



Modelling

A Comprehensive User's Guide

The *SOFTIMAGE 3D User's Guide* was written by Susan Duciaume, Camille Lemberg, Suzanne Gerrior, Grahame Fuller, and Edna Kruger; edited by Edna Kruger; illustrations by Richard Z'Graggen; and layout by Paul Verrall.

Special thanks to Nancy Nadeau, Richard Laperriere, Marc Stevens, Andre Foisy, and Laurent Ruhlmann for their assistance in assuring the technical integrity of this guide.

© Copyright 1996 Softimage Inc., a wholly owned subsidiary of Microsoft Corporation. All rights reserved.

Microsoft®, MS®, MS-DOS®, Windows®, and Windows NT® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

SOFTIMAGE® is a registered trademark of Softimage Inc., in the United States, Canada, and/or other countries.

3D Studio Max™ is a trademark of Autodesk, Inc.

Adobe Illustrator® is a registered trademark of Adobe Systems, Inc.

Abekas® is a registered trademark of Abekas Video Systems, Inc.

Alias® is a registered trademark of Alias Research Inc.

Apple® and Macintosh® are registered trademarks of Apple Computer, Inc.

Aurora® is a registered trademark of Chyron Corporation.

AutoCAD® is a registered trademark of AutoDesk, Inc.

Flock of Birds® is a registered trademark of Ascension Technology Corp.

Indigo® is a registered trademark and Indy™ is a trademark of Silicon Graphics, Inc.

Magellan™ is a trademark of Logitech, Inc.

Macromedia® is a registered trademark and Freehand™ is a trademark of Macromedia, Inc.

PostScript® is a registered trademark of Adobe Systems, Inc.

SEGA Saturn™ is a trademark of Kabushiki Kaisha Sega Enterprises.

Targa® is a registered trademark of Truevision Inc.

TIFF™ is a trademark of Adobe Systems, Inc.

UNIX® is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

V-LAN® is a registered trademark of Videomedia, Inc.

Wacom® is a registered trademark of Wacom Co., Ltd.

WSD® is a registered trademark of Accom, Inc.

All other product names mentioned in this guide may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

This document is protected under copyright law. The contents of this document may not be copied or duplicated in any form, in whole or in part, without the express written permission of Softimage Inc. This document is supplied as a guide for SOFTIMAGE 3D products. Reasonable care has been taken in preparing the information it contains. However, this document may contain omissions, technical inaccuracies, or typographical errors. Softimage Inc. does not accept responsibility of any kind for customers' losses due to the use of this document.

Document No. 90436-0596

Printed in Canada.

Table of Contents

CHAPTER ONE

Introduction to Modelling	1
Introduction	3
Building Blocks of Modelling	4
Methods of Modelling	8

CHAPTER TWO

Basic Modelling Tasks	11
Introduction	13
Selecting and Deselecting Elements	14
Deleting Elements	17
Viewing Object Information	19
Displaying and Hiding Elements	21
Transforming Elements	26
Duplicating Objects	32
Performing Boolean Operations	38
Editing Points	42

CHAPTER THREE

Polygon Mesh Objects	49
Introduction	51
Creating a Polygon Mesh Object	52
Converting Objects to Polygon Mesh	56
Merging Objects	58
Cutting Along a Plane	60
Editing Polygon Mesh Objects	61
Modelling Selected Polygons	74

CHAPTER FOUR

Curves	87
Introduction	89
Some Basics about Curves	92
Some Basics about NURBS Curves	94

Drawing Curves.	101
Copying a Curve	104
Opening and Closing Curves.	108
Merging Curves.	110
Cutting a Curve.	114
Adding Lines and Arcs.	115
Converting a Curve	117
Editing a Curve	119
Making 3D Objects from Curves	130
Relational Modelling	143
Projecting Curves onto Surfaces	149
Creating Faces.	150

C H A P T E R F I V E

Patch and NURBS Surface Objects	155
Introduction	157
Creating Surface	160
Opening and Closing Surfaces.	162
Converting Surface Objects	166
Joining Surfaces	168
Extracting a Curve from a Surface	173
Cutting a Surface	175
Editing a Surface	177
Modelling NURBS Surfaces with Projected Curves	185

C H A P T E R S I X

Meta-clay Objects	205
Some Basics of Meta-Clay	207
Creating Meta-clay Elements and Systems.	209
Viewing Meta-clay Elements	211
Editing Meta-clay Systems	212
Controlling How Meta-clay Elements Blend	215
Animating Meta-Clay System Shapes.	218
Using Meta-clay Elements as Envelopes	222
Converting a Meta-Clay System.	223

C H A P T E R S E V E N

Skeletons	225
Introduction	227
A Closer Look at Articulated Chains	229
Drawing Articulated Chains	231
Defining a Skeleton	235
Using Objects in Skeletons	238
Colour Coding Joints and Vertices	241
Adding Skin (Envelopes) to a Skeleton	242

C H A P T E R E I G H T

Envelopes	243
About Envelopes	245
Assigning an Envelope	247
Viewing Envelopment Assignment	257
Editing an Envelope	259
Undoing Envelope Operations	269

C H A P T E R N I N E

Null Objects	271
Introduction	273
Creating a Null Object	274
Using Nulls	275
Examples Using Nulls	277

C H A P T E R T E N

Hierarchies	279
Introduction	281
Creating a Hierarchy	282
Selecting Elements in a Hierarchy	283
Cutting Links in a Hierarchy	285
Manipulating Hierarchies	286
Creating a Hierarchy for Articulation	288

C H A P T E R E L E V E N

Deforming an Object's Shape291

Introduction293

Deforming with Lattices294

Deforming by Curves and Surfaces298

Deforming with Clusters of Points308

Shape Deformation311

Deforming Envelopes312

Deforming with Quick Stretch314

C H A P T E R T W E L V E

Optimizing Models317

Overview of Optimizing Models319

Reducing the Number of Polygons321

Webbing Models354

Generating Quadratic Texture Maps356

I N D E X

C H A P T E R O N E

Introduction to Modelling

Introduction

Before starting the process of modelling, you should know some of the basic terminology and concepts of modelling, and how they are implemented in SOFTIMAGE 3D.

This section describes the features of the Model module, its purpose and special capabilities, and how it fits in with other modules of SOFTIMAGE 3D.

If you have never used a 3D modelling tool, you should read the entire chapter as a useful introduction to the task.

If you are already familiar with 3D modelling, you should familiarize yourself with the types of objects available in SOFTIMAGE 3D.

Building Blocks of Modelling

Modelling is the process of creating and assembling basic shapes into a representation of the desired object. You can group various objects together to work as a model. How a model is built can also determine how it will be animated. SOFTIMAGE 3D is highly interactive and allows you to try different approaches to modelling. By experimenting with different types of building blocks, you can foresee problems you may encounter as a result of your design.

Here are the terms you will see in the course of using this guide.

Points

All elements in 3D space (polygons, curves, and surfaces) are made of points, with each point being an xyz location in space. Point coordinates provide the minimum information from which SOFTIMAGE 3D can calculate the geometry for an element. When you model an object, you enter point coordinates to define the size and shape of the element.

Centres

The 3D world has a centre (called the *origin*) and each object in the 3D world has its own centre. The centre is the reference for defining the object's shape, location, orientation, and size.

For practical purposes, this means that all transformations you perform on an object are really performed on the object's centre.

The centre is related to the coordinate system you choose. For example, the global centre (called the origin) is the centre of the 3D world in which you are drawing.

When you create an object, its (local) centre is located inside the object. However, you can move this centre to another location for performing different modelling tasks. On a polygon mesh object, individual polygons also have their own centre.

This means you can transform objects around one of several possible centres and using coordinate systems, with very different results.

For more information, see *Manipulation and Transformation* modes in the *Interface* chapter of the *Reference Guide*.

Geometry

Geometry refers to the control points owned by an object. These control points are usually seen with objects that can be rendered. For example, a cube's geometry is composed of eight control points. By this definition, a curve has geometry since it is also composed of one or more control points, whereas nulls and meta-clay elements have no geometry.

Curves

Curves are defined by a set of control points. More precisely, they are a collection of curve segments attached by their ends, or *knots*, to make a curve. The look of the resulting curve varies depending on the manner of interpolation between the control points.

In SOFTIMAGE 3D, there are five types of curves: NURBS, B-Spline, Bezier, Linear, and Cardinal.

Objects

An object is anything that has a centre and control points, and can be moved and modelled in the 3D world. Important exceptions are the null object, which has only a centre but no control points, and meta-clay elements, which are based on density rather than geometry.

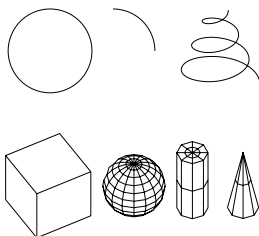
Cameras and lights are not objects since they have no control points. Instead, they are called *elements*.

Objects in SOFTIMAGE 3D can be classified in several ways. They can be classified according to whether they are polygon mesh, surface, curves, faces, meta-clay, or nulls.

Predefined Primitives

The simplest objects to use are predefined objects called primitive objects. These are basic 2D and 3D shapes such as circles, spheres, cubes, spirals, multi-sided figures, and so on. These primitives can be used “as is” or modified. Typically, you select a primitive form and then transform it, cut it into pieces, or combine it with other modified pieces, which is much easier than drawing it point by point.

Primitives can be polygon mesh objects, surface (patch or NURBS), or predefined curve shapes (arcs or circles).



Some primitives

Polygon Meshes

Polygon mesh objects are made up of ... polygons! Polygons can be boxy, such as the traditional cube, or smooth, depending on the number and size of polygons used to create the object.

No matter how you define the object for modelling purposes, all objects are rendered as polygons.

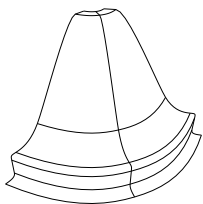
Curves

Most of the time, you will draw a curve and model with the resulting profile. You can also select a few predefined curves such as arcs or circles. Curves come in various types, each with its characteristic behaviour and possible modelling uses.

Faces

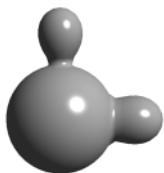
A face is basically a curve which is closed and filled in. A face with holes in it is made up of an exterior curve and interior curves. In SOFTIMAGE 3D, predefined text is made out of face objects. Faces can also be extruded to form 3D objects.

Surfaces



A surface is an object made up of parametric surface curves, rather than geometrical polygons. Surface objects in SOFTIMAGE 3D can be either patch or NURBS. Most operations that apply to curves apply equally to surfaces; however, surfaces can also have textures applied to them which curves cannot. Surfaces can usually be converted to polygon mesh objects, but the opposite is rarely true.

Meta-Clay Elements



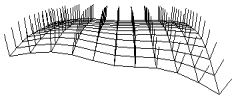
Meta-clay elements are objects with no geometry, only density and blending ability. Typically, they are “blobs” which you can place together to form smooth, soft-edged organic objects. As you pull them apart, the blobs stretch out like taffy, and finally separate into separate blobs.

Nulls



A null has a centre with no geometry, but can still be transformed like any other object. It is very useful as a reference tool, for precision modelling, constraining animation, and for building hierarchies of objects. It also provides you with an easy way to indicate a precise spot or orientation.

Normals



On faces, polygon mesh objects, and surface objects, the control points form closed areas. Normals are vectors perpendicular to these closed areas on the surface. Their purpose is to indicate the visible side of the object and its orientation to the camera. Normals are computed to optimize shading between surface triangles.

By default, the renderer shows only the side with the normals pointing to the camera; however, it can also render the back side or both sides.

When normals are oriented in the wrong direction, causing modelling or rendering problems, you can use the **Effect ▶ Inverse** command in the Model module to invert them.

In SOFTIMAGE 3D, normals are represented by thin blue lines.

For More Information

For information on the modelling tasks you can perform using these basic building blocks, see the following chapters:

- *Polygon Mesh Objects* on page 49
- *Curves* on page 87
- *Patch and NURBS Surface Objects* on page 155
- *Meta-clay Objects* on page 205
- *Null Objects* on page 271

Methods of Modelling

These are some of the ways you can model an object in SOFTIMAGE 3D. Many of them have different results depending on the type of object you are modelling, but the basic procedure is the same for all objects.

Transforming Elements

Transforming refers the changing an element's size (scale), orientation (rotation), or position (translation) in 3D space. All modelling is based on an object's transformations.

