumination: (or GI for short) A rendering technique, that creates images at a very high level of realism. It takes into account the accurate distribution of light in an environment by considering all light that exists globally in a scene, including indirect illumination. It is a term that summarizes a number of individual techniques, such as final gather, ambient occlusion and radiosity.

glow: A reflection parameter that determines the degree of self illumination of a surface. Glowing surfaces are not actual light sources but are simply rendered as if they were illuminated from behind. Available in form•Z RenderZone Plus.

graduated background: A background color which is derived from two colors phasing into each other. Only available in form•Z RenderZone Plus.

graphic/keyed heights: A mode selected in the Heights menu by which height values required by the extrusion and other operations are entered interactively through graphic input, rather than being selected from the height values contained in the menu.

grid: See reference grid.
**grid snap**: One of the window tools, which is a switch used to turn on or off grid snapping. When on, the mouse cursor can only be positioned at “rounded” locations determined by a user selected modular increment. See also **snapping**.

**group**: This is a collection of objects that are linked together. Groups can contain an object or other groups and can be defined at different hierarchical levels. Group is the highest topological level in modeling. This is also a modeling tool used to group objects.

**highlight**: A method by which selected entities are shown in a distinctly identifiable manner. In **form•Z**, picked objects and elements are highlighted with color, which is red by default, but can be changed to any user selected color.

**hither and yon planes**: Planes perpendicular to the line of sight which may be used to clip a modeling scene when views are rendered. Their position can be set numerically or through the **cone of vision** environment (see respective entry). Hither is the plane closest to and yon the plane farthest from the viewer.

**hole**: A piece missing from a solid or a 2D shape, and the method for representing it. On solids, holes are represented as all the other parts of the objects, through faces that bound their surface. On 2D shapes, holes are represented by negative outlines positioned inside the outer boundary of the shape. In **form•Z**, holes can be produced by a variety of operations, such as the Boolean **difference** (see respective entry). They can also be created directly using one of two insertion operations, called insert hole and insert opening. With respect to the end result, the term “opening” is synonymous to “hole” and is only used to distinguish between two operations that have different requirements and are executed differently.

**horizontal dimensions**: See **linear dimensions**.

**hybrid radiosity**: A method which allows high quality images to be rendered from a **radiosity solution** (see respective entry), which incorporates secondary illumination generated by the radiosity process and direct illumination generated at rendering time. Traditional radiosity based renderings incorporate both, direct and secondary illumination from the radiosity solution. The advantage of the hybrid method is that radiosity meshing can be kept low since there is no need to create fine meshing at shadow boundaries for high quality images. Available in **form•Z RenderZone Plus**.

**hexahedron**: Also known as a cube. One of the Platonic solids, bounded by six equal squares. See also **Platonic solids**.

**hidden line**: A modeling plotting method by which only the visible edges of objects are displayed. Edges on the back sides of objects or edges that are hidden by other objects are not shown.
triangles. See also *Platonic solids*.

**ill formed**: The opposite of well formed and synonymous to *incompletely formed*. See *well formed*.

**illuminance**: A quantity which describes the luminous flux (lumens) that hits a surface in a unit square area. Its units are *lux*, which is defined as one lumen per square meter, or *footcandle*, which is defined as one lumen per square foot. Illuminance is a useful measure for determining the light intensity on a unit area of a surface. The illuminance of a scene can be measured in *form-Z RenderZone Plus* through an analytical color coded rendering. See also *luminance*.

**illumination analysis**: A rendering option in *form-Z RenderZone Plus*, which color codes the illuminance or luminance of a scene. This helps visualize properly lit surfaces for architectural lighting layouts.

**image background**: A precaptured image used to render the background of a modeling scene. Available in *form-Z RenderZone Plus*.

**image element**: A special drafting element used to place an image into drafting. It is created by either pasting it from the Clipboard or by importing it through one of the image format files available in *form-Z*.

**image exposure**: A measure of how the light reflected off a surface is mapped into the visual color range of a display medium, such as photographic film or a computer screen. Overexposure occurs, if the light intensity in a scene exceeds the display range of the display medium. Underexposed images are created if the light intensity of a scene occupies only the lower portion of the display range. Both effects can be corrected in *form-Z RenderZone Plus* by applying a mapping curve, which compresses or stretches the intensity into the computer screen’s range.

**Imager**: A *form-Z* batch utility that can render any number of images in any of the rendering modes available. It can run from within *form-Z* or independently.

**image textures**: See *precaptured textures*.

**import/export**: The process of opening files saved in other applications or saving project files with the intention of opening them in another application. The process typically involves the translation of the *form-Z* data into another file format, when exporting, or the reverse, when importing.

**incompletely formed**: The opposite of well formed and synonymous with ill formed. See *well formed*.

**indirect illumination**: Light applied to a surface during rendering that bounces off other surfaces after it is emitted from a light source. Accurate indirect illumination is expensive to compute. See also *global illumination*.

**insert**: Six *form-Z* modeling tools that can be used to insert points, segments, outlines, faces/volumes, holes, and openings. See also *insertions*. A *form-Z* drafting tool used to insert points into polylines.

**insertions**: Adding new topological entities to the topological structure of an object. In *form-Z* you can insert points, segments, outlines, faces, and holes (or openings).

**instance**: See *symbol instance*.

**intersection**: One of the Boolean operations. When applied to solids, it derives a new solid that consists of only the volume which is in both of the original objects. That is, the volume at which the original objects overlap. When applied to 2D shapes, the results are analogous.

**iso line**: Lines or wires used to approximate the surface of smooth models of objects, mostly for display purposes.

**isometric**: One of the 3D view types available in *form-Z*. It is an artificial view that preserves the true dimensions of 3D objects. It is derived by plotting the X and Y coordinates at constant inclinations relative to the horizontal orientation, and the Z coordinates parallel to the vertical orientation.

**iteration**: A single pass during the processing of a radiosity solution, during which the light emitted by
a light source, or a reflective surface, is distributed to all other surfaces in the scene. A complete radiosity solution is created by continuous iterations, until a satisfactory distribution of light in a scene is achieved.

**J**

**join**: A form-Z modeling operation by which two or more separate objects become one. A form-Z line editing operation, available in both modeling and drafting, by which two open vector lines (or polylines) that have coincident end points become one continuous vector line.

**K**

**keyframe**: A location in an animation that defines critical camera parameters and timing information. The animation will always pass through the keyframe using its parameters. Frames between the keyframes are generated by interpolating between adjacent keyframes.

**knurl**: A pattern consisting of a grid of pyramidal shapes extending from a surface. It is generated by one of the procedural bump textures available in form-Z RenderZone Plus.

**L**

**landform model**: See terrain model.

**lathed object**: See revolved object.

**layer**: A method by which objects and elements can be organized in different groups called layers.

**layout space**: One of the two types of spaces available in drafting. It is the space where drawings can be composited. The other type is drafting space (see respective entry).

**leader line**: A drafting element used to attach notes to or to relate different drafting elements. It may or may not be associated.

**lens flares**: An artifact in optical photography, that causes light rings to appear in a photo. They are caused by the optics of the camera lens. Lens flares can be artificially added to a RenderZone rendering through a post process effect.

**light**: By rendering terms, a complement to color or surface style whose intensity, brightness, and shade it affects. It simulates real world light. In form-Z RenderZone Plus eight types of light are available: ambient, distant, cone, point, area, projector, environment, and custom. Only the two first types are available in the regular version of form-Z. See respective entries for details on light types. Quick Paint and Surface Render use only ambient and a single distant light called sun. Shaded Render and RenderZone accept multiple lights of different types.

**light attributes**: They are: visibility, whether the light is shown with wire frame displays; shining, whether the light actually shines or is turned off; shadow casting, whether a light casts shadows or not.

**light direction**: The orientation towards which light is emitted. Only distant and cone lights have directions, represented by a single arrow or a cone, respectively, when a wire frame is displayed and their visibility attribute is on.

**light source**: A point in 3D space that emits light according to the light type it represents.

**light types**: In the regular version of form-Z two types are available: ambient and distant. form-Z RenderZone Plus offers cone, point, area, projection, environment, and custom lights. See respective entries.

**linear dimensions**: These are the horizontal, vertical, and parallel dimensions. The first two are generated in the respective orientation regardless of the slope between the two points whose distance they measure. The latter is generated at a slope parallel to the slope of the points they measure.

**linear falloff**: See falloff.
line light: One of the light types available in form•Z RenderZone Plus. It is defined by the geometry of a line object, such as an open or closed wire. The light is emitted radially along the lines. It can be used to simulate linear light fixtures, such as neon tubes.

line of sight: The line defined by the point of view and center of interest. It represents the direction from which a scene is viewed.

line style: A user defined line type, that may consist of a variety of dash line combinations, used for drawing the boundaries of drafting elements. Line styles are displayed and selected from the Line Styles palette.

line weight: A user defined weight for lines used to delineate drafting elements. Line weights are displayed and selected from the Line Weights palette.

local axes: See object axes.

local coordinate system: A coordinate system that is local to an object and is relative to an origin that follows the object when it moves. The local origin of an object has a position in the world space, which allows coordinate values to be translated from local to global and vice versa.

lock marker: A little lock icon used in palettes to represent the selectability status of entities. Also used in some dialogs to lock related parameters to the same value.

loop: Relative to animation, it is a mode of running an animation. The animation starts over when it reaches its end.

luminance: A quantity which describes how much of the light energy that hits a surface is reflected back into the environment at a given direction per square area. Its units are measured in candelas per square foot or square meter. This is a useful measure for determining how bright a surface appears to an observer. The luminance of a scene can be measured in form•Z RenderZone Plus through an analytical color coded rendering. See also illuminance.

luminous flux: A quantity that defines the energy emitted by a light source in a unit time interval. It is measured in lumens. This measurement is used frequently by manufacturers of light fixtures and light bulbs as an indicator of the output power of a light. Unlike the energy specification of light sources in watts, lumens is a direct indicator of the brightness of a light source. The light intensity of all light types in form•Z RenderZone Plus, except distant lights, can be defined in lumens. See also luminous intensity.

luminous intensity: A quantity that describes how much light energy is emitted by a light source in a unit time interval in a specific direction. It is measured in candelas. While luminous flux indicates the overall light output, luminous intensity measures how strong a light shines in a certain direction. This measurement also takes into account the volumetric shape of the light. In form•Z RenderZone Plus, the intensity of cone, point, projector and custom lights can be defined in candelas. See also luminous flux.

M

macro transformation: A sequence of distinct geometric transformations executed as a single transformation. In form•Z there are both graphic and numeric methods for defining macros, which are then assigned to a tool with which they are executed.

magnifying glass: A zoom tool. An icon commonly used to represent the zoom operation.

match perspective: The process of extracting the perspective viewing parameters from a background image.

measure: A tool that measures lengths and distances.

memory box: An information box on the lower margin of form•Z windows that displays how much memory is currently free.

mesh: A pattern of cells superimposed on a surface or a complete object. A mesh pattern typically consists of rectangular cells, but it can also be a triangular pattern, or any irregular pattern. In form•Z, plain
meshes can be generated on any existing object or c-meshes can be generated from control lines. See also **controlled meshes**.

**meshed terrain model**: A terrain model whose land form depicting surface is a rectangular mesh of adjustable density, whose points are positioned relative to the elevations of the contour lines from which the terrain model is derived. See also **terrain model**.

**meshmove**: One of the deformation operations available in **form•Z**. An area of a mesh is smoothly moved according to the form of a profile shape. See also **profile**.

**metaball**: One type of the **metaformz** (see respective entry). A spherical surface, that has the property of blending smoothly with other surfaces of the same type. Its blending function is controlled by its weight parameter, which may have an additive or subtractive effect.

**metaform** (plural: **metaformz**): A 3D shape defined by an implicit surface that has the property of smoothly blending with other metaformz. It includes the metaballs, which have a spherical shape, but can also take a variety of other shapes.

**mirror**: The geometric transformation by which an object or an element is reflected relative to a line of reflection, a point, or a plane. • Reflection shader in **form•Z RenderZone Plus**.

**model**: As used in **form•Z**, it is a computer representation that corresponds to a physical entity.

**model type**: As used in **form•Z**, it is used to distinguish between different methods for representing objects. The main types are faceted and smooth objects.

**modifier**: A **form•Z** tool that does not by itself produce a result, but determines the result of other operations. In **form•Z** there are four groups of modifiers: object type, insertions, topological levels, and self/copy modifiers. The modifier tool icons appear in teal or magenta colors. For a different type of a **form•Z** tool, see **operator**.

**morphing**: The form of one object changes to the form of another object. As implemented in **form•Z**, in-between another forms can also be generated.

**move**: One of the geometric transformations. Synonymous with translate. A **form•Z** tool that can be used to place an object, an element, or their parts to a different position. When combined with one of the Copy modifiers, one or more copies are placed to new positions.

**multi-copy**: A **form•Z** tool (modifier) that makes a preset number of transformed copies of objects or elements, when a geometric transformation is applied.

**navigate**: The process of setting up a view or a sequence of views through a modeling scene.

**neon**: A type of transparency available in **form•Z RenderZone Plus** that produces a glowing effect simulating neon lights.

**network rendering**: Renderings distributed over a number of computers connected through a network.

**noise**: A random disturbance that is applied to patterns of certain procedural textures in **form•Z RenderZone Plus**. Three different algorithms can be used, giving increasingly better results at the cost of rendering speed.

**NTSC** (National Television System Committee): The broadcast video standard for many countries on the American continent as well as many Asian countries, including Japan.

**NURBS**: The term stands for Non Uniform Rational B-Spline. It is the most general type of a spline whose control parameters include knots and weights. See also **controlled curves**.

**nurbz**: A surface or solid generated from control lines, by applying NURBS interpolations.
object: A term used in varied ways by different applications. It is typically used to describe what a program constructs. In form•Z, object is a physical entity constructed by the modeling module of the program. Entities constructed by the drafting module are called elements. Object is also a modeling topological level positioned between the face and the group.

object axes: X, Y, and Z (Cartesian) axes that each object has and are distinct from the global axes.

object movie: A type of a QuickTime movie, where the virtual views are from the outside towards the center of a scene or object. The other type is panoramic movie (see respective entry).

object of revolution: Derivative object that is created from a profile called source shape, that is revolved (rotated) about an axis of revolution. In form•Z, the axis of revolution may be any of the Cartesian axes or a line drawn somewhere, or a segment of a previously created object. As a source is revolved, it may retain the same position relative to the length of the axis of revolution or it may be incrementally moving, producing lathed objects and helixes, respectively.

object snaps: A set of window tools that turn on or off snapping to parts of objects or elements. See also snapping.

object type: As used in form•Z, it distinguishes between different kinds of objects, such as cube, sphere, revolved object, sweep, nurbz, metaformz, etc.

oblique: One of the 3D view types available in form•Z. It is an artificial view that preserves the true dimensions of 3D objects. It is derived by plotting the X direction parallel to the horizontal orientation, the Y directions at a constant inclination relative to the horizontal orientation, and the Z direction parallel to the vertical orientation.

octahedron: A symmetrical spherical object and one of the Platonic solids. It consists of eight equal triangles. See also Platonic solids.

one copy: A form•Z tool (modifier) that causes the geometric transformations to make one copy of the object/element and apply the transformation to it.

one-sided surface: A surface object or mesh that has one positive and one negative side, in contrast to the two-sided surface or surface solid (see respective entry).

opening: See hole.

operator: A form•Z tool that executes an operation and always produces a visible result. The icons of these tools are shown in black and white colors. For a different type of form•Z tool, see modifier.

origin: The point where the Cartesian axes intersect and whose coordinate is (0,0,0) in 3D space or (0,0) in 2D space. See also axes of Cartesian space. Objects and symbols have their own local origin. With symbols it is used as a reference point when placing them in the world space. form•Z RenderZone Plus texture maps and decals also have their own local origin and coordinate system, used when they are mapped.

orthogonal coordinate system: See Cartesian coordinate system.

outline: A closed sequence of segments used to describe (delineate) a shape. One or more outlines delineate a face. The outline is a modeling topological level positioned between the segment and the face. Even though a true outline is a closed shape, the term is also used with open lines. Outlines are selected by two points on them or one point inside the area they bound. Outlines have a direction. In form•Z, the clockwise direction is considered positive and the counter clockwise direction is negative.