You can choose what to transform in a particular operation: the entire object, several objects at a time, only the centre of objects, tagged points on objects, only polygons, and so on.

Added to this, you can indicate whether you want to rotate, scale, or translate an object with reference to its own centre, around the global origin of the 3D space, around another object's centre, and so on. The possibilities are discussed in *Transforming Elements* on page 26.

Duplicating Objects

You can duplicate objects in different ways: creating a single copy, multiple copies, object "instances," or virtual representations of the same model.

Duplicating does not necessarily refer to duplicating entire models. You can create a symmetrical (mirrored) copy of a curve for example, and then mirror it and extrude. This can be useful in modelling shapes which might otherwise be difficult to model precisely. See *Duplicating Objects* on page 32 for more information.

Editing Points

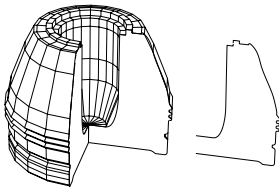
One of the fundamentals of modelling is once you create an object, how can you change its points?

Some of the various means are tagging them (a visual cue for keeping track of point selection) and then grouping, adding, and deleting individual points or rows of them. You can deform (change the position of) all the points on an object in one action, or only selected groups of points. For more information, see *Editing Points* on page 42.

Merging Objects Together

There are different approaches to merging objects and defining the results you want. The methods depend on the type of objects you are merging: curves, polygon mesh objects, patch, or NURBS surfaces. Refer to the respective chapters on these object types.

Creating 3D Objects from Curves



You can draw a curve, and create any kind of 2D or 3D object from it, using such methods as extrusion, revolution, and skinning. The techniques for doing this are explained in *Making 3D Objects from Curves* on page 130.

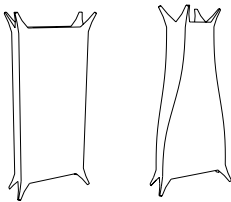
You can also alter the shape of a polygon mesh object dramatically by selecting individual polygons and extruding them along a curve that you have drawn. This is discussed in *Modelling Selected Polygons* on page 74.

Relational Modelling

As the name suggests, relational modelling creates a relationship between something you use to generate an object, such as a basic curve that you draw, and the object that you generated from it. For example, if you drew a curve and extruded it to form a path or road that snaked over the hills into the distance, it would be easy to just edit the original curve shape and see the course of the path immediately change.

For more information, see *Relational Modelling* on page 143.

Deforming an Object's Shape



Another quick, indirect modelling method is deformation. Rather than pulling individual points on an object to edit it, you could use a more global, automated technique.

Typically, deformation methods act upon the object indirectly, using one object to affect the shape of another object. For example, model a pie and make it deform to the face of the character – splat! – and then have it run down his face.

Some deformation techniques include lattices, deforming an object along a curve or a surface, creating clusters of points to be deformed, waves, and the cartoon-like Quick Stretch effect. These techniques and others are discussed in the chapter *Deforming an Object's Shape* on page 291.

There's a lot more to read about in this and other SOFTIMAGE 3D User's Guides, but now it's time to go to the 3D drawing table.

Happy modelling!

C H A P T E R T W O

Basic Modelling Tasks

Introduction

Before you start modelling objects, there are many basic operations that you need to perform, such as:

- Selecting or deselecting elements.
- Viewing information on objects to decide how to modify them.
- Applying an operation to the whole object or just part of it.

Based on your selections, you can then:

- Duplicate objects.
- Transform objects.
- Modify the geometry of an object by:
 - Adding or deleting points.
 - Moving points.
 - Tagging points.

The operations described in this chapter are applicable to most types of objects, unless otherwise specified. Where information and procedures are object-specific, the appropriate chapter is referenced.

Selecting and Deselecting Elements

Selecting and deselecting objects and other elements (lights, cameras, etc.), is one of the simplest and most frequently performed tasks that you perform in SOFTIMAGE 3D. There are several ways to select and deselect elements.

Selecting Single and Multiple Elements

In any window, including the Schematic and Spreadsheet, you can select one or more elements using either the commands in the **Select** menu cell or the space bar Supra key.

1. Press the space bar. The status line now says “Select object.”

This is the equivalent of choosing the **Select → Select Mode** command.

2. Click on the object.

The object is now highlighted in white in the windows. Your next operation such as duplication, transformation, and so on applies to the selected object.

To select multiple objects:

1. Choose the **Multi** menu cell.
2. Press the space bar Supra key and click on each object in turn, or draw a rectangle around all the objects to select them.

To select all objects in the scene, choose the **Select → All** command.

Selecting versus Picking

Selecting and picking are two similar actions with different conditions applied to them. Selecting indicates that the object (or objects) which is highlighted is to receive the actions that you are going to perform.

Picking is related to already being in a certain mode of action. When you pick, you simply click an object without holding down the space bar. For example, if you have drawn several curves and choose the **Surface → Skin** command, you are in Skinning mode so you can click on each curve without pressing the space bar.

Deselecting Elements

There are several ways to deselect one or more elements:

- Pressing the space bar and clicking on the selected object toggles the selection off, without selecting anything else. This is the equivalent of choosing **Select → Unselect Mode** and clicking the object(s).
- Selecting another item: the current selection is automatically deselected (in **Single** selection mode only).
- Choosing **Select → Clear** deselects all elements in the scene.

Inverting Selection Status

There are several ways to invert the selection of elements:

- Choosing **Select → Toggle** inverts the selection status of all elements in the scene.
- You can also toggle the selection of one element on and off simply by clicking on it with the space bar Supra key.
- When you hide an object using **Display → Hide → Toggle and Desel. Hidden**, you can hide the object and deselect it at the same time.

Selecting Elements in a Hierarchy

When objects are in a hierarchical relationship, you can use the different mouse buttons to select a single object in a hierarchy, a branch of the hierarchy, or the whole tree.

This is different than selecting multiple individual elements in **Multi** mode because it respects the hierarchy links that are created. For a description of how objects in hierarchies respond to transformation, see *Hierarchies* on page 279.

Selecting a Node

1. Press the space bar.
2. Left-click on any object in the hierarchy, whether it is a parent or a child.

Only the object is selected, not the rest of the branch.

Selecting a Branch

1. Press the space bar.
2. Middle-click on the parent of a branch. The branch is selected.

This only works on the parent. If you middle-mouse click on the child, this does not select the branch.

With this type of selection, you can only deselect the whole branch and not individual elements. For an alternative, see *Multi-Node Selection* below.

Selecting the Tree

1. Press the space bar.
2. Right-click on any object in the hierarchy.

The entire tree containing the object is selected.

Multi-Node Selection

When you choose the **Select → Multi-Node** command and select a hierarchy, you can then deselect individual nodes, as desired.



This is the same as if you node-selected individual elements of the branch or tree in **Multi** mode. The approach you choose depends on whether you want to apply an operation to only a few elements in a large hierarchy, or to most of them. In both cases, you can transform selected nodes.

By contrast, if you use the typical middle-mouse to select a branch, you can only deselect the whole branch and not individual elements.

1. Choose the **Multi** menu cell.
2. Choose the **Select → Multi-Node** command.
3. Select a branch or tree.
4. Click any branch or node in the hierarchy to deselect it.

Deleting Elements

There are a few different ways to delete elements. You can also choose to delete any number of elements at a time.

Deleting Selected Elements

To delete an element, the easiest method is to use the Backspace Supra key:

1. Press the Backspace key on the keyboard.
2. Click on the object(s) you want to delete.

You also can select one or more objects and use the **Delete → Selection** command to delete them. If you have just saved the scene, the selected elements disappear; otherwise, a message box appears prompting you to confirm the deletion.

Deleting Polygons

You can delete polygons on a polygon mesh object by tagging points that circumscribe them and then deleting the points:

1. Tag the points of the polygons you wish to delete (using the t Supra key).
2. Select the TAG manipulation mode.
3. Press the Backspace key and click on any of the points to delete.

In the Shade view, you see that the associated polygons are deleted.

Note

The vertices around the deleted portion remain tagged. You may mistakenly try to delete the points, in which case the message appears “There are no tagged polygons.”

You can also delete selected polygons using the **Delete → Selection** command when you are in the POL manipulation mode:

1. Choose the POL manipulation mode,
2. To select the polygon, press the g Supra key and click on the desired polygon(s). Selected polygons are highlighted in magenta.
3. Choose **Delete → Selection**.

The selected polygons are deleted.

Clearing the Workspace

If you want to clear the current workspace, choose **Delete → All**. You should save any changes to your current scene before using this command.

1. Choose the **Delete → All** command.

If your scene has not been modified since the last time you saved it, a message box appears asking you to confirm that you want to delete your work.

If your scene has been modified since the last time you saved it, this warning box is displayed.



2. Click **Ok** to delete everything, or **Cancel** to exit the message box.

Tip

When you want to start a new scene from scratch after working on many projects in a long session, it is better to use the **EXIT** menu cell and exit the software, not use **Delete → All**.

Why? Here are two good reasons:

- Memory, just like the hard drive, becomes fragmented over time and makes the system run more slowly. When you exit the software, the operating system process is terminated which frees up all allocated RAM and memory swap usage. This effectively gives the system a “clean-slate.”
- The **Delete → All** command does not recuperate the memory used by scene elements since they are still stored in history. Choose the **History → Dispose** command to make sure you clean everything.

It is also a good idea to log out from your account/machine once in awhile. This ensures that you are starting again from scratch.

Viewing Object Information

You can select any object and view information on its attributes.

1. Select any curve, face, polygon mesh object, surface object, or meta-clay element.
2. Choose the **Info** → **Selection** command.

A dialogue box specific to the object type is displayed. The type and amount of information depends on the object type, as shown in the following example.

NURBS Surface Info

Prefix :

Name :

Parent : nil

Sibling : nil

Child : nil

Triangles: 60

Modify : SRTGH

Selection: NODE

Matte ☐ Garbage Matte

	U	V
Degree :	CUBIC	QUADRATIC
Point :	6	3
Close :	no	no
Curve :	1	1
Step :	10	1
Capping:	☐ Bottom ☐ Top	☐ Bottom ☐ Top
Parameterization:	NON UNIFORM	

For most object types, the Info dialogue box allows you to edit a few parameters which cannot be edited anywhere else.

Some examples:

- For curves and surfaces, you can edit the **Step** value (resolution) which affects their appearance in the Shade view and in rendering.
- For surfaces, you can turn **Capping** on or off to determine how the U or V borders can be manipulated.
- For Bezier curves, you can edit the continuity of each point.

- For polygon mesh objects, you can change the method of computation of the object's normals.

The **Info → Selection** command is used in many editing operations, described in the relevant chapters for each object type.

Custom Effects for Information

The **Info → ModelNote** command in the Model module is a tool for adding text notes to an object or model, such as how the model was built, what remains to be done, or messages to other people working on the model or scene. Choose the **Info → ModelNote** command and click on the model you want to notate. A standard text editor appears on the screen. Previous notes made with ModelNote are automatically included in the text editor window.

The **Info → Measure** command in the Motion module computes a variety of “measurable” aspects of an object such as surface area, volume, or length. The result of the calculation is displayed in a small window on top of the SOFTIMAGE 3D display. Transformations are included in the results, and are measured in SOFTIMAGE 3D units.

For more information on using these commands, see their command descriptions in the *Reference Guide*.

Displaying and Hiding Elements

The **Show** and **Display** menus contain most of the toggle commands for displaying or hiding specific elements in the scene. Hiding and displaying elements lets you work with them, apply operations only to them, or exclude them from an operation.

Displaying and Hiding Visual Elements

Elements such as object centres, control points, and object normals are useful to visualize for many modelling and animating tasks. For example, to find out the direction of a curve, you could display the edge flags; or to verify in Wireframe view whether a surface is correctly oriented for rendering purposes, try displaying the object's normals.

The **Show** menu commands, available in all modules except Tools, allow you to toggle these selections on or off. To do so, simply choose the **Show** command you want.

Some operations automatically toggle these selections. For example, when you use the **t Supra** key to tag points, **Show → Tag** is automatically toggled on so that you can immediately see the results.

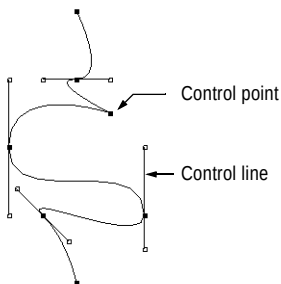
When you draw a curve, **Show → Point** and **Show → Line** are automatically toggled on, but toggled off again when you press the **Esc** key to end the drawing mode.