P

PAL (Phase Alternating Line): The broadcast video standard for most European countries except for France (where SECAM is the standard).

palettes: Small floating windows that support a variety of features, such as displaying coordinates and prompts, layers, names of objects and groups, colors or surface
styles, etc. The term is also used in the content of the modeling and drafting torn off palettes.

**palindrome:** A mode of playing an animation. The animation is played backwards after it reaches its end.

**pan:** Moving the image displayed on the screen in any direction determined by mouse input. It resembles window scrolling, except that the latter is restricted to horizontal and vertical moves only.

**pane:** An element of the layout space used in drafting for composing drawings. It is a window into the drafting space. Panes of the same layout space may be in different scales and display different details of the drafting space.

**panoramic:** A special composite rendering derived by incrementally rotating a camera about a complete circle (360°). It can be a still image or the basis for a **panoramic movie** (see respective entry).

**panoramic movie:** A type of a QuickTime movie, where the virtual views are from the center of a scene. The other type is **object movie** (see respective entry).

**parallel dimensions:** See **linear dimensions**.

**parallel element:** A drafting element derived from another polyline by generating parallel segments for (or applying a parallel offset to) each segment of the original polyline. The distance of the parallel lines may be determined interactively or by numeric input.

**parallel objects:** A derivative object created from another object by generating surfaces parallel to the surfaces of the original object. These surfaces can be generated on both sides (double parallel) or only on one side (single parallel) of the original object. Parallel objects can also be generated from open vector lines.

**parallel offset:** A drafting tool that creates **parallel elements** (see respective entry).

**parametric:** An entity that can be generated by a set of parameters and can thus be saved by storing its parameters. This makes it easy to edit and change after its initial generation. In geometric modeling parametrically represented entities typically need to be evaluated and approximated by facets or polygons in order to be viewed.

**parametric texture mapping:** See **texture map types**.

**paste image:** A **form•Z** menu item and an operation by which an image previously captured and stored in the Clipboard is placed into a drafting window to create an image element.

**patches:** Four or three sided parametric tiles, which can be combined side-by-side to parametrically define open or closed surfaces. Two types are available: **bi-cubic Bezier** and **Coons patches**.

**patterned polygons:** Polygonal shapes that end in a variety of gear or star like “teeth”.

**perpendicular switch:** One of the window tools that turns on or off the perpendicular lock. When on, movement is restricted to an orientation perpendicular to the **reference plane** (see respective entry).

**perspective:** One of the 3D view types available in **form•Z**. It is derived by applying a perspective transformation to the 3D coordinates of objects, resulting in images which simulate real world perspective illusions.

**pick:** The act of identifying an entity, so that an operation may be applied to it. Synonymous to select. Also, a tool in both modeling and drafting used to select entities. The Pick tool is also called the Arrow tool. See also **prepick** and **postpick**.

**pick parade:** A method by which overlapping entities can be selected one after the other. In **form•Z** this is done by using the shift key in combination with the Pick tool.

**photometric intensity:** Physically accurate light intensity defined in candelas or lumens and an optional **color temperature**. The light intensity of all light types in **form•Z RenderZone Plus**, except distant lights, can be defined in photometric intensity. See also **radiometric intensity**.
pixel ratio: The ratio of the width to the height of each pixel in a camera’s image.

place: A form•Z tool used to execute placements of shapes on vector lines. See also placement.

placed symbol: See symbol instance.

placement: The operation by which a shape is placed relative to a segment or a point of a vector line, called the placement line. Shapes are placed perpendicular to segments or along the bisectors of the angles at points.

plain color background: One of the background types available in form•Z RenderZone Plus, where the background is rendered with a uniform color, which can be any color selected from the Color Picker dialog.

plain object: An object that is not parametric.

planar face: See planarity.

planarity: Whether a surface (face or outline) is planar or not. It is planar when all its points lie on the same plane. A triangular surface is always planar. A surface delineated by more than three points may or may not be planar, but can be transformed to planar by triangulating it. A complete object is planar when all its faces are planar.

planar object: See planarity.

planar outline: See planarity.

planar surface: See planarity.

Platonic solids: Spherical and perfectly symmetrical solid objects, first identified by the classical philosopher Plato. They are all bound by equal polygons. They are: tetrahedron, hexahedron, octahedron, dodecahedron, and icosahedron (see respective entries).

point: Also called vertex, is a position in 3D or 2D space represented by its coordinates. A collection of points represents the geometry of an object or element. Properly connected sets of points completely describe an object or element. Point is the lowest topological level.

point cloud: A set of independent points, each of which has a position in 3D space defined by a triplet of coordinates (x, y, z). Point cloud data is typically produced by laser scanners.

point disturbance: A slight movement applied to the geometry of a group of points either randomly or according to a mathematical pattern. A typical pattern would be a linear or concentric wave.

point light: One of the light types available in form•Z RenderZone Plus. It emits rays from a given point outwards in all directions.

point marker: A tool that places or removes a mark to/from a point of a shape, to be used by the skin operation as a source or a path.

point object/element: An object (modeling) or element (drafting) that consists of a single point.

point sampling: A method for determining the color of a pixel on the screen when an image map, which has a resolution larger than the rendered image, is applied. It uses the value of the pixel in the image map that corresponds to the rendered pixel, and ignores the other pixels in the image. For another method see area sampling. While point sampling is faster, area sampling usually produces better quality images. Only available in form•Z RenderZone Plus.

polar angle: See polar coordinate system.

polar coordinate system: It expresses the position of a point in 2D space through two values: distance and angle. Distance is the distance of a point from another point, which is a constant origin, when absolute polar coordinates are used, or a previous point, when relative coordinates are used. Angle is the slope of the line connecting the origin (or previous point) with the coordinate point relative to the horizontal orientation. Polar coordinates are available in form•Z for input purposes. Internally coordinates are stored using the Cartesian coordinate system (see respective entry).

polar distance: See polar coordinate system.

polyarc: A vector line (polyline) approximation of an arc. See also arc.

polygon: A symmetrical closed shape that consists of equal sides. Typical polygons are the triangle
(3-sided), diamond (4-sided), hexagon (6-sided), octagon (8-sided), etc. In form•Z any n-sided polygon can be generated.

**polygonal**: See *facetted*.

**polyline**: See *vector line*.

**postpick**: A method for executing form•Z operations by which the entity or entities to which an operation is applied are picked directly with the tool that also executes the operation.

**post processing**: A method in form•Z RenderZone Plus, which applies effects to a RenderZone rendering, after it has been completed, without recalculating the rendering. The effects are applied based on a pixels color and depth. Lens flares, exposure correction and depth blur are post processing effects.

**precaptured textures**: Also called image textures, these are textures generated from bitmapped images saved in one of the formats accepted by form•Z. These images can be created from photographs, by a scanner, or by a rendering or painting program. These textures are always of the wrapped (flat) texture type. See also *texture types*. Textures are only available in form•Z RenderZone Plus.

**preferences**: Saving one’s own parameters to be used as defaults. See also *defaults*.

**prepick**: A method for executing form•Z operations by which any number of entities are selected first and then an operation is applied to all of them.

**preset heights**: Menu items in the Heights menu containing default height values, which can subsequently be changed by a user. See also *heights*.

**primary source**: A light source in radiosity, which actively emits light. Primary sources are all light sources, which are currently shining. See also *secondary source*.

**primitive**: A basic geometric shape that most commonly includes the cube, the cone, the cylinder, the sphere, and the torus.

**procedural backgrounds**: Procedural textures such as brick, checker, grid, marble, polka dots, and stripes, used to render backgrounds in form•Z RenderZone Plus.

**procedural textures**: Textures generated by an algorithm (rather than been derived from a previously captured image). In form•Z RenderZone Plus procedural textures can be one of two types: wrapped (or flat) and solid, each applied in a different manner. Wrapped textures are 2D textures and are mapped to the surfaces of objects in a manner similar to the image textures. Solid textures are 3D textures which are mapped onto the surface of objects by calculating the intersection of the 3D texture with the surface where they are mapped. Solid textures have the advantage of continuity over edges and require no special mapping procedures, as the wrapped textures do (see *texture mapping types*).

**profile**: A 2D shape that determines the form of a mesh movement. These shapes (which can be defined by the user) are stored and selected from the Profiles palette. The term is also used to describe shapes used as source shapes for objects of revolution, extrusions, sweeps, etc.

**projection objects**: Derivative objects generated from orthographic projections of objects, 3D views of objects, or unfolded patterns of objects.

**projector light**: One of the light types available in form•Z RenderZone Plus. It emits light from a given point in the direction defined by a pyramidal shape. The light is filtered through a texture map, whose image is projected onto the scene.

**quadratic curve**: See *quick quadratic*.

**quadratic subdivisions**: A method by which the faces of an object are subdivided into smaller four-sided faces, any number of times. It is used for increasing the representational resolution of an object and may be combined with curving the surfaces as they are subdivided.

**query**: A form•Z tool, available in both modeling and drafting, used to invoke information about objects/elements, their parts, and their attributes. It can also be used to change the parameters of controlled objects and to measure distances between points, segments, and surfaces.
**quick cubic:** A spline consisting of 3rd degree Bezier curves, created using a simplified (and thus faster) Bezier formula.

**quick paint:** A rendering method in modeling, which displays the surfaces of objects in an order according to their distance from the viewer, but does not decompose intersecting surfaces, concave surfaces, and surfaces with holes. This allows it to be fast but does not always produce correct renderings. It is intended for preview level renderings.

**quick quadratic:** A spline consisting of 2nd degree Bezier curves, created using a simplified (and thus faster) Bezier formula.

**QuickTime:** An Apple API and file format for storing audio and video data including animations. It is available for both Macintosh and Windows computers and supports a variety of file formats.

**R**

**radial bend:** One of the deformation operations that bends one or both ends of a meshed object following a circular shape.

**radial shear:** One of the deformation operations that repositions the middle region of a meshed object, in a manner that forms a circular shape with its ends.

**radiometric intensity:** Physically accurate light intensity defined in watts. The light intensity of all light types in RenderZone Plus can be defined in radiometric intensity. See also **photometric intensity**.

**radiosity:** A technique that simulates the transfer of light energy between surfaces in an accurate manner. The representation of this simulation is called a **radiosity solution** (see respective entry) and describes the distribution of light from all sources across all surfaces in a scene. It also accounts for light that bounces off surfaces and illuminates other surfaces, called secondary illumination. Such a radiosity solution can be used as the base for a rendering in form-Z RenderZone Plus, which will display a scene with accurate lighting information, replacing the uniformly applied ambient light of a rendering executed without the support of radiosity.

**radiosity mesh:** The geometry in a scene, subdivided into smaller mesh polygons, suitable for radiosity processing. The vertices of each mesh polygon capture the light intensity emitted from each light source. As a result, a denser radiosity mesh gives a better definition of the light distribution, but takes longer to compute. The radiosity mesh can also be generated in an adaptive manner, where more mesh polygons are generated at areas of high contrast. See also **adaptive meshing**.

**radiosity patch:** A part of the **radiosity mesh** (see respective entry), which emits light. Radiosity patches are a coarser subdivision of the geometry in a scene. Each patch may be further subdivided into additional mesh polygons, defining the radiosity mesh. The sum of all light intensity values of the mesh polygons of a patch define the light energy which is emitted from the patch in an **iteration** (see respective entry).

**radiosity solution:** The representation of the light distribution in a **radiosity mesh** (see respective entry). A radiosity solution contains a percentage of completion, which indicates, how much energy emitted by all active light source has been absorbed. For example, a 70% complete solution means, that 30% of the light in a scene is still unaccounted for and remains to be distributed. Accurate solutions take increasingly longer to generate, since the radiosity process is by nature exponential.

**radius/diameter dimensions:** They are used to measure the distance from the center (radius) or the diameter of arc elements. They can be generated in a horizontal, vertical, or arbitrary orientation.

**random disturbance:** One of the deformation operations that applies random displacements to the points of an object.

**raytracing:** A rendering algorithm, available only in **form-Z RenderZone Plus**, which determines the closest surface pixel by casting a ray from the viewpoint through each screen pixel and intersecting it with each face. The face whose intersection point is closest to the viewer is used to calculate the pixel color. Rays which hit reflective or transmissive sur-
faces are reflected or refracted according to the ray’s angle of incidence and the surface normal at the intersection point. This produces renderings with accurate reflections and transparencies. Raytracing typically requires more time than other rendering algorithms, and the time it requires increases significantly when a scene contains a large number of reflective or transmissive surfaces.

**reduce mesh**: Decreasing the resolution (number of facets) of a meshed surface, by taking into consideration factors such as angle of faces and size of edges.

**reference grid**: A graphic representation of a reference plane (see respective entry) in the 3D world space. The grid is shown as sets of parallel lines or dots at user defined increments and color. It indicates the origin, position, and orientation of its reference plane, but not the limits of the plane.

**reference plane**: A plane relative to which you draw. In **form•Z**, a reference plane can be any of the Cartesian planes (XY, YZ or ZX) or any arbitrarily positioned plane. A reference plane is visible when its reference grid is also displayed (see respective entry). The reference plane is generally independent from the 3D world space, but may also coincide with it. Likewise, the axes of the reference planes are independent from the world space axes.

**reflection**: See **mirror**.

**reflection maps**: Precaptured or procedural textures, which define the distribution of ambient reflection, diffuse reflection, specular reflection, glow, reflectivity and transmission across a surface. Available in **form•Z RenderZone Plus**.

**reflection shader**: It determines the reflective and transmissive properties of the surface to which it is applied. In **form•Z RenderZone Plus**, reflective shaders are based on material simulating procedures. When reflective, objects may reflect each other, or they may reflect an environment mapped onto a cube or sphere. See **environment map**. See also **shader**.

**refraction**: A shader parameter used by the accurate and simple glass reflection shaders to simulate the distortion occurring when an object enters a transparent mass. Rays which are transmitted through a surface are bent according to the index of refraction of that surface. This generates effects such as viewing through a magnifying glass or a stick appearing broken when it enters water. Each transparent material in the real world has a unique index of refraction. Available in **form•Z RenderZone Plus**.

**relative coordinates**: Coordinates measured from the previous point, rather than from a constant origin, which the **absolute coordinates** do (see respective entry). Relative coordinates can be used with both the Cartesian and polar coordinate systems (see entries).

**rendered environment**: An image produced from the non-reflective objects in a scene, which is then mapped onto an environment cube from which it is reflected back onto the reflective surfaces of the objects in the scene. See also **environment cube**.

**repeat copy**: A **form•Z** tool (modifier) that makes repeated copies, one for each time the mouse is clicked, when a geometric transformation is applied.

**resize**: See **scale**.

**revolved object**: One of the objects of revolution (see respective entry) generated by revolving a source about an axis without moving it along the axis. Also known as lathed object.

**revolved sphere**: See **sphere**.

**rotate**: The geometric transformation by which the points of an object or element change position by moving about a center, called the center of the rotation.

**round**: The tool in **form•Z** modeling with which edges and points of solid objects can be smoothly rounded or beveled. See **rounding**.

**rounding**: The operation by which points and/or edges of solid objects can be rounded by fitting an arc fillet to them. The operation can also be used to bevel edges and/or points. A special variation of the operation can be applied to closed sequences of segments, called stitches, where faces on each
side of the segments are smoothly blended using parametric meshes.

**ruled surface**: A surface generated by a straight line as it moves in space. Ruled surfaces have curvatures in one direction which allows them to be unfolded.

**rulers**: Bands with measurements that can optionally be displayed on both modeling and drafting windows. They can be displayed on any one or more of a window’s edges.

**S**

**s-loft**: Stands for *skinned loft*. This is an operation that combines procedures used in lofting and skinning. It uses sources and implicit or explicit paths to derive smooth objects. It is also capable of generating objects that branch.