When you select a polygon on a polygon mesh object, **Show → Polygon** is automatically toggled on to show your selection.

Showing Object Attributes

You can show or hide:

- An object or curve's control points to facilitate editing.
- An object's centre in order to scale, rotate, or translate it.
- Null objects in the scene: for example, nulls used as cluster centres or reference frames.
- Skeleton chains in order to properly position models as envelopes over the skeletons.
- Any lattices on objects in the scene so that you can edit the lattice.
- Edge flags to open or close an object and offset the point at which the object is open or closed. This can be useful when trying to

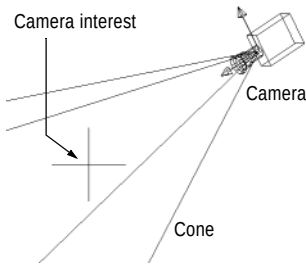


merge open objects if the open segments of the objects do not match up. It can also help you see the direction of a curve because edge flags show you the beginning of a curve.

- Bounding models controlling envelope assignment.
- Links between a parent and children in a hierarchy. The link is shown as white line from the centre of the parent object to the centre of its child(ren).
- Selected polygons on a polygon mesh object.
- To make it easier to model objects in relation to others, you can also show points, lines, tagged points, and the camera cone for objects which are *not* currently selected.

Not all commands for viewing are in the **Show** menu and there is often interaction between selections in the **Show** menu and other menu selections. For example:

- To show the area of influence of a meta-clay element, choose **Meta-Clay → Show Influence**.
- To show a wireframe representation of the blended surfaces of all meta-elements, choose the **Meta-Clay → Turn ON Surfaces** command.
- To show the camera, choose **Camera → Show**. If **Camera → Show** is selected, you can also toggle **Show → Cone** on or off to show or hide a cone outline representing the area shown by the camera view or the area covered by the cone of the selected spotlight. If the interest of the spotlight is selected, this command displays the cone of the light.



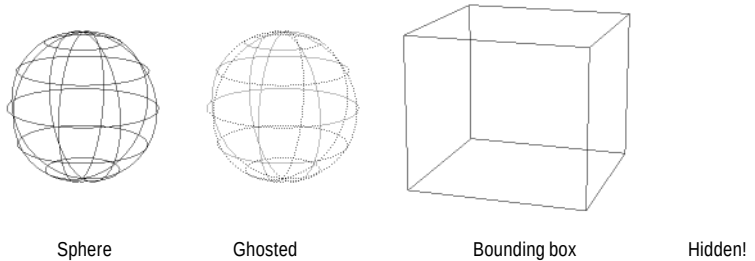
Default Settings

When you save a scene, the settings in the **Show** and **Preference** menu commands are saved in a setup file for that scene. This is stored in the database in the **SETUP_SOFT** chapter in the `<scene_name>.sts` file.

You may save preferred settings to an `.sts` setup file using the **Preferences → Setup File → Save** command, then load it using the **Preferences → Setup File → Load** command, or using the `-s` option when you start **SOFTIMAGE 3D** from the command line. For more information, see *Saving and Loading a Setup File* on page 102 of the *Working with SOFTIMAGE 3D User's Guide*.

Displaying and Hiding Elements

The **Display** menu commands are toggles which allow you to display objects in simplified forms or to make them completely invisible to simplify a crowded scene or to speed up the refresh rate. This menu is available in all modules except Tools.



Hiding Elements

The **Display** → **Hide** commands allow you to make selected elements (objects, cameras, lights) invisible in a window. It may be useful to simplify a crowded display, or if you want to use an object in a scene for modelling purposes, but you don't want to render it.

You can hide or show all elements in the scene in a single operation, or toggle elements' status individually.

When you hide or show elements, you can also keep them selected or automatically deselect them, depending on your preference.

A hidden object is not visible in any window or in the Shade view. In the Schematic window, it is displayed as an empty rectangle. When you save a scene, all hidden objects are saved as "hidden."

1. Select the object, camera, or light.
2. Choose one of the **Display** → **Hide** commands:
 - To show or hide the selected object and automatically deselect it, choose **Toggle and Desel. Hidden**.
 - To show or hide the selected object and keep it selected, choose **Toggle and Keep Selected**.
 - To hide all objects in the scene which are not currently selected, choose **Unselected**.
 - To show all objects in the scene, regardless of their selection status, choose **Unhide All**.



If you want to “unhide” objects individually, select them in the Spreadsheet and use the on/off toggles in the NVisibility column.

- Conversely, it is sometimes useful to ghost objects (represented with a dotted line) rather than completely hiding them. If you have hidden objects and you want to display them as ghosted instead, choose **Display → Ghost → Hidden**.
- To hide objects which are currently ghosted, choose **Ghosted**.

Animating the Visibility of Objects

You can animate the visible/invisible status of an element as follows:

1. Select the object (or branch of a hierarchy).
2. Choose either the **SaveKey → Object → Node Visibility/Branch Visibility** or the **SaveKey → Branch → Node Visibility** command in the Motion module.

This saves the current status of the object as Visible.

3. Move the time line to the frame where the object should be hidden.
4. Choose the **Display → Hide → Toggle & Keep Selected** command.
5. Again, choose the appropriate **SaveKey → Object or Branch Visibility** commands.

This saves the current status of the object as Hidden.

Displaying Objects in a Simplified Form

The **Display → Bounding Box** command displays a simplified representation (a box) of one or all objects in the scene, which accelerates the screen refresh.

Like ghosting, this is useful for speeding up transformations. It can also give a rough idea of how an animation sequence is running when you don't need all objects to be shown in their final form.

Ghosting Objects

The **Display → Ghost** command displays selected objects with a dotted line representation in the windows. However, ghosting does not change the object's representation in the Schematic window.

Ghosted objects are not refreshed during a transformation, but only when you release the mouse button. During the transformation, they are displayed in the simplified form of a bounding box.

Unlike hidden objects, a ghosted object can still be tagged, transformed, and otherwise manipulated but at an optimized refresh rate.

 This command is especially useful when working with lattices to deform objects. Ghost the lattice to move its points faster. The object deforms when you release the mouse button.

1. Select an object, camera, or light.
2. Choose the **Display → Ghost** command and one of its subcommands:
 - **Toggle Selection:** If the object was previously unghosted, it is now ghosted, or vice-versa. This operation applies only to the selected object, not to all objects in the scene.
 - **Unselected:** All unselected objects become ghosted.
 - **Unghost All:** All objects in the scene become ghosted.
 - **Hidden:** Objects that were previously hidden are now ghosted.
3. If you want to change all ghosted objects to hidden objects, choose the **Display → Hide → Ghosted** command.

Transforming Elements

When you transform an element, you change its scale (size), rotation (orientation), or translation (position). Naturally, this is one of the most fundamental and frequent modelling tasks in creating a model and setting up a scene.

Depending on which manipulation mode you're using (see *Manipulation Modes* on page 13 of the *Reference Guide*), you can transform these elements:

- In OBJ (object) manipulation mode, you transform objects.
- In TAG (tagged points) manipulation mode, you can scale, rotate, and translate only the area designated by tagged points.
- In CTR (centre) manipulation mode, you transform only the object's centre.
- In TXT (texture) manipulation mode, you transform the object's texture.
- In POL (polygon) manipulation mode, you transform only selected polygons of polygon mesh objects.

Modes are Important!

Always make sure that you're using the right manipulation mode. For example, if you're trying to rotate an object and nothing happens, you might be in CTR mode! The same thing applies to the transformation modes. Objects will refuse to rotate around their own centres if you are in GBL (Global) transformation mode.

Changing an Element's Size (Scale)

The **Scale** menu cells let you scale objects or points on objects interactively with the mouse, or numerically in the **Scaling** dialogue box. An object is scaled from its own centre (local).

Scaling Objects

When an object is created, its scale is set to 1. For example, a cube created with a length of 5 and scaled with a factor of 2 has a length of 10. A negative scaling yields an inverted object.

1. Select the item and the appropriate manipulation mode.

For example, if you plan to scale the entire object, select OBJ mode, whereas if you want to scale only the tagged points on it, select TAG mode.

2. Choose the entire **ScaleXYZ** menu cell group (click on the right edge of the menu cells) or one of its menu cells (**ScaleX**, **ScaleY**, or **ScaleZ**).

You can also press and release the x Supra key to scale.

3. Select the Scaling mode from the default XYZ (see *Scaling Modes* on page 1248 of the *Reference Guide*).
4. Scale the object using the mouse buttons or the keyboard.
 - If the entire **ScaleXYZ** command is selected, you can press one or all of the mouse buttons at once and scale the object interactively. The **Scale** menu cell values are updated interactively.
 - If you click inside one of the menu cells, you can scale interactively with the mouse or enter a new value in the menu cell to scale the object only in that direction
 - If you click in the tab in the upper-left corner of one menu cell, the Scaling box appears in which you can enter scaling values for one or all of the xyz axes.

Scaling		
X:	1.0	Add Set Cancel
Y:	1.0	
Z:	1.0	

Add increases or decreases the scaling by the value in the X, Y, Z text boxes. For example, if the default for X is 1, you change the value to 2 and click **Add**, the value of 1 is multiplied by 2, and the result is 2.

Set sets the scaling to the value you specify in the X, Y, Z text boxes.

You can also use negative values, which are added or set accordingly.

Scaling Tagged Points

You can limit scaling to an area on an object by tagging points first.

1. Select an object and choose **Show → Point** to display its points.
2. Tag points on the object to define the area you want to scale.
3. To limit the scaling to the tagged area, select the TAG manipulation mode.
4. Scale the points using the steps described for scaling objects.

Changing an Element's Orientation (Rotation)

The **RotXYZ** menu cells let you change the orientation of objects or tagged points on objects interactively with the mouse, or numerically in the Rotation dialogue box.

The orientation of an object is set by the rotation of its centre. All possible orientations can be set with a combination of three angles of rotation: x-axis, y-axis, and z-axis.

Objects rotate around their centres (local), but will rotate differently on the axis depending on the rotation mode defined (LCL, GBL, ADD, REF, and PLN – described in detail in *Rotation Modes* on page 1219 of the *Reference Guide*).

Rotating Objects or Tagged Points

When an object is created, its rotation is set to 0.

To rotate the entire object or tagged points on it, follow the same procedures as those for Scaling on page 26, simply substituting the **RotXYZ** menu cells and rotation modes. The Rotation dialogue box is identical to the Scaling dialogue box except that it relates to scaling.

The c Supra key selects Rotation menu cells.

Moving an Element (Translation)

The **TransXYZ** menu cells let you change the location of elements or tagged points on objects interactively with the mouse, or numerically in the Translation dialogue box. The location is specified in XYZ coordinates.

When an object is created, it is located at the global centre (origin), and its translation values are set to 0 in the **TransXYZ** menu cells.

Objects translate along the three axes, but translate differently depending on the Translation mode selected (LCL, GBL, PAR, REF, or DRG – defined in detail in *Translation Modes* on page 1427 of the *Reference Guide*). Points translate along all three axes, but the movement depends on the translation mode (LCL, GBL, and so on).

Translating Objects or Tagged Points

When an object is created, its translation values are set to 0.

To translate the entire object or tagged points on it, follow the same procedures as those for Scaling on page 26, simply substituting the **TransXYZ** menu cells and translation modes. The Translation dialogue box is identical to the Scaling dialogue box except that it relates to translation.

The v Supra key selects the Translation menu cells.

Transforming in Hierarchies

The way objects in hierarchies respond to transformations depends on two commands in the **Preferences** menu cell: **Hierarchy Compensation** and **Hierarchy Manipulation**, which can be toggled on or off.

Hierarchy Compensation affects the relative positioning of objects when you link them in a hierarchy.

Hierarchy Manipulation allows you to maintain the position of children in relation to their parent when you transform the parent, regardless of whether you are in node, branch, or tree selection mode.

For details on these two commands, refer to the *Hierarchies* chapter on page 279, or to the command descriptions in the *Reference Guide*.

Returning to Initial Size, Orientation, and Position

If you make a mistake, you can remove any changes you made in scaling, rotation, or translation values and return the object to their initial size, orientation, or position. To do this, apply the same steps as for editing transformations.

For example, if you have scaled the object, you can select it and use the mouse or the Scaling dialogue box to reset any or all of scaling values to the default of 1.

For rotation and translation, you can reset the values to the default of 0.

Note

This operation changes the object's appearance. Contrast this with freezing an object's transformations, which leaves the object unaffected.