**sampling**: A process which chooses points on the surface of a *radiosity patch* and emits lights to all radiosity mesh polygons. How many points are sampled determines the quality of the illumination in a *radiosity solution* (see respective entry).

**scale**: The geometric transformation by which the size of an object or element or their parts is changed. Synonymous with resize. In *form•Z* there are two scale operations. The independent scale applies scaling factors independently to each of the three Cartesian directions (X, Y, and Z). The uniform scale applies the same scaling factor to all three directions. The term scale is also used for the size at which 3D views and 2D drawings are presented on the screen or on paper, when they are printed. See also *display scale*.

**scene**: A collection of display characteristics including active objects, layers, and lights; views and rendering parameters; grid, reference plane, and underlay settings.

**screw**: A composite object of revolution generated by *form•Z* according to variable parameters. One of the variations is the bolt, which has a flat end.

**secondary source**: A light source in radiosity, which emits light after it absorbs light during a previous *iteration* (see respective entry). Secondary sources are all radiosity patches generated from the scene’s geometry. See also *primary source*.

**section**: A 2D shape derived by intersecting an object with a plane. In *form•Z* it is also the name of a tool that derives sections, which can be 2D or 3D sections. The latter are again derived by intersecting a solid with a plane, but the result of the operation are two or more solid pieces to which the original object is split by the cutting plane.

**segment**: Also called edge, is the link of two points. In simple terms, a segment is a line drawn between two points. The points, which are defined by coordinates, give the segment a position in 3D or 2D space. Segment is the second lowest topological level in both modeling and drafting.

**segment object/element**: An object (modeling) or element (drafting) that consists of a single segment.

**select**: See *pick*.

**self/copy modifiers**: *form•Z* tools that determine whether geometric transformations are applied to the original object/element or to copies of the object/element.

**separate**: A *form•Z* operation by which each volume in an object (if more than one) or certain parts of an object become an independent objects.

**shaded render**: The highest level of rendering available in the regular version of *form•Z* (and is also available in *form•Z RenderZone Plus*). It uses a z-buffer algorithm and can produce smoothly shaded surfaces, transparencies, antialiasing, and shadows from multiple lights.

**shader**: A rendering procedure which produces different rendering effects that can be superimposed and used to render surfaces of objects. In *form•Z RenderZone Plus*, surface styles are defined by up to four layers of shaders: *color, reflection, transparency*, and *bump shader*. See respective entries. Shaders are also used to calculate a background pattern, depth effects, and the pattern of the environment cube or sphere, shown in reflective surfaces.
shader antialiasing: An antialiasing method that applies super sampling selectively to surfaces, which are rendered by a particular shader. Available in form•Z RenderZone Plus.

shadow casting light: See light attributes.

shear: One of the deformation operations that applies an inclination to a meshed object.

shining light: See light attributes.

simple glow: The effect of light shining in a foggy environment simulated through a simple approximation. Glow can be generated for cone, point, and projector lights. Available only in form•Z RenderZone Plus. See also accurate glow.

singly curved: See developable surface.

skin: An operation that transforms a complete or partial wire outline of an object, consisting of source and path shapes, into a faceted object.

sky: A procedural texture available in form•Z RenderZone Plus for rendering backgrounds using two colors and producing a pattern that simulates clouds in the sky.

smooth line: A smoothly curved polyline derived using NURBS splines. See also spline.

smooth mesh: A process that increases the resolution (number of facets) of the surface of an object, while it may also smooth its sharp edges.

smooth model: A type of model available in form•Z, by which an object is represented by a closed set of bounded curved surfaces.

smooth shade: A rendering technique that paints surfaces by applying smooth gradations of color calculated from the angles at which light hits the surfaces.

snapping: Constraining the movement of the mouse to specific locations, which are determined by the selection of snapping switches available in the window palette of form•Z, in both modeling and drafting. The types of available snaps are grid snap, directional snaps, and object snaps. See respective entries.

snow: One of the depth effects available in form•Z RenderZone Plus that simulates the effect of falling snow flakes.

soccer ball: A spherical object bounded by sixteen equal hexagons and eight equal pentagons. Formally it is derived by intersecting a dodecahedron and an icosahedron.

solid: Any modeling object which is completely bounded by surfaces. A true solid contains a volume and its mass properties can be calculated.

solid textures: See procedural textures.

source shape: The shape used to generate another object such as in extrusions, objects of revolution, or sweeps.

specular reflection: A reflection parameter that determines how much of the light illuminating a surface is reflected at the incoming angle. Specular reflections typically create a “hot spot” on curved surfaces. Available in form•Z RenderZone Plus.

sphere: A symmetrical solid all points of which are at the same distance from its center. In form•Z, a sphere can be generated as an analytic primitive, as a revolved object, and as a geodesic representation that consists of triangles and is structurally similar to the Platonic icosahedron. All types of spheres are parametric objects from which faceted representations can also be generated.

spherical objects: Solid objects all points of which are at the same distance from their center. In form•Z, the spherical objects are the set of Platonic solids, the soccer ball, the revolved and geodesic sphere. See respective entries.

spherical texture mapping: See texture mapping types.

spiral: See helix.

spiral stair: A staircase of a spiral shape generated by form•Z as a composite object of revolution.

spline: A term used for parametric curves. It is a smoothly curved polyline that, in form•Z, can be
either drawn directly or derived from a control line. See also *controlled curve.*

**split:** A composite Boolean operation. When applied to solids, one object is split at the boundaries of the other. It can also be executed as a two-way split, where each object is split by the other. When applied to 2D shapes, the results are analogous. • A *form-Z* operation by which one surface is cut into two or more pieces by another surface.

**spot light:** See *cone light.*

**square falloff:** See *falloff.*

**stepped terrain model:** A terrain model where the elevation of each contour line is represented as a step in a manner analogous to cardboard models. See also *terrain model.*

**stick font:** A single line font for drafting that resides in *form-Z* and redraws fast. It can be used temporarily as a substitute for the real text, during the process of design and drafting.

**stitch:** Establishing the links of reversely coincident sequences of open (unattached) segments. This operation results in two surfaces being connected into one, along open segments that coincide. • A closed sequence of segments meeting at large angles.

**stream line:** A polyline where segments of a preset size are automatically generated by the program, as the mouse travels a certain distance. In *form-Z,* it is also the name of a tool with which stream lines are drawn.

**stretched ball:** A type of metaform.

**stretched ellipsoid:** A type of a metaform

**subtractive weight:** A metaform parameter that subtracts the volume of one metaform from another.

**sun:** A distant light which is the only light affecting the quick paint and surface render types of rendering. It is treated as all the other distant lights by shaded render, and RenderZone. See also *distant light.*

**super sampling:** A method that produces images with smooth edges, which can be used instead of antialiasing. The image is rendered in a resolution significantly higher, which is then reduced to the actual size.

**surface objects:** Any modeling object which is not a solid. It can be an open line, a closed shape (such as a polygon), which may or may not contain a hole. It can also be a collection of closed shapes attached at their edges to form a continuous non-planar surface. For an object to be a surface object some of its edges should be “free,” which is they should not be attached to other edges. If an object has no free edges, it is a solid or a surface solid. Surface objects are sometimes also called one-sided surfaces to distinguish them from the surface solids which are two-sided surfaces.

**surface render:** A rendering method in modeling that uses an algorithm known as “painter’s algorithm.” The surfaces of objects are first broken into convex polygons, which are ordered according to their distance from the viewer. More recently painted surfaces cover those previously painted, which produces a hidden surface effect. This algorithm preserves the original boundaries of the surfaces when the image is saved.

**surface solid:** This is a double sided surface which has the structural characteristics of a solid. That is, it has no free edges and is completely bounded but contains no volume. A simple example would be two equal squares, facing in opposite directions, and attached together as a single object.