Using a Custom Effect

You can also reset all transformations applied to an object using the **Effect → ResetTransf** custom effect in the Model module, using a single mouse click instead of having to changing the values in the transformation dialogue boxes.

For more information on this effect, see its command description in the *Reference Guide*.

Freezing Transformations

The **Effect → Freeze** commands in the Model module allow you to freeze one set of transformations or all of them.

When you freeze an object's transformations, the object is unchanged in the windows. However, the transformation menu cell values are reset to the default values: Scaling to 1, and Rotation and Translation to 0. Furthermore, the object's centre is scaled to 1, 1, 1, it is rotated to 0, 0, 0, and it is translated to 0, 0, 0, the scene's origin.



This procedure is not reversible.

1. Select an object.
2. Choose the **Effect → Freeze** command and one of the following subcommands:
 - **Transformations** resets scaling, rotation, and translation values for all axes in the transformation menu cells to their default values, without affecting the object's position, orientation, or size in the windows.
 - **Scaling** resets values in all the **Scale** menu cells to 1.0 without affecting its size in the windows.
 - **Rotation** resets values in all the **Rotation** menu cells to 0.0 without affecting the object's orientation in the windows.
 - **Translation** resets values in all the **Translation** menu cells to 0.0. The object's centre is moved to the scene's origin (0,0,0 or the "centre of the world") without affecting the object's position in the windows.

Why would you do this? When you want the current state of the object to be its “home” or “neutral” state. Further transformations are performed using the current size, rotation, or translation as “ground zero.” This helps to avoid any unpredictable offsetting (unexpected placement of objects in space) when you apply a deformation.

For example, before you deform an object along a curve or surface, you should move the object to be deformed back to the origin and freeze the object’s transformations. For more information, see *Deforming by Curves and Surfaces* on page 298.

Custom Effects for Copying Transformations

The **Duplicate → CopyTransform** custom effect in the Model module allows you to copy a source object’s transformation to another object. For more information on this effect, see its command description in the *Reference Guide*.

Duplicating Objects

There are several ways of duplicating elements in SOFTIMAGE 3D, including:

- Basic duplication
- Instantiation
- Creating a mirror image of the object
- Using some custom effects

Duplicating Objects One or More Times

You can duplicate objects singly or repetitively using the **Duplicate** commands **Immediate** or **Repetition**. You can also duplicate an object along an animation path using the **Duplicate → From Animation** command.

When you duplicate an object, actual copies of objects (or hierarchies) are created. That is, when you create one object and duplicate it 10 times, you get 10 objects. Contrast this with object instantiation, described in *Creating Instances of Objects* on page 33.

1. Select one or more elements or hierarchies.
 - To duplicate tagged points, you must be in TAG mode.
 - To duplicate polygons, you must be in POL mode: select the object, then select the desired polygon(s), using the **g Supra** key.
2. To make one copy, choose the **Duplicate → Immediate** command, or use the **d Supra** key and pick the object to be duplicated.
 - The left mouse button duplicates a node.
 - The middle mouse button duplicates a branch.
 - The right mouse button duplicates a tree.In POL mode, Duplicate works the same for all mouse buttons.
3. To create multiple copies, choose **Duplicate → Repetition**. The Repetitive Duplication dialogue box appears in which you can specify the number of duplications and successive transformations (scaling, rotation, and translation) to be applied to each of the duplicates.

The duplicates are created in a hierarchy with a null object as parent.

4. To create copies of an animated object at regular frame intervals, choose the **Duplicate → From Animation** command. The related dialogue box lets you generate either:
 - An exact number of duplicates.
 - or
 - A duplicate at a specified frame interval, such as at every 4 frames.

Duplicating Selected Polygons on Objects

Using **Duplicate → Repetition**, you can practically automate the process of duplicating and transforming selected polygons, which greatly accelerates the modelling process.

Selected polygons on a polygon mesh can be duplicated and then transformed without affecting the rest of the object. You can automate this process using either **Duplicate → Repetition**, or using a process of extruding the selected polygon along a curve.

Examples of the type of results you can achieve, and the procedures involved, can be found on page 79.

Creating Instances of Objects

With object instances, you create one master object (or hierarchy), then create many instances which “point to” the single master. An instance is like a virtual replication of the master.

Instantiation is a quick and easy way to replicate objects for depicting groups, such as many buildings for a cityscape, people in crowds, soldiers in a parade, and so on.

When you select an object and choose **Duplicate → Instance** or **Instance Repetition**, it allows you to reference the object any number of times from different locations in a scene, without copying it each time.

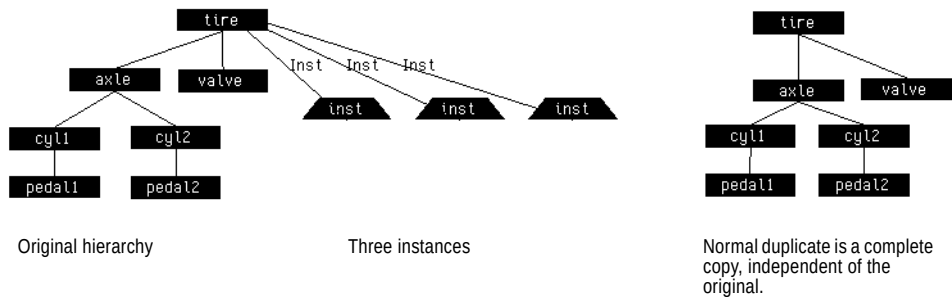
It has the following advantages:

- Instances use much less disk space than duplicates.
- Editing multiple objects is very simple.
- Wireframe, shading, and memory operations are faster.

However, there are some limitations for using instances. For a few suggestions on how to avoid problems, see *Tips on Building Hierarchies with Instances* on page 35 and *Textures and Instances* on page 35.

Structure for Instances

In the Schematic window illustration below, compare the relationship of objects and their duplicates versus instances.



In this example, the hierarchy on the left has one duplicate (shown on the right) and 3 instances.

Notice that duplicate's structure is completely independent and mirrors the hierarchy of the original. On the other hand, each instance is linked to its master. An instance is shown as a trapezoid in the Schematic window, a collapsed version of its master's hierarchy; this collapsed structure can be expanded.

How Do Instances and Masters Interact?

Instances are implemented as null objects. When you choose **Info** → **Selection**, the Null Info dialogue box appears, and the Nature parameter indicates that it is an instance.

Transformations (scaling, rotation, translation) are local to the instance. However, any change in the *geometry* of the master is reflected globally in its instances.

Note To view the links between a master and its instances in the scene, you must be in Model mode in the Schematic window.

Creating Instances

1. Create your master object or hierarchy.
2. Select one of the **Duplicate** menu commands related to Instances. You can create one instance using **Duplicate** → **Instance** or many using **Duplicate** → **Instance Repetition**.

Converting Instances to Objects

You can “expand” an instance which replaces the instance with a real copy of the master. This master can then be manipulated independently.

For procedures and examples, see the **Duplicate → Expand** and **Duplicate → Expand All** commands on page 347 and page 348 in the *Reference Guide*.

Swapping Masters

You can swap master “A” for master “B” using the **Duplicate → Swap** command. For example, swap two masters and presto! 20 cars are now 20 trucks. You might also draw a new model of your car to add some nifty fins, swap it with the current master car, and have 20 instances of the new model, all with nifty fins.

For a tutorial on creating and working with instances, see the *Flower Power* tutorial in the *3D Self-training Workbook*.

Tips on Building Hierarchies with Instances

There are some restrictions to using instances and masters in hierarchies.

- You can only instantiate the root of a hierarchy.
- Instances can also be parents in a hierarchy. On the other hand, an instance cannot be both an instance of a given object and a child of that object at the same time.
- Masters and instances can be siblings, but the order in which you parent them is important and can result in problems rendering instances.

Textures and Instances

An instance can inherit the material and 2D texture of its master. Alternately, you can apply a different material and texture to an instance, which will override that of the master; for example, you can apply colours to a single instance without affecting the master or other instances.

Instances cannot inherit the 3D texture of its master; attempting to do so may give unpredictable results for both instances and masters. Instead, give each its own copy of the texture.

Creating a Mirror Image of an Object

You can model an object and then create a mirror image of it using the **Effect → Symmetry** menu command. This is useful for modelling symmetrical objects, such as two hands, which might otherwise be time-consuming to model.

You can mirror an object along a specified global plane or a user-defined plane.

1. Select an object and position it.
2. Choose the **Effect → Symmetry** command. The Symmetry dialogue box is displayed.
3. Select the plane (xy, yz, xz) on which you want to create the mirrored object, or set a specific plane equation.
4. Click Ok to accept the settings or Cancel to exit the dialogue box.

Example

You could model half an object and use **Effect → Symmetry** to mirror it. Use the **Effect → Merge** command to merge the two symmetrical halves, then choose **Effect → Cleanup** to eliminate any “messy” edges, if necessary.

As another example, say you want to model a TV set with its rounded corners. This can be difficult to do this using a curve, and the process would lack precision.

1. Turn grid lock on in the window Layout dialogue box.
2. Get a primitive NURBS arc to create the top-right corner of the TV. Accept the default values for Begin and End.
3. Translate the curve 5 units along x.
4. Choose **Effect → Symmetry** in the Model module.
5. In the Symmetry dialogue box, choose the YZ plane with $x = 0$.

The mirror of the corner is created at -5 units in x.

6. Use **Draw → Merge Curves** to join the two with straight line segment.
7. Mirror the result to create the bottom edge and corners.
8. Use successive merge operations to connect all the segments.
9. Finally, use **Surface → Extrusion** to make a filleted box.

Duplicating Using Custom Effects

There are four custom effects that also let you duplicate an object in different ways: **Bridge**, **Duplicator**, **VolumeDuplicate**, and **CopyCenter**. For detailed information on using these effects, see their command descriptions in the *Reference Guide*.

Duplication Effects

The **Duplicate → Bridge** command in the Model module is a flexible duplication tool that allows you to duplicate or “bridge” a polygon mesh, patch, or NURBS surface object between two surfaces you specify. The surfaces may be polygon mesh, patch, NURBS surface, or curves, and you can duplicate the model at each vertex, at each tagged vertex, or randomly in a specified number of times.

The **Duplicate → Duplicator** command in the Model module allows you to create multiple copies of a single object and position them on a selected surface (polygon mesh, patch, NURBS surface, or curve). You can specify that the duplicates be distributed randomly on the surface a given number of times, or be located at each vertex or tagged vertex. The orientation of the duplicates can be fixed or aligned with the normals, and the scale, rotation, translation, and shape can all be independently jittered in x, y, and z to create a more random effect.

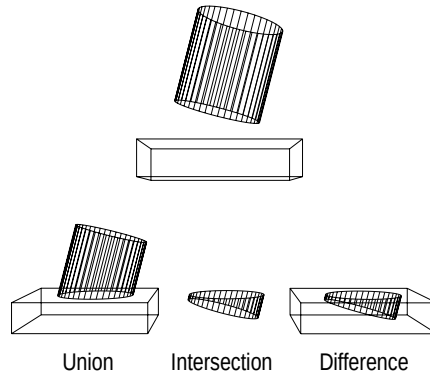
The **Duplicate → VolumeDuplicate** command in the Model module allows you to duplicate a model a specified number of times within a polygon mesh that you define. The locations of the duplicates within the polygon mesh can be randomly determined or placed at regular intervals along the three axes, and the size, shape, and orientation of the duplicates can all be independently jittered.

The **Duplicate → CopyCenter** command in the Model module allows you to copy the location of a source object’s centre to the location of a destination model’s centre.

Performing Boolean Operations

Boolean operations can be performed on polygon mesh objects or patch or NURBS surface objects.

The result of a Boolean operation is always a polygon mesh object.



To create a Boolean object:

1. Create or select two objects. Make sure that they are positioned so that they overlap.
2. Select one of the objects in the Boolean operation.
3. Choose the **Boolean** ➔ **Static** command. The Boolean Operations dialogue box is displayed.
4. Select one of the three operations: Union, Intersection, or Difference.
5. Select an option for the **Resulting Object**: A and B together or separate, or A or B only.
6. Select one of the options for Material, if you like.
7. Click Ok to accept the settings or Cancel to exit the dialogue box.
8. Pick the second object used in the Boolean operation.

One or two Boolean objects are created, depending on the options you chose (for **Resulting Object**).

In the Schematic window, any Boolean object is labeled “Bool*n*”.



Boolean operations are performed on two objects. If you want achieve a union of multiple objects, use the **Effect → Merge** command. The resulting object is also a polygon mesh.

Tips for Using Boolean Operations

- Make sure that the normals of both objects are correct before you attempt a Boolean operation. Boolean operations are based on the way normals were computed when the object was created; the Boolean operations recompute the normals in such a way that it is no longer possible to recover the original information on object normals later. For more information on different options for computing normals, see *Polygon Info* on page 879 of the *Reference Guide*.
- Problems sometimes occur when points are close together (numerical inaccuracy). Scaling objects up by a factor of 10 or 20, freezing them, then scaling them down can make a difference in the quality of the result.
- If the object is complex or has internal curves, holes, etc., scale the centre of the object down as much as possible. In many cases, the results are cleaner.
- Start from simple to complex. For example, if you want a curved surface with holes, grooves, etc., follow these steps:
 - Start with a flat shape that has as few polygons as possible.
 - Cut the required bits away (for multiple holes, use a single merged cutting object).
 - Bevel or round the result if necessary.
 - Subdivide this as much you need to.
 - Patch or curve-fit the result onto a mould object.

Animating Boolean Operations

You can animate the Boolean operation using the **Boolean → Animation** command, as described on page 112 of the *Reference Guide*, and creating keyframes.

In this example, one (invisible) object moves through another object and takes a chunk out of it as it goes by:

1. Keyframe the movement of the moving object by choosing **SaveKey → Object → Explicit Translation → All**.
2. Create a Boolean relationship:
 - Position the moving object so that it overlaps the static one.
 - Choose the **Boolean → Animation** command and select a Boolean Difference operation.
3. When you replay the animation, the moving object takes a chunk out of the static object at any point where they intersect.

Using Custom Effects for Boolean Operations

To perform Boolean operations on two closed linear curves, you can use the **Draw → Boolean2D** command in the Model module. You can perform the traditional union, difference, and intersection Boolean operations in any of the three drawing planes (xy, xz, or yz) to create complex shapes from simple curves. The resulting object is a closed linear curve or a hierarchy of linear curves. For more information on using this command, see its description in the *Reference Guide*.

Slicing a Hierarchy

The **Effect → Slice** command in the Model module allows you to use a grid to perform a Boolean-like slicing operation on a hierarchy. A new model is created consisting of all the portions of the hierarchy on one side of the grid, and also the cross-sectional pieces. These may be merged into the corresponding model pieces, or they may be created in a separate hierarchy. Moreover, the cross-sectional pieces may inherit the materials and textures of the grid or of the models. With Slice, you can easily obtain cross-sectional views of complicated hierarchies, such as a building or a piece of machinery.

The **Effect → MultiSlice** command in the Model module is a variation on Slice that lets you slice a hierarchy by multiple slicing planes at once. The MultiSlice effect is computed only for those models in the hierarchy that are polygon meshes. Models that aren't

polygon meshes are replaced by null objects in the resulting hierarchy. If you want to slice through non-polygon mesh objects such as patches or NURBS surfaces, they must first be converted to polygon meshes.

For more information on using these effects, see their command descriptions in the *Reference Guide*.

Editing Points

Editing points on an object is the most basic way of changing its geometry. The first step to editing points is usually to tag (select and highlight) them for subsequent operations.

Adding and Deleting Points

Using the commands in the **Edit** menu in the **Model** module, you can add or delete points on curves, polygon mesh objects, and surface objects. The procedures vary depending on the type of objects. For more information, refer to the respective chapters on each object type.

Tagging and Untagging Points

Before you perform a modelling operation, it is often useful to identify the points or area which you want to edit. You can do this by “tagging” selected points.

Once you tag points on an object, they are highlighted in red when the object is selected. If you are in **TAG** mode, transformations are applied only to the tagged points and the rest of the object is unaffected.

1. Select an object.
2. To display object points, make sure **Show ➔ Points** is selected.
3. Use the **t** Supra key and pick points or rows. You can either click directly on the points, or use a click-drag-release motion to sweep over an area of points.
 - Left mouse button tags selected points.
 - Middle mouse button untags selected points.
 - Right mouse button toggles the selection.

You can also choose one of the **Tag ➔ Points** subcommands:

- **Point only** selects only the point you pick in the next step.
- **Row in U**, **Row in V** selects the row of points (in the specified direction) closest to your mouse selection.

Note

The **Row in U** and **Row in V** commands apply only to patch or NURBS surface objects. On other objects, these commands have the same effect as choosing the **Point Only** command.

4. When you release the mouse, the tagged points are highlighted in red.

Editing Tagged Points

In TAG manipulation mode, scaling, rotation, and translation commands are limited to tagged points only. For example, if you choose any of the **Rotate** menu cells, the rotation is applied to the whole group of tagged points.

In OBJ manipulation mode, the **Edit** commands (add, delete, move, and so on) can apply to tagged and untagged points alike. You can also scale, rotate, and translate the whole object.

However, in OBJ mode, you can still limit a Move Point operation only to tagged points on an object using the **Edit → Move Tag Only** command. After choosing this command, the next time you use the **m** Supra key or the **Edit → Move Point** command, you can only move a tagged point.

Such a restriction can be useful when editing clusters, which prevents inadvertently moving untagged points while attempting to move the tagged points.

Creating Clusters

You can tag a group of points, called a *cluster*, save the group by a name, and then edit these points as a group. By giving the clusters a name, you can later select one from the Cluster List to model that group.

Clusters are useful for quickly deforming the shape of an object, as described in *Deforming with Clusters of Points* on page 308. Most modelling deformations can also be animated, as described in the *Animating User's Guide*.

1. Select an object and tag a group of points, using the **t** supra key.

Note

Points can belong to more than one cluster group.

2. Choose the **Tag → Set Cluster** command in the Model module. The Set Cluster dialogue box is displayed.
3. Enter the cluster's name in the **Name** text box.

4. If you want the cluster to have its own centre for purposes of deformation or animation, select the **Create Center** option.
5. Click **New** to create the cluster. The name of the new cluster is added to the Cluster List.
6. Click **Ok** to accept the settings or **Cancel** to exit without creating the cluster.

Tagging Using Custom Effects

You can also perform different types of tagging operations using custom effects. For more information on using these effects, see their command descriptions in the *Reference Guide*.

The **Tag** → **SplineTag** command in the Model module allows you to perform tag, untag, or reverse tag operations on all points of a model that are contained within the boundary of a curve that you create. This effect is particularly useful for tagging a group of points on models with many points.

The **Tag** → **SurfaceTag** command in the Model module allows you to perform tag, untag, or reverse tag operations on all points of a model that are contained within a polygon mesh that you create. This effect is particularly useful for tagging a group of points on models with many points. It is also a convenient tool for tagging points that are far apart from each other on a complex model.

Moving Points

To move one point at a time on any object (curve, polygon mesh, or surface).

1. Use the **m** Supra key or choose the **Edit** → **Move Point** command in the Model module.
2. Translate the desired control point.

Entering New Point Coordinates

You can also enter coordinate values for a point to move a point to a precise location in x, y, or z.

This can be useful for control points on NURBS curves or surfaces which, in addition to the XYZ coordinates, have a Weight parameter that can be edited to modify the shape of the curve at the point.

For information on editing point coordinates for curves, polygon mesh objects, and surface objects, refer to the respective chapters in this guide.

Using Proportional Editing

You can define a falloff area around any point and define the way that area responds when a point is moved. When you move a single point, this also moves some of the points that surround it as well.

1. Select an object and choose either the **Edit → Proportional** or the **Edit → Proportional Setup** command in the Model module.
2. If you choose **Edit → Proportional**, the current settings of the Proportional operation are applied during the next move.
3. If you want to edit these settings, choose **Edit → Proportional Setup**. The Proportional dialogue box appears, in which you define both the falloff area around any point and its response to editing.

Settings in this dialogue box apply globally to all objects in the scene.

For more information, see the **Edit → Proportional Setup** command on page 417 of the *Reference Guide*.

Transforming Clusters

As described on page 43, you can create groups of tagged points called clusters. A cluster can be edited and transformed as a group, and the changes can be keyframed to animate the changes. Clusters can have their own centre. The effects of transformations differ according to whether you apply them to the cluster or to the cluster's centre.

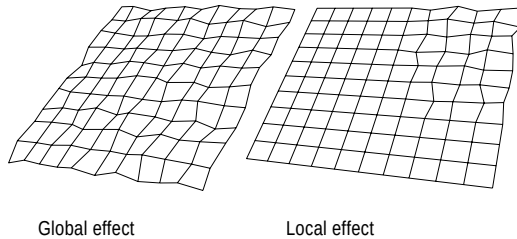
For details, see *Deforming with Clusters of Points* on page 308. For information on using cluster constraints, see the *Animating User's Guide*.

Applying Random Displacement

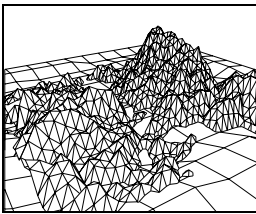
The **Effect → Randomize** command in the Model module is a quick editing technique in which you can randomly displace all or some points on a polygon mesh or surface object for a “bumpy” effect.

You can apply the effect to the whole object or tag points to apply the effect to a specific area. You can define the maximum displacement of points.

1. To apply the effect over the whole object, select the object for a global effect.
2. To apply the effect to a specific area of the object for a more local effect, tag points on the object and choose the TAG manipulation mode.
3. Select the **Effect** → **Randomize** command. The Randomize dialogue box is displayed.
4. Set the **Size** for each axis: range $[0, \infty]$. This is the maximum displacement of points along each axis, measured in SOFTIMAGE units.
5. Set the number of **Repetitions** of random calculation: range is from 0 to ∞ .
6. Click Ok to accept the settings or Cancel to exit the dialogue box.



Custom Effects for Creating Fractals



The **Polygon** → **Fractalize** command in the Model module allows you to apply fractal iterations to any polygon mesh object. For example, a realistic mountain, a meteor, or rolling countryside can easily be created by applying Fractalize to a simple object. You can control the number of iterations, the magnitude, the jaggedness of the resulting object, as well as several flags controlling the displacements created.

For more information on using this effect, see its command description in the *Reference Guide*.

Using Custom Effects to Align Objects

The **Duplicate → Align** command in the Motion module allows you to align elements of one model with the elements of another. You can align a point, edge, or polygon of one object to a point, edge, or polygon of another object. You can also choose to have the normals of the two objects line up where they are connected. The first object is automatically positioned according to the alignment information. In addition, the transformation of the icon serves as a convenient way for moving the aligned object: rotations pivot about the alignment in x and y and revolve about the alignment in z. Translations slide along the alignment in x and y and perpendicularly away from the alignment in z.

For more information on using this effect, see its command description in the *Reference Guide*.

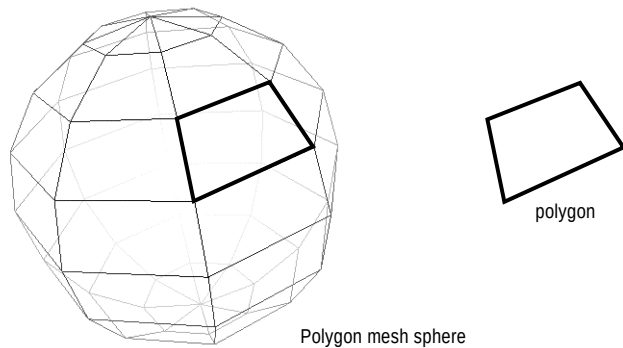
C H A P T E R T H R E E

Polygon Mesh Objects

Introduction

As the name implies, a polygon mesh object is an object made of *polygons*. A polygon is a closed 2D shape described in points (vertices), and enclosed by three or more straight line segments. These line segments intersect in pairs and only at the vertices. The number of vertices equals the number of line segments. The simplest polygon is a triangle made of three points.

To approximate a smooth object surface from straight lines depends on the number of polygons defined and also on the treatment of the surface normals (defined on page 7) within the rendering process.



Each polygon on the object may or may not be planar (flat). As you add and move vertices around in 3D space, you can create non-planar polygons. However, when objects are automatically triangulated for rendering, non-planar polygons are divided into triangles.