**surface style:** Composite color which includes a variety of rendering effects such as texture maps, reflections, transparencies, and bumps. They are used in *form-Z RenderZone Plus* in the place of plain colors used in the regular version of *form-Z.*

**sweep:** The operation by which a profile (called source shape) is swept along a path line (called path shape) and then faces are generated between corresponding pairs of segments. The sweep operation is formally considered to be the single basic operation that can produce any form. For example, the extrusion is essentially a sweep and so is a revolved object. In *form-Z,* variations of the basic sweep, are the two source sweeps, the two path sweeps, and the boundary sweeps.
symbol: A method by which groups of objects or elements that need to be repeated in a project, or are frequently used in a variety of different projects, can be stored in libraries and then placed as many times as desirable, by referring to a single copy of their representation. In form•Z, both 3D symbols are available in modeling and 2D symbols in drafting. Note that what are called symbols in form•Z and many other applications are called blocks in AutoCAD and cells by Intergraph. In form•Z, 3D and 2D symbols can be stored in the same library. See also symbol definition and symbol instance.

symbol definition: An internal structure where the components and parameters of a symbol are stored. These are kept in symbol libraries, which are displayed in the Symbols palette from where they can be selected and placed as instances. See also respective entries.

symbol instance: A placed symbol. To place a symbol, it is selected from the Symbols palette, which displays symbol definitions. When the same symbol is placed in the same project more than once, a single internal representation of the symbol entity is referenced. This offers the advantage of saving memory and also makes it possible to change all the instances of the same symbol by simply changing the symbol definition which they reference. See also symbol definition and symbol library.

symbol library: An internal structure where symbol definitions are stored. In form•Z, symbol libraries are displayed in the Symbol palette from where they can be selected to be placed as instances. See also respective entry.

system colors: Colors reserved and used with interface features such as the reference grid, the windows background, the highlight, etc. They can all be changed through a dialog.

tangent curve: A spline consisting of 2nd degree Bezier curves, generated from a vector line which is tangent to the original control line. The tangent vector line is constructed by generating segments perpendicular to the bisectors of the points in the original vector line. An attribute of the tangent curves is that they pass through the control points. In form•Z tangent curves can be normal, shallow, or deep.

taper: One of the deformation operations that resizes one end of a meshed object, resulting in a trapezoid form.

terminators: Part of the dimension and leader line elements. They are markers, such as arrows, slashes, dots, or crosses that are drawn at the ends of the dimension lines or leader lines (see respective entries).

terrain model: A 3D model depicting land forms derived from a set of 2D contour lines and a site shape to whose boundary the model is trimmed. In form•Z four types of terrain models are available: meshed, triangulated mesh, stepped and triangulated contour models. See respective entries for more details.

tetrahedron: A symmetrical spherical object bounded by four equal triangles. See also Platonic solids.

text: A form•Z entity available in both modeling and drafting. In modeling text can be generated as either plain text, or as text object, which can be a solid, surface solid, or surface object. Different methods and parameters for placing text are available, including placing text along or between control lines, which are editable. Text in modeling is always created from outline fonts (TrueType or PostScript). In drafting, text may be placed as plain text or text element, and can also be derived from bitmapped fonts.

texture caching: A method that allows for textures to be written to disk and loaded when needed during a rendering. As a result of this method, renderings that otherwise could not be completed because of limited memory, can be completed. Available in form•Z RenderZone Plus.

texture group: A group of faces to which the same texture is applied. Available in form•Z RenderZone Plus.

texture mapping types: The method used to place a texture image on the surface of an object. They apply to wrapped textures only and not to solid
Textures (see respective item). Five types are available in form-Z RenderZone Plus. Flat is the simplest type, which places a texture map individually on each surface of an object by mapping it to the plane of that face. Cubic maps the texture on the faces of a boundary cube and then projects them to the object. Cylindrical maps the texture on the surfaces of a bounding cylinder, where a single image is mapped to the round surface of the cylinder, and then projects the textures to the object using projection lines that converge to the axis of the cylinder. Spherical maps the texture as a continuous image on the surface of a bounding sphere, from where it is projected to the surface of the object, with the projection rays converging to the center of the sphere. Parametric texture mapping can only be applied to c-meshes, nurbz, and primitives, where the texture follows the parametric direction of the object. UV mapping is the general purpose method that can freeze any other type of texture mapping to specific surface coordinates, relative to an object's vertices. It is the texture mapping method by which textures can be exported.

Texture maps: An image placed on one or more surfaces of an object. Available only in form-Z RenderZone Plus, they can be precaptured or procedural (see respective entries) and can be mapped in a variety of different ways. See texture mapping types.

Tile windows: Four small modeling windows, each occupying one quadrant of the screen. Three of the windows show orthographic views (top, front, and side) and the fourth a 3D view of the modeling view. Graphic input can occur in any of the four windows, while the others are also updated.

Time line: A ruler that measures the duration of an animation.

Tool: An icon with which an operation can be executed.

Tool bar: A linear arrangement of tool icons that can be oriented vertically (form-Z default) or horizontally.

Tool palette: A floating group of tools that can be torn off from the tool bar.

Topological levels: The different levels through which an object or an element is represented on the computer. In modeling they are the point (or vertex), segment (or edge), outline, face, object, and group. In drafting, they are point, segment, element, and compound.

topological transformations: The operations by which new topological entities (such as segments and faces) are inserted (added) to an object, or existing topological entities are deleted.

topology: The part of an object's or element's representation that describes how the points are connected.

Track: An animation term used for each parameter or characteristic of an entity that can be animated.

Transformation: The operation that changes the characteristics of an object. Specifically, there are two types of transformations in form-Z: geometric, which affect the geometry of objects and include the translation, rotation, scale, and mirror; and topological, which affect the topology of objects (how vertices are connected) and include the insertions and deletions.

Translation: By solid geometry terminology, synonymous to move (see respective entry), one of the geometric transformations. The term is also used to describe the process of saving form-Z models and elements into a format other than the native form-Z format.

Transparency shader: Available in form-Z RenderZone Plus, it calculates the transparency of surfaces, which may be defined through plain color, procedural textures, or precaptured images. See also shader.

Triangular subdivision: A method by which the faces of an object are subdivided into smaller triangular faces, any number of times. It is used for increasing the representational resolution of an object and may be combined with curving the surfaces as they are subdivided.

Triangulated contour model: A terrain model derived by inserting the contour lines at their proper elevation and then triangulating the surfaces between them. This terrain model preserves the positions of the contour lines. See also terrain model.
triangulated mesh terrain model: As the meshed terrain model (see respective entry) except that the rectangular cells of the mesh are triangulated. See also terrain model.

triangulation: The process by which non-triangular faces, including concave and faces with holes, are subdivided into triangular pieces.

time line: The linear display of the animation timing sequence shown in the animation palette. The time line shows the locations of keyframes and the current frame in time.

trim: A form-Z modeling operation by which one 3D surface is cut (trimmed) by another 3D surface. • A line editing operation, available in both drafting and modeling, by which segments of two vector lines are trimmed to their point of intersection.

twist: One of the deformation operations that applies a helical pattern to a meshed object.

two-sided surface: See surface solid.

underlay: A bitmapped image that can be placed in both modeling and drafting windows to be used as background or for tracing over it.

unfold: a form-Z operation which unfolds a given object. A flat projection of the faces of an object, where neighboring faces remain attached to the extent possible. See also unfolded projection.

unfolded projection: An object derived by unfolding an object and projecting the unfolded pattern onto the reference plane.

unghost: A tool, in both modeling and drafting, used to unghost ghosted objects/elements. (see respective entry).

ungroup: One of the form-Z grouping tools available in modeling. It removes an object from the group where it was previously placed, or it completely breaks a group.

union: One of the Boolean operations. When applied to solids, it derives a new solid object that consists of the volumes in either and both of the original objects. That is, it merges the volumes of the original objects into one. When applied to 2D shapes, the results are analogous.

uv coordinates: See texture mapping types.

vector line: A sequence of segments, or a sequence of connected points, that can be open or closed. Synonymous to polyline. In form-Z it is also the name of a tool used to draw vector lines.

velocity: The rate at which the camera's position and parameters are changed as it follows the animation path.

vertex: See point.

vertical dimensions: See linear dimensions.

viewer position: The location from which a modeling scene is viewed. Synonymous to point of view and camera, a term used by other applications. It can be set through the cone of vision environment (see entry) or through numeric input.

viewing spin: A viewing parameter corresponding to tilting one's head when viewing a scene. It can be set through the cone of vision environment (see respective entry).

visibility of light: See light attributes.

walkthrough: A type of animation that is generated by moving a camera through a static scene. • An interactive tool for navigating through a scene.

wave deformation: One of the deformation operations that transforms a mesh surface to a 3D wave pattern, which can be linear or concentric.

weight of metaformz: A parameter that determines the extent of blending and may cause the volume of one metaform to either be added to or subtracted from another metaform, depending on its sign.
**well formed**: This term is relative to solid objects and is almost synonymous to the term “solid.” When all the edges of all the faces of an object have a pair edge of another face and the object has no open edges, the object is solid and well formed. Otherwise the object is incompletely formed or ill formed as a solid.