Creating a Polygon Mesh Object

In SOFTIMAGE 3D, you can create a polygon mesh object in several ways:

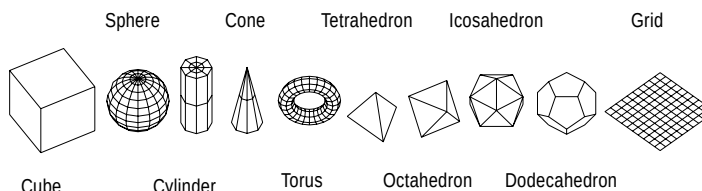
- Select a predefined 3D primitive in the **Get → Primitive** menu cell commands and make it of the object type “polygon.”
- Make your own custom polygon mesh object by drawing one polygon at a time.
- Draw curves, and extrude, revolve, or skin them to make a polygon mesh object. The resulting 3D object can then be further modified.

Polygon mesh objects are very versatile: you can add or delete polygons from them, add or delete vertices and edges, and apply a different material or texture to each polygon as you like.

Selecting a Predefined Primitive Object

In the **Get → Primitive** menu, the Cube, Cylinder, Cone, Tetrahedron, Octahedron, Icosahedron, and Dodecahedron commands are all polygon mesh by definition.

The Sphere, Torus, and Grid commands can also be created as polygon meshes.



Primitive objects

The Circle, Arc, Square, and Spiral commands are primitive curves, but when you extrude them using the **Surface → Extrusion** or **Surface → Revolution** command, the resulting 3D object can be defined as a polygon mesh.

1. Choose **Get → Primitive** and the appropriate command, such as Sphere or Icosahedron.
2. The related dialogue box appears, with one or more parameters to set the initial size or step (subdivision) value of the object. Fill in the parameters or accept the default values.

3. Click Ok to accept the settings or Cancel to exit the dialogue.

You can now modify this polygon mesh by adding, deleting, extruding polygons, and so on. Refer to *Modelling Selected Polygons* on page 74 for more information.

Drawing a Polygon Mesh

Two commands in the Model module allow you to create a custom polygon mesh object. As you draw new polygons, you must attach them to existing ones.

- **Draw → Polygon** represents the “drawing” mode. It creates new vertices to define the outline of a new polygon.
- **Polygon → Polygon** represents the “attaching” mode. It picks existing vertices of polygons to which the new polygon is attached.

You don’t have to switch between the two commands as you are modelling; the selections are toggled for you and the mode switching is handled seamlessly.

Drawing the First Polygon

1. To start the process, choose the **Draw → Polygon** command in the Model module. The status line says “Draw Mesh” and the mouse line prompts are:
 - Left mouse button adds a vertex after the previous one.
 - Middle mouse button adds a vertex between two existing ones.
 - Right mouse button attaches the new polygon.
2. Create the initial polygon. Continue to left-click and middle-click until you have the desired polygon shape, then right-click to finish.

When you right-click, you enter the “add polygon” mode in which you start adding another polygon to the first. This corresponds to choosing the **Polygon → Polygon** command.

Adding Polygons

After creating the initial polygon, you are ready to add more. The status line says “Add Polygon.” The new mouse line prompts are:

- Left mouse button picks a vertex on an existing polygon to which any new polygon will be attached.

- Middle mouse button accepts the current polygon. The status line message reads “The polygon has been added to the mesh.”
 - Right mouse button ends the additive mode. The status line message reads “Polygon creation aborted.”
3. Left-click on at least two vertices on the object. Picked vertices are highlighted in yellow. The status line says “Select vertex #*n* of the polygon.”
 4. If you left-click anywhere off the object, you return to the “draw” mode because you are now defining a vertex for a new polygon.

If you left-click on another existing vertex of the object, you remain in “attach” mode. The new polygon (when completed) will also be attached to this vertex of the object.

5. Keep picking or drawing vertices off in space to make a polygon of the desired shape.
6. To complete the polygon, middle-click. The completed polygons is added to the polygon mesh.
7. Repeat the operation until you have the desired polygon mesh object.
8. When satisfied with the object, right-click.

Rules for Adding Polygons

- You can only add a polygon to an open edge. To help you see which edges are open, choose the **Show → Edge Flags** command. When you activate this command, open edges are green, and closed edges are white.
- To keep the same surface normals, you must draw the new polygon in the same direction that the existing ones were drawn. In most cases, this means drawing *counterclockwise* to keep surface normals pointing outward.

The error message “Bad edge orientation” is displayed if you try to add polygons onto a closed edge, or pick the vertices in the wrong direction for surface normals.

- When filling in an empty polygon, pick *all* of its vertices. For example, pick all four vertices of an empty polygon on a cube.

- To add a new polygon onto the edge(s) of existing polygon(s), pick the vertices of the appropriate edges. You must pick at least two existing vertices before adding a new one off the polygon

For more tasks related to editing polygon meshes, see *Editing Polygon Mesh Objects* on page 61.

Making a Curve into a Polygon Mesh Object

You can draw a curve profile using any kind of curve with the **Draw** → **Curve** menu commands in the Model module.

You can then extrude or revolve the curve into a 3D object, or skin over several curves using the **Surface** commands. The resulting object can be a polygon mesh, or a patch or NURBS surface.

For more information, see *Making 3D Objects from Curves* on page 130.

Converting Objects to Polygon Mesh

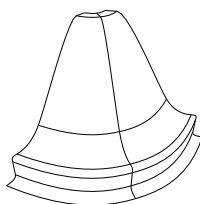
You can convert patch or NURBS surfaces, meta-clay elements, or faces to polygon mesh objects. This is useful if you need to model spiky-looking objects or export a model to a platform which accepts only polygon mesh objects. Also, many custom effects, such as Explode or Breakup, work only on polygon mesh objects.

Note

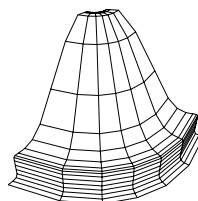
You can only convert polygon mesh objects to surface objects using the **Web** effect described in *Optimizing Models* on page 317 or the **Deformation → ShrinkWrap** effect described in the *Reference Guide*.

Patch or NURBS Surfaces to Polygon Mesh

The **Effect → Convert** command in the Model module allows you to create a polygon mesh object from a selected patch or NURBS surface.



NURBS surface



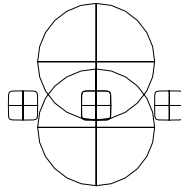
Polygon mesh

1. Select a patch or NURBS surface object.
2. Choose the **Effect → Convert** command.
 - A NURBS surface is converted immediately to a polygon mesh.
 - For patch surfaces, the Convert Patch dialogue box is displayed. Select accept the default **Polygon** option and click Ok. The object is converted to a polygon mesh object.

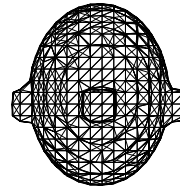
Meta-Clay to Polygon Mesh

The **Meta-Clay → Convert to Polygon** command creates a polygon mesh object from the selected meta-clay system. This allows you to model with the smooth, organic surface of a meta-clay object, but

speed up manipulation by converting it into a polygon mesh object when you're finished modelling.



Meta-clay system



Converted to polygon mesh

Tip

Before you convert, make sure the wireframe and rendering slice sizes are not too low. You can edit these by choosing the **Meta-Clay → Setup** command and setting their values in the Meta-Clay Setup dialogue box. See *Setting the Resolution* on page 213 for more information.

1. Select a meta-clay system or a meta-clay element.
2. Choose the **Meta-Clay → Convert to Polygon** command in the Model module.

The active system is converted to a polygon mesh object.

3. If the converted object has an excessive polygon count, you can optimize it greatly using the **Effect → Polygon Reduction** command as described in *Optimizing Models* on page 317.

For minor optimizations, you can also use the **Effect → Cleaup** command and select the **Merge polygons if angle smaller than** option.

Faces to Polygon Mesh

The **Effect → Convert** command in the Model module also allows you to create a polygon mesh object from a selected face.

1. Select a face.
2. Choose the **Effect → Convert** command. The Convert Face dialogue box is displayed.
3. Set a sampling value for the new object in the **Step** text box.
4. Click Ok to accept the default **Polygon** option, or Cancel to exit the dialogue box.

The original face remains available, and the new polygon mesh object is created.

Merging Objects

The **Effect → Merge** command allows you to create a single object from a group of objects which are not necessarily of the same type, such as merging a polygon mesh with a patch or NURBS surface.

This operation is similar to a Boolean union. In both operations, all objects are converted to polygon mesh objects and the original objects are preserved.

While **Effect → Merge** does not remove unnecessary edges and vertices in the resulting polygon mesh model, a Boolean union does. On the other hand, Boolean operations are limited to two objects, but the **Effect → Merge** command can merge multiple objects.

1. Choose the **Multi** menu cell to select multiple objects.
2. Select all of the objects to be merged.
3. Choose the **Effect → Merge** command. The Merge dialogue box appears.
4. To manually optimize the resulting object, you can choose merge points which are closer than n SOFTIMAGE units by specifying the value for the **Merge near points if distance less than** option.
5. Click Ok to accept the settings or Cancel to exit the option.

While the resulting object can be scaled, rotated, and translated as one object, the operation does not bring the component objects any closer on the screen or create any merge/blend portion between them. Contrast this with the **Draw → Merge Surfaces** command in which patch or NURBS surfaces are combined to form one connected surface.



Effect → Merge is useful for combining objects modelled as mirror images of each other. For example, you could model half an object and use **Effect → Symmetry** to mirror it. Then use the **Effect → Merge** command on the two symmetrical halves and choose **Effect → Cleanup** to eliminate any “messy” edges, if necessary.

Custom Effects for Merging Objects

The **Polygon → MergeGeometry** command in the Model module allows you to merge the geometry of any curve with that of an arbitrary polygon mesh object. You simply model the curve and the polygon mesh and then determine the projection plane in which the merge will take place – the vertices are added automatically.

The **Polygon → Stitch** command in the Model module allows you to create seamless joints between free or open edges of polygon meshes or curves by automatically creating joining triangles between nearby vertices of participating models. When stitching polygon meshes, you simply tag the portion of the free edges to stitch together. Difficult modelling tasks, such as joining a circle to a square, or connecting fingers smoothly to a hand, are easily performed using this effect.

The **Polygon → Weld** command in the Model module is used for merging nearby vertices of a polygon mesh. Weld identifies all vertices that are within a user-defined threshold from each other, even if they belong to different polygons.

The **Effects → Shoe** command in the Motion module allows you to glue one entire hierarchy to another hierarchy. This can be useful for filling in gaps between models in a hierarchy, such as the fingers and palm of a hand.

The **Effects → Glue** command in the Motion module allows you to “glue” tagged points of one object to the tagged points of another surface object. You can apply the Glue effect to polygon meshes, patches, NURBS surfaces, or curves. Glue is a motion effect, so either of the models can be animated, and the glued portions will stick together. Falloff parameters allow the effect to be applied smoothly to nearby points.

The **Draw → PolySkin** command in the Model module is an extrusion tool that can skin between any two polygon mesh objects if that they have the same number of points and polygons.

For more information on using these effects, see their command descriptions in the *Reference Guide*.

Cutting Along a Plane

The **Effect → Plane Clipping** command allows you to cut a polygon mesh at the origin along the x-axis, just as you might slice the top off a coconut, leaving it open.

This command also adds control points to a Linear, Bezier, Cardinal, or B-Spline curves along the global intersection plane you specify. For more information, see *Adding Points along a Plane* on page 125.

1. Select a polygon mesh object.
2. In the Front or Right window, position the object so that the portion you want to cut away is above the cutting plane, which is at the global origin (0 on the x-axis).
3. Choose the **Effect → Plane Clipping** command.

The clipping is performed immediately.

Editing Polygon Mesh Objects

To edit the shape polygon mesh objects, you can add or delete polygons, vertices, or edges. You can increase the resolution of the object, bevel the edges, optimize the number of polygons, or reorder them.

Selecting Polygons on an Object

Whatever manipulation mode you are in (OBJ, POL, etc.), you can select polygons on a polygon mesh using commands in the **Polygon** menu in the Model module.

When you select polygons, you can transform, delete, or extrude them, and so on, to produce some cool results starting from simple polygon mesh objects. For more information and procedures, see *Modelling Selected Polygons* on page 74.

In any manipulation mode:

1. Select a polygon mesh object.
2. If you choose **Polygon → Select by Raycasting** or the **g Supra** key, simply click on the polygon(s) anywhere on the object.
3. If you choose **Polygon → Select by Rectangle** or the **y Supra** key, drag the mouse to surround one or more polygons: the selection is highlighted when you release the mouse button.
4. If you choose **Polygon → Select by Tag Vertex**, tag the vertices of the polygon(s) you want to select. To highlight the selection, click anywhere within the selected polygons.

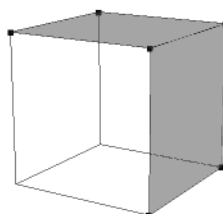
If you want to select non-adjacent polygons, untag the vertices of the first selected polygon before repeating the procedure for the next.

5. Whatever the selection method, use the mouse buttons as follows:
 - Left mouse button selects the polygon.
 - Middle mouse button deselects the polygon.
 - Right mouse button toggles the polygon selection.
6. Press Esc to exit the mode.

Whichever selection method you use, selected polygons are highlighted in magenta, which makes your selection easier to see.



One polygon selected.



Polygon pair selected.

You can select non-adjacent polygons or adjacent ones (which share an edge). However, if you are in POL mode, adjacent polygons share a common coordinate system, which is the average of the individual coordinate systems of each polygon.

Transformations on Adjacent Polygons

Assuming you are in POL manipulation mode and select LCL (local) transformation mode:

- If you were to transform adjacent polygons, they would be transformed according to their shared coordinate system.
- To manipulate adjacent polygons independently, you need to give them separate centres. See *Transforming Polygons* on page 75 for more information.



At this point, if you want to transform, delete, or extrude only the selected polygons, change to POL manipulation mode, described in *Modelling Selected Polygons* on page 74.

Selecting by Raycasting

When you choose the **Polygon → Select by Raycasting** command (or the **g Supra** key), you can simply click to select a polygon. Keep pressing the mouse to select adjacent polygons or release and click on another, non-adjacent polygon.

Unlike the **Polygon → Select by Tag Vertex** and **Select by Rectangle** commands, this command selects only the polygon nearest the camera. As a result, you never inadvertently select the polygon directly behind your selection, on the other side of the 3D object.

Note

This assumes that the polygon's normals are facing the camera. If the object is inversed, it is the polygon on the opposite side from the camera which is selected.

Selecting by Tag Vertex

The **Polygon → Select by Tag Vertex** command allows you to select polygons by tagging their vertices and then clicking on the polygon.

This method often selects both the intended polygon and the one directly “behind” it on the other side of the 3D object. Check that only the intended polygon is selected before starting to model with it.

Selecting by Rectangle

The **Polygon → Select by Rectangle** command (or the y Supra key) allows you to draw a rectangle around one or more polygons.

As with selecting by tagging vertices, this command often selects the intended polygon and the one directly “behind” it on the other side of the 3D object.

Adding Polygons

If you want to draw new polygons on an existing polygon mesh object, you must first attach them to existing ones. Using the **Polygon → Polygon** command, you can pick existing vertices of polygons to which the new polygon will be attached.

A related command, **Draw → Polygon**, represents the “drawing” mode. It creates new vertices, to define the outline of a new polygon.

Note

You don't have to switch between the two commands while modelling; the selections are toggled for you and the mode switching is handled seamlessly.

For the procedure for creating your own polygon mesh object, see *Drawing a Polygon Mesh* on page 53.

Rules for Adding Polygons

You can either:

- Fill the “empty” space of a deleted polygon.

or

- Add a new polygon off an existing one. The existing polygon must have an open edge.

For either case:

- You can only add a polygon to an open edge. To help you see which edges are open, choose the **Show → Edge Flags** command. When you activate this option, open edges are green and closed edges are white.
- To keep the same surface normals, you must draw the new polygon in the same direction that the existing ones were drawn. In most cases, this means drawing *counterclockwise* to keep surface normals pointing outward.

The error message “Bad edge orientation” is displayed if you try to add polygons onto a closed edge, or pick the vertices in the wrong direction for surface normals.

Procedure

1. Starting from an existing polygon, choose the **Polygon → Polygon** command in the Model module.

The status line says “Add Polygon.” The mouse line prompts are:

- Left mouse button picks a vertex on an existing polygon to which any new polygon will be attached.
 - Middle mouse button accepts the current polygon. The status line message reads “The polygon has been added to the mesh.”
 - Right mouse button ends the additive mode. The status line message reads “Polygon creation aborted.”
2. Left-click on at least two vertices on the object. Picked vertices are highlighted in yellow. The status line says “Select vertex #*n* of the polygon.”
 3. If you left-click anywhere off the object, you return to the “draw” mode because you are now defining a vertex for a new polygon. If you left-click on another existing vertex of the object, you remain in “attach” mode. The new polygon (when completed) will also be attached to this vertex of the object.
 4. Keep picking or drawing vertices off in space to make a polygon of the desired shape.
 5. To complete the polygon, middle-click. The completed polygons is added to the polygon mesh.
 6. When satisfied with the object, right-click to end the mode.

Using Custom Effects for Adding Polygons

You can add triangular polygons to a polygon mesh object using the **Polygon → Mosaic** command in the Model module. This effect is useful for performing subsequent shape animation operations, and when using the Cometa effect. The triangles of the source model are cut in two according to size. The largest triangles are cut first, creating two new polygons that substitute the original.

The **Polygon → Coplanar** command in the Model module adds border polygons around the edges of a polygon mesh's polygons. It is used to give the appearance of rounded edges without complex rounding operations, and is often used as a kind of bevelling tool.

For more information on using these effects, see their command descriptions in the *Reference Guide*.

Deleting Polygons

You can select one or more polygons on a polygon mesh and then delete them.

1. In any manipulation mode, select one or more polygons (see *Selecting Polygons on an Object* on page 61).
2. To delete the selected polygon(s), you must switch to POL manipulation mode and then choose **Delete → Selection**.

If you do not switch to POL mode, you will delete the entire selected object.

Custom Effect for Eliminating Polygons

The **Polygon → Eater** command in the Model module allows you to eliminate polygons from a complex polygon mesh object or hierarchy to speed up the screen display and reduce memory demands. It leaves holes where the polygons have been removed.

For more information on using this effect, see its command description in the *Reference Guide*.

Adding and Deleting Vertices

You can add or delete vertices on a polygon mesh using commands in the **Polygon** menu and in the **Edit** menu in the Model module.

Polygon Menu

The **Polygon → Vertex** command in the Model module allows you to add or delete vertices on a polygon mesh object. When adding vertices, you cannot drag them to a different position.

1. Select a polygon mesh object.
2. Choose the **Polygon → Vertex** command and use the mouse buttons as indicated:
 - Left mouse button adds a vertex at the point you pick, when you pick an edge.
 - Middle mouse button adds a vertex in the middle of the picked edge.
 - Right mouse button deletes the vertex you pick.
3. To end the mode, press the Esc key.

Edit Menu

You can also use the **Edit → Add Point** or **Delete Point** commands.

Edit → Add Point differs from **Polygon → Vertex** in two ways:

- As you click to add the point (vertex), you can drag it to another spot to modify the shape of the polygon using **Edit → Add Point**. This allows you to create some oddly shaped polygons, which can be good or bad depending on what you want to do.
- This command also allows you to click outside the polygon mesh object and (inadvertently) add a point out in space, so it is important to end the adding mode at the appropriate moment.

1. Select a polygon mesh object.
2. Choose the **Edit → Add Point** command.
3. When adding points (vertices), use the mouse buttons as follows:
 - Left mouse button adds a point after the last point.
 - Middle mouse button adds a point between points.
 - Right mouse button adds a point before the first point.

Note

As the point is added, you can drag it to another spot to modify the shape of the polygon.

4. To delete points (vertices), choose the **Edit → Delete Point** command. Any mouse button deletes the selected point.
5. To end the add or delete mode, press the Esc key.

Custom Effects for Creating an Equal Number of Vertices

The **Polygon → Equalize** command in the Model module adds vertices to polygons on a polygon mesh objects, so that all polygons in that object have the same number of vertices. You can add vertices anywhere on the object, not just the current polygon. This effect is particularly useful when used in conjunction with the Explode and Polyskin effects on objects such as spheres or other polygon meshes that contain polygons with different numbers of vertices.

For more information on using this effect, see its command description in the *Reference Guide*.

Editing Vertices

You can edit vertices on a polygon mesh using various techniques, such as selecting and translating an individual vertices, moving tagged groups of vertices, or applying random displacement to vertices.

Most of these techniques are applicable to all object types. For more information, see *Editing Points* on page 42 of this guide.

Adding and Deleting Edges

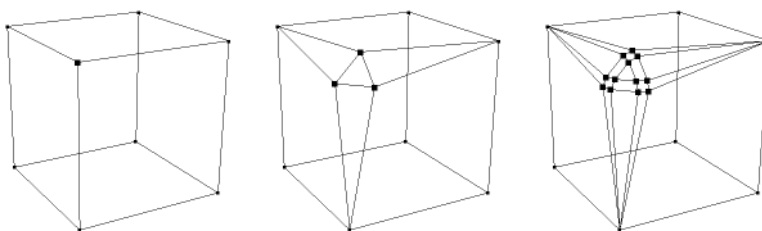
You can add or delete edges on a polygon mesh using the **Polygon → Edge** command in the Model module.

1. Select a polygon mesh object.
2. Choose the **Polygon → Edge** command and use the mouse buttons as indicated:
 - Left mouse button adds a single edge between two vertices. Pick the first vertex, then the second.
 - Middle mouse button allows you to add a series of connected edges by picking the first vertex, picking the second to create the edge, then picking a third to create an edge between the second and the third, and so on.
 - Right mouse button removes the edge that you pick.

Bevelling Edges

The **Effect → Bevel** command allows you to apply bevelling to all of a selected polygon mesh object, or locally to a tagged portion of the object. The bevel is applied in SOFTIMAGE units, and is calculated from the normals to the edges of the object.

1. Select a polygon mesh object.
2. If you wish to bevel only part of the object:
 - Tag vertices to define the area where bevelling is to be applied
 - Make sure to select TAG manipulation mode.
3. Choose the **Effect → Bevel** command. The Bevel dialogue box is displayed.
4. Set the width of the **Bevel** in SOFTIMAGE units.
5. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.



Local bevelling. First bevel on one point was set at 0.8. Second bevel was applied to three points at a factor of 0.15 (cube length = 5).

A similar effect, called rounding, is described next.

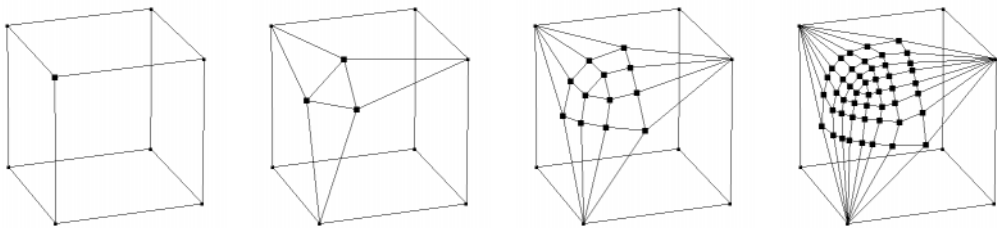
Rounding Edges

The **Effect → Rounding** command allows you to apply rounding to the edges of a whole polygon mesh object or part of it. The rounding factor is a fraction of the distance of the polygon vertices from the centre of each polygon.

For example, if use 0.2 as the rounding factor on a cube, the resulting edges is 0.8 the length of the original edge ($1.0 - 0.2$).

1. Select a polygon mesh object.

2. If you wish to apply rounding to only part of the object:
 - Tag vertices to define the area where rounding is to be applied
 - Make sure to select TAG manipulation mode.
3. Choose the **Effect → Rounding** command. The Round dialogue box is displayed.
4. Enter the desired rounding value (fraction) in the **Round** text box. Values can be between 0 and 1.
5. Click Ok to accept the settings or Cancel to exit the dialogue box.



Local rounding on one point of a cube. New points were kept selected for each successive transformation. (Rounding factor was set to 0.5 for this cube of 5 SOFTIMAGE units.)

Custom Effect for Engraving Bevels

The **Surface → PolyBevel** command in the Model module lets you engrave bevels into a polygon mesh object. PolyBevel engraves a user-tagged region of a polygon mesh and bevels the outline of the region with a user-defined bevel. This is especially useful in conjunction with MergeGeometry effect, which allows you to merge the geometry of a curve into a polygon mesh. You can also use it in conjunction with the SplineTag effect to simplify the process of tagging the interior of a region created this way.

For more information on using this effect, see its command description in the *Reference Guide*.