**window frames**: A special modeling window set up that subdivides the window in a number of frames. These are four by default, but any number can be added and sized. Frames are views into the same modeling space and allow continuous drawing from one frame to another.

**wire**: As used in *form-Z*, it is an open or closed line (shape) that is not planar. The term takes a special meaning in “wireframe” (see respective entry).

**wire frame**: A modeling plotting method by which all the edges of objects are shown, as if the objects were constructed with wires, and regardless of whether they may be hidden by other objects.

**witness lines**: Part of the dimension elements, they are a pair of lines extended from the measured points to the location where the *dimension line* is drawn (see respective entry).

**working units**: Units of measurement that are used to describe the geometry of objects or elements. In *form-Z* both English and metric units can be used.

**world space**: The Cartesian coordinate system used in *form-Z* globally for the representation of its geometric entities. This space is 3 dimensional in modeling and 2 dimensional in drafting.

**wrapped textures**: See *procedural textures*.

---

**X**

**X axis**: One of the axes of the Cartesian coordinate system used in *form-Z*. See also *axes of Cartesian space*.

---

**Y**

**Y axis**: One of the axes of the Cartesian coordinate system used in *form-Z*. See also *axes of Cartesian space*.

**yon plane**: See *hither and yon planes*.

---

**Z**

**Z axis**: One of the axes of the Cartesian coordinate system used in *form-Z*. See also *axes of Cartesian space*.

**z-buffer**: A rendering algorithm used for determining the visibility of a surface at a given pixel. That is, each surface to be rendered is scanned pixel by pixel and the distance of each pixel from the view point is recorded in a memory structure called the z-buffer. The pixel of the surface which is nearest to the view point is used to determine the color at that pixel location on the screen. In *form-Z* this algorithm is used in conjunction with the flat, Gouraud, Phong, preview z-buffer, and full z-buffer methods of rendering. In *form-Z RenderZone Plus*, the latter two methods in addition employ a hybrid algorithm when a modeling scene includes reflective surfaces. While plain z-buffer procedures are used for non-reflective surfaces, accurate reflections and transparencies are calculated using raytracing procedures. See also *raytracing*.

**zoom**: Changing the scale of a drawing or 3D view to make it larger or smaller, depending on whether a zoom in or a zoom out operation is executed. Also, *form-Z* window tools that execute zooming.
Appendices
Appendix A: form•Z Symbol Libraries

The four symbol libraries shown in this appendix are included with form•Z. All libraries contain a modeling and a drafting part.

Kitchen and Dining

3D modeling symbols

2D drafting: top views

2D drafting: front views

2D drafting: side views
Bathroom

3D modeling

2D drafting: top views

2D drafting: front views

2D drafting: side views
Bedroom

3D modeling

2D drafting: top views

2D drafting: front views

2D drafting: side views
Living Room

3D modeling symbols

2D drafting: top views

2D drafting: front views

2D drafting: side views
## Appendix B: Keyboard Commands

<table>
<thead>
<tr>
<th>Macintosh</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: Ctrl A:</td>
<td>Select All UnGhosted (Edit)</td>
</tr>
<tr>
<td>B: Ctrl B:</td>
<td>Redo (Edit)</td>
</tr>
<tr>
<td>C: Ctrl C:</td>
<td>Copy (Edit)</td>
</tr>
<tr>
<td>D: Ctrl D:</td>
<td>Duplicate (Edit)</td>
</tr>
<tr>
<td>E: Ctrl E:</td>
<td>Edit Cone Of Vision (Edit)</td>
</tr>
<tr>
<td>F: Ctrl F:</td>
<td>Fit All (Window)</td>
</tr>
<tr>
<td>G: Ctrl G:</td>
<td>Grab Image (Edit)</td>
</tr>
<tr>
<td>H: Ctrl H:</td>
<td>Hidden Line (Display)</td>
</tr>
<tr>
<td>I: Ctrl I:</td>
<td>Import (File)</td>
</tr>
<tr>
<td>J: Ctrl J:</td>
<td>OpenGL (Display)</td>
</tr>
<tr>
<td>K: Ctrl K:</td>
<td>RenderZone (Display)</td>
</tr>
<tr>
<td>L: Ctrl L:</td>
<td>Shaded Render (Display)</td>
</tr>
<tr>
<td>N: Ctrl N:</td>
<td>New [Model] (File)</td>
</tr>
<tr>
<td>O: Ctrl O:</td>
<td>Open (File)</td>
</tr>
<tr>
<td>P: Ctrl P:</td>
<td>Print (File)</td>
</tr>
<tr>
<td>Q: Ctrl Q:</td>
<td>Quit/Exit (File)</td>
</tr>
<tr>
<td>R: Ctrl R:</td>
<td>Surface Render (Display)</td>
</tr>
<tr>
<td>S: Ctrl S:</td>
<td>Save (File)</td>
</tr>
<tr>
<td>T: Ctrl T:</td>
<td>Quick Paint (Display)</td>
</tr>
<tr>
<td>U: Ctrl U:</td>
<td>Underlay (Windows)</td>
</tr>
<tr>
<td>V: Ctrl V:</td>
<td>Paste (Edit)</td>
</tr>
<tr>
<td>W: Ctrl W:</td>
<td>Wire Frame (Display)</td>
</tr>
<tr>
<td>X: Ctrl X:</td>
<td>Cut (Edit)</td>
</tr>
<tr>
<td>Z: Ctrl Z:</td>
<td>Undo (Edit)</td>
</tr>
<tr>
<td>1: Ctrl 1:</td>
<td>30 60 View (View)</td>
</tr>
<tr>
<td>2: Ctrl 2:</td>
<td>45 45 View (View)</td>
</tr>
<tr>
<td>3: Ctrl 3:</td>
<td>120 20 View (View)</td>
</tr>
<tr>
<td>4: Ctrl 4:</td>
<td>220 45 View (View)</td>
</tr>
<tr>
<td>5: Ctrl 5:</td>
<td>60 30 View (View)</td>
</tr>
<tr>
<td>6: Ctrl 6:</td>
<td>[+ XY] Top (View)</td>
</tr>
<tr>
<td>7: Ctrl 7:</td>
<td>[- XY] Bottom (View)</td>
</tr>
<tr>
<td>8: Ctrl 8:</td>
<td>[+ YZ] Right (View)</td>
</tr>
<tr>
<td>9: Ctrl 9:</td>
<td>[- YZ] Left (View)</td>
</tr>
<tr>
<td>0: Ctrl 0:</td>
<td>[+ ZX] Back (View)</td>
</tr>
<tr>
<td>Macintosh</td>
<td>Windows</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>⌘ =:</td>
<td><strong>OpenGL</strong></td>
</tr>
<tr>
<td>⌘ -:</td>
<td><strong>[- ZX] Front</strong></td>
</tr>
<tr>
<td>⌘ {=:</td>
<td>Ctrl {=:</td>
</tr>
<tr>
<td>⌘ }=:</td>
<td>Ctrl }=:</td>
</tr>
<tr>
<td>⌘ [:=</td>
<td>Ctrl [:=</td>
</tr>
<tr>
<td>⌘ ]:=</td>
<td>Ctrl ]:=</td>
</tr>
<tr>
<td>⌘ =:</td>
<td>Ctrl =:</td>
</tr>
<tr>
<td>⌘ ;=:</td>
<td>Ctrl ;=:</td>
</tr>
<tr>
<td>⌘ &lt;:=</td>
<td>Ctrl &lt;:=</td>
</tr>
<tr>
<td>⌘ ,:=</td>
<td>Ctrl ,:=</td>
</tr>
<tr>
<td>⌘ .:=</td>
<td>Ctrl .:=</td>
</tr>
<tr>
<td>⌘ /:=</td>
<td>F1:=</td>
</tr>
<tr>
<td>⌘ ?:=</td>
<td>Ctrl ?:=</td>
</tr>
<tr>
<td>⌘ `:=</td>
<td>Ctrl `:=</td>
</tr>
<tr>
<td>⌘ tab :=</td>
<td>Ctrl tab :=</td>
</tr>
<tr>
<td>⌘ del :=</td>
<td>Ctrl del :=</td>
</tr>
<tr>
<td>⌘ →:=</td>
<td>Ctrl →:=</td>
</tr>
<tr>
<td>⌘ ←:=</td>
<td>Ctrl ←:=</td>
</tr>
<tr>
<td>⌘ ↑:=</td>
<td>Ctrl ↑:=</td>
</tr>
<tr>
<td>⌘ ↓:=</td>
<td>Ctrl ↓:=</td>
</tr>
<tr>
<td>⌘ shift,↑:=</td>
<td>Ctrl+shift ↑:=</td>
</tr>
<tr>
<td>⌘ shift,↓:=</td>
<td>Ctrl+shift ↓:=</td>
</tr>
</tbody>
</table>

Opt A: Ctrl+shift A: | Arbitrary Reference Plane | (Window) |
Opt C: Ctrl+shift C: | Outline | (Modeling) |
Opt F: Ctrl+shift F: | Face | (Modeling) |
Opt G: Ctrl+shift G: | Grid Snap on/off | (Window) |
Opt H: Ctrl+shift H: | Hole/Volume | (Modeling) |
Opt O: Ctrl+shift O: | Object | (Modeling) |
Opt P: Ctrl+shift P: | Point | (Modeling) |
Opt S: Ctrl+shift S: | Segment | (Modeling) |
Opt U: Ctrl+shift U: | Group | (Modeling) |
<table>
<thead>
<tr>
<th>Macintosh</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt V: Ctrl+shift V:</td>
<td>Perpendicular on/off</td>
</tr>
<tr>
<td>Opt X: Ctrl+shift X:</td>
<td>Reference plane to XY</td>
</tr>
<tr>
<td>Opt Y: Ctrl+shift Y:</td>
<td>Reference plane to YZ</td>
</tr>
<tr>
<td>Opt Z: Ctrl+shift Z:</td>
<td>Reference plane to ZX</td>
</tr>
<tr>
<td>Opt ←: Ctrl+shift ←:</td>
<td>Uniform Scale Decrease</td>
</tr>
<tr>
<td>Opt →: Ctrl+shift →:</td>
<td>Uniform Scale Increase</td>
</tr>
<tr>
<td>Ctrl ↑: Ctrl+Alt ↑:</td>
<td>Rotate Positive X</td>
</tr>
<tr>
<td>Ctrl ↓: Ctrl+Alt ↓:</td>
<td>Rotate Negative X</td>
</tr>
<tr>
<td>Shift+Ctrl ↑: Shift+Ctrl+Alt ↑:</td>
<td>Rotate Positive Y</td>
</tr>
<tr>
<td>Shift+Ctrl ↓: Shift+Ctrl+Alt ↓:</td>
<td>Rotate Positive Y</td>
</tr>
<tr>
<td>Ctrl →: Ctrl+Alt →:</td>
<td>Rotate Positive Z</td>
</tr>
<tr>
<td>Ctrl ←: Ctrl+Alt ←:</td>
<td>Rotate Negative Z</td>
</tr>
<tr>
<td>Shift+Opt A: Ctrl+Alt A:</td>
<td>Dir. to ortho+diagonal</td>
</tr>
<tr>
<td>Shift+Opt C: Ctrl+Alt C:</td>
<td>Obj. s. to face center</td>
</tr>
<tr>
<td>Shift+Opt D: Ctrl+Alt D:</td>
<td>Direct. snap to angle</td>
</tr>
<tr>
<td>Shift+Opt E: Ctrl+Alt E:</td>
<td>Obj. s. to end point</td>
</tr>
<tr>
<td>Shift+Opt F: Ctrl+Alt F:</td>
<td>Object snap to face</td>
</tr>
<tr>
<td>Shift+Opt I: Ctrl+Alt I:</td>
<td>Obj. s. to intersection</td>
</tr>
<tr>
<td>Shift+Opt L: Ctrl+Alt L:</td>
<td>Object snap to line</td>
</tr>
<tr>
<td>Shift+Opt M: Ctrl+Alt M:</td>
<td>Obj. s. to mid point</td>
</tr>
<tr>
<td>Shift+Opt N: Ctrl+Alt N:</td>
<td>No object snap</td>
</tr>
<tr>
<td>Shift+Opt O: Ctrl+Alt O:</td>
<td>Dir. snap to ortho</td>
</tr>
<tr>
<td>Shift+Opt P: Ctrl+Alt P:</td>
<td>Object snap to point</td>
</tr>
<tr>
<td>Shift+Opt R: Ctrl+Alt R:</td>
<td>Dir. snap to radial</td>
</tr>
<tr>
<td>Shift+Opt S: Ctrl+Alt S:</td>
<td>Obj. snap to segment</td>
</tr>
<tr>
<td>Shift+Opt X: Ctrl+Alt X:</td>
<td>No directional snap</td>
</tr>
<tr>
<td>Shift C: Shift C:</td>
<td>Close Drawing Through Point</td>
</tr>
<tr>
<td>Shift E: Shift E:</td>
<td>End Drawing Through Point</td>
</tr>
<tr>
<td>Shift ↑: Shift ↑:</td>
<td>Move Positive Z</td>
</tr>
<tr>
<td>Shift ↓: Shift ↓:</td>
<td>Move Negative Z</td>
</tr>
</tbody>
</table>

*form-Z Directory • Appendices*
Single keys:

**shift:**
- Pick by Frame: preserves previous picks.
- Pick tool: activates Pick Parade mode.
- Extend Grid tool: extends grid to limits of objects.

**esc:** Cancels dynamic operation (i.e. rubber-banding).

**delete** (Macintosh) / **Backspace** (Windows): Deletes the currently selected entities.

**tab:** Goes to the next text field of the dialog, Tool Options palette, and Prompts palette.

**return:**
- Drawing: functions like a mouse click.
- Numeric input: completes the input line.
- Dialog boxes: selects default button.

→: Move Positive X (Modeling)
←: Move Negative X (Modeling)
↑: Move Positive Y (Modeling)
↓: Move Negative Y (Modeling)

F1: Help-General (Windows only) (Help)
F2: Screen Digitizer Input (Modeling)
F3: World Digitizer Input (Modeling)
F4: Capture Digitizer Mesh (Modeling)
F5: Radiosity Options (Modeling)
F6: Initialize Radiosity (Modeling)
F7: Generate Radiosity Solution (Modeling)
F8: Exit Radiosity (Modeling)

C: Close Drawing (Modeling)
E: End Drawing (Modeling)
Keys with clicking:

Option clicking (Mac) or Ctrl+Shift clicking (Windows) on icon or menu item with an asterisk:

Invokes the dialog for that tool or operation.

Option clicking (Mac) or Ctrl+Shift clicking (Windows) to execute a tool that invokes a preview dialog:

Sizes the dialog for a 640 x 480 screen. Otherwise, it is sized to the dimension of the screen in use.

Option clicking (Mac) or Ctrl+Shift clicking (Windows) in a palette:

Invokes the dialog for the respective palette.

Shift clicking to change the visibility of a layer in the Layers palette, lights in the Lights palette or object/group in the Objects palette:

Delays the redraw of the screen until the shift key is released.

Option clicking (Mac) or Ctrl+Shift clicking (Windows) in the title bar to change attributes in the Layers, Lights, or Objects palette:

Turns attributes for all entries on, when at least one of the attributes is off. Otherwise, it reverses the status of each entry.