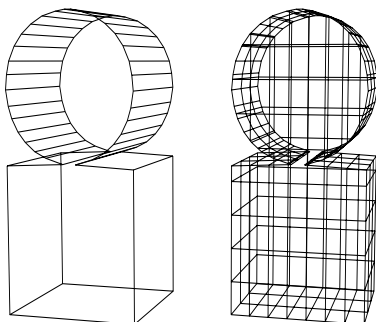
Subdividing a Polygon Mesh Object

The **Effect → Subdivision** command allows you to subdivide all polygons of a polygon mesh object, thus increasing the overall resolution of the object.

When you subdivide polygon meshes, the effect is to add edges; for instance, out of 1 polygon with 4 edges, you get 2 polygons and 7 edges.

This is especially useful for deformation or lattice operations.

You can also subdivide NURBS curves as described on page 125, and NURBS surfaces as described on page 179.



Subdivisions
x = 6, y = 10, z = 2

1. Select a polygon mesh object.
2. Choose the **Effect → Subdivision** command. The Polygon Subdivision dialogue box appears.
3. Set the number of **Subdivisions** desired for each axis (X, Y, and Z).
4. Click Ok to exit the dialogue box or Cancel to exit without accepting the settings.

Triangulating a Polygon Mesh Object

You can convert a quad polygon mesh object to a triangulated polygon mesh object (one with only triangular polygons).

1. Select a quad polygon mesh object.
2. Choose the **Effect → Convert** command.

The polygon mesh object is immediately triangulated.

Reordering Polygons

The **Effect → Order** command automatically reorders the polygons of a polygon mesh object to make the direction of all normals consistent.

To do this, it compares each polygon to its neighbours starting with the first polygon, and reorders the vertices on any “troublesome polygons” to make them consistent with their neighbours. If there is a conflict that creates a collision between polygons, the appropriate point or edge is duplicated.

This command is especially useful for objects converted from other file formats (such as DXF, Wavefront, or Vertigo), since not all systems calculate front/back faces or apply the same “right hand rule” to the orientation of the object’s normals.

It can also be useful when you have added polygons to an object, and introduced some polygons whose normals are inconsistent with the rest of the model.

1. Select an object.
2. Choose the **Effect → Order** command.

The vertices are reordered immediately.

Custom Effect for Reordering Polygons

You can also use the **Polygon → PolyOrder** command in the Model module to define the order of the individual polygons of a model. The polygons can be ordered randomly, or sequentially along each axis. PolyOrder is useful in conjunction with other effects such as PolyGrow.

For more information on using this effect, see its command description in the *Reference Guide*.

Optimizing the Number of Polygons

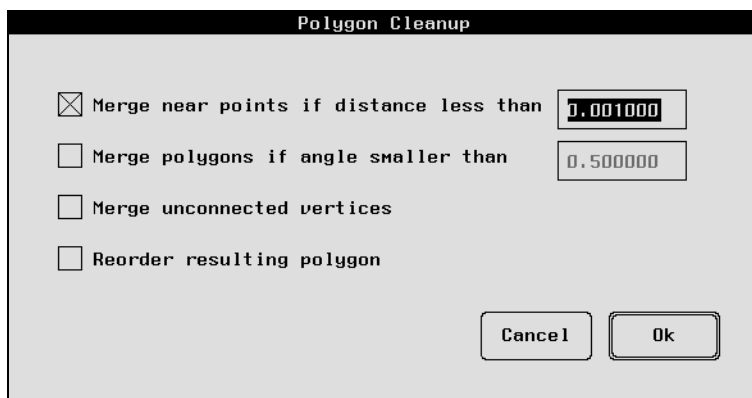
The **Effect → Cleanup** menu command optimizes the number of polygons on a polygon mesh object by removing unnecessary edges and vertices, such as when polygons are coplanar.

Note

This command is a fairly basic optimization technique. For more sophisticated optimization options, refer to *Optimizing Models* on page 317.

1. Select a polygon mesh object.
2. Choose the **Effect → Cleanup** command.

The Polygon Cleanup dialogue box is displayed.



3. Set the desired cleanup options and values to merge and reorder as you like. The parameter descriptions are found in the **Effect → Cleanup** command on page 441 of the *Reference Guide*.
4. Click Ok to accept the settings or Cancel to exit the dialogue box.

Precision Modelling on Polygon Mesh Objects

The **Polygon → Reference Frame** command creates a reference null object for a selected polygon. A reference null has its coordinate system aligned with that of selected polygon(s). Creating a reference null allows you to easily align a polygon to a certain plane for modelling.

1. Select a polygon mesh object
2. Select one or more polygons on the object. It is easiest to do this using the **g Supra** key or the **Polygon → Select Polygon by Raycasting** command.
3. Choose the **Polygon → Reference Frame** command. You need not be in POL mode to invoke this command.

A reference null object is created for each selected polygon.

Why Use a Reference Null?

You can use a reference null for a number of reasons, but the most obvious one is to assist you in modelling.

Since the reference null lines up with the centre and local axes of a polygon on a polygon mesh object, you can draw in precise reference to that centre.

For example, say you wanted to have a ball move along a path in the y-axis. In the Top window, select a polygon on the ball object to create a reference null with the interest on y-axis.

Draw the curve (the path) in the Front window; the curve begins exactly at the centre of the selected polygon and extends up its y-axis. You could then use path animation to make a ball or other object move along that path.

Instead of using the Top, Right, or Front windows, you could also choose the Ortho window, and simply choose the appropriate axis (using the X, Y, Z selections in the Ortho window title bar) to look at the model along any of the specified axes.

Using a reference null, you could easily apply constraints to it using the **Constraint** commands in the Actor or Motion module.

Custom Effects for Precision Modelling

You can also use the **Polygon → Wakeup** command in the Model module to bring a selected polygon of a polygon mesh object to one of the planes (xy, xz, yz) as shown in the parallel projection windows (Front, Top, Right, and Ortho). Because it is planar, it is easier to draw new elements, vertices, etc.

For more information on this effect, see its command description in the *SOFTIMAGE 3D Reference Guide*.

Modelling Selected Polygons

You can select and model individual polygons on a polygon mesh object. For example, instead of twisting an entire object, you can select a few polygons, duplicate and transform them in such a way as to apply a twist only to them, leaving the rest of the object unchanged. You can use this modelling technique to quickly model a Viking helmet with horns, for example.

For a quick example of using this feature, try your hand at the tutorial on page 79.

Polygon Manipulation Mode

When you are in OBJ mode, translations and deletions apply to the entire object. However, in POL (Polygon) manipulation mode, any transformations, duplications, or deletions are applied only to the selected polygon(s).



In POL mode, transformations use the axes specific to the polygon.

Polygon Coordinate System

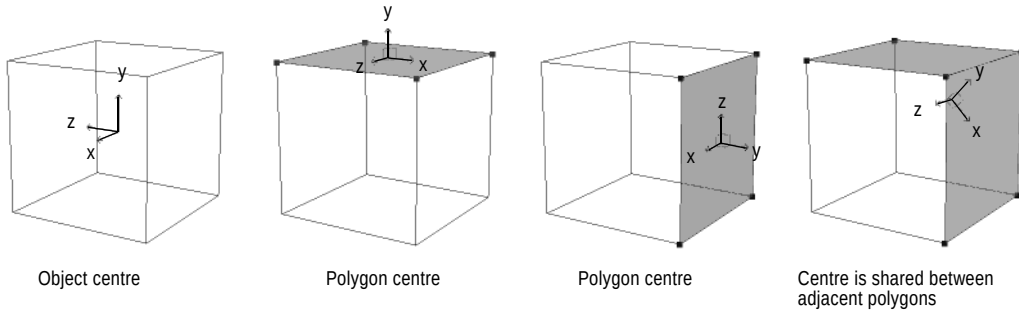
There is a coordinate system which is specific to a polygon. In this coordinate system:

- The y-axis is always the normal of the selected polygon(s).
- The x and z-axes lie in the plane of the polygon and are perpendicular to each other.

These axes are visible when you are in POL mode and the **Show → Centre** command is activated.

To view the difference between a polygon's axes and an object's axes, get a primitive cube, make sure **Show → Centre** is selected, and select one polygon using **Polygon → Select by Raycasting**.

Keeping the same polygon selected, switch back and forth between POL and OBJ mode. Notice the difference in the representation of the centre when you change modes.



Notice that in POL mode, adjacent polygons share a single centre. If you transform them in LCL (local) mode, they are transformed according to this shared centre by default. To find out how to transform adjacent polygons independently, see *Transforming Adjacent Polygons* on page 76.

Selecting Polygons

Whatever manipulation mode you are in:

1. Select the polygon mesh object.
2. To select specific polygons on the object, choose one of the following commands:
 - Polygon → Select by Raycasting or the g Supra key
 - Polygon → Select by Rectangle or the y Supra key
 - Polygon → Select by Tag Vertex

If you need more information on using these commands, see *Selecting Polygons on an Object* on page 61.

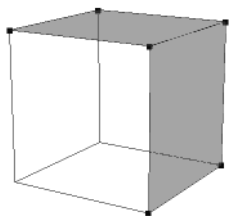
You can select adjacent or non-adjacent polygons. However, if you are in POL mode, adjacent polygons share a common coordinate system, which is the average of the individual coordinate systems of each polygon.

Transforming Polygons

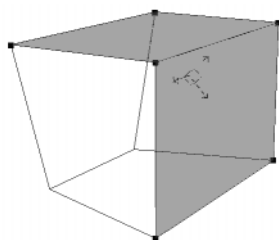
If you are in POL mode, all polygon transformations are performed according to the polygon's coordinate system and the selected transformation mode.

While you can select any transformation mode (LCL, GBL, and so on), you will most likely select LCL translation and rotation to transform the polygon according to its own local XYZ coordinate system.

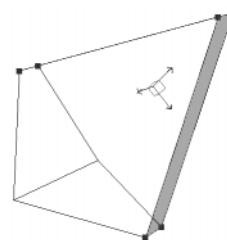
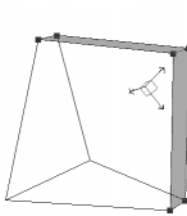
You can easily transform the selected polygon(s) which changes the shape of the whole polygon mesh object.



Transform the selected polygon pair to change the cube's overall shape.

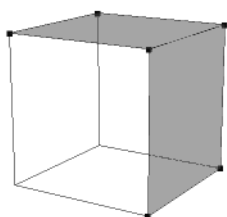


Scale in z direction (two examples above). Polygon expands and contracts along the shared coordinate system.

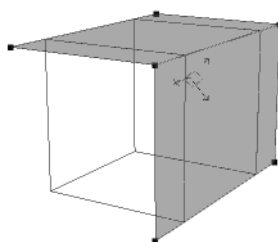


Transform again by scaling in y direction.

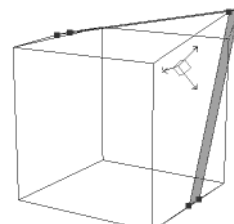
Alternately, you can duplicate the selected polygons *first* and then apply transformations to the duplicates. The purpose of doing this is to transform only the polygons while preserving the shape of the basic object.



Duplicate the polygon pair before transforming to preserve the cube's basic shape.



Scale in z direction (two examples above). Polygon expands and contracts while cube's original vertices remain.



Transform again by scaling in y direction and then translating in y.

Transforming Adjacent Polygons

If you select polygons which share an edge, they are said to be *adjacent* and share a common centre.

In POL manipulation mode (and assuming you are in LCL transformation mode), adjacent polygons are transformed according to a shared coordinate system by default, as illustrated on page 75.

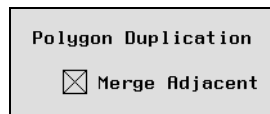
Creating Independent Centres

What if you want to transform and manipulate adjacent polygons *independently*, according to each one's separate coordinates? In that case, you have to give each polygon its own centre.

To do this, you must change the default Duplication Setup before duplicating the polygons.

1. Choose the **Duplicate → Setup** command. The Duplicate Setup dialogue box appears.

One parameter applies specifically to polygon duplication.



2. Deselect the **Merge Adjacent** option and click Ok to accept the settings.

Now as you duplicate polygons singly or repetitively, they revert to separate centres.

Duplicating Polygons for Transformation

The purpose of duplicating polygons is to allow you to transform them without affecting the shape of the rest of the object.

If you are in POL (Polygon) manipulation mode, the **Duplicate → Immediate** command makes a single duplicate of the selected polygons. To model a twisty shape extending from the selected polygon, you could perform successive duplications and transformations, but this would be very tedious.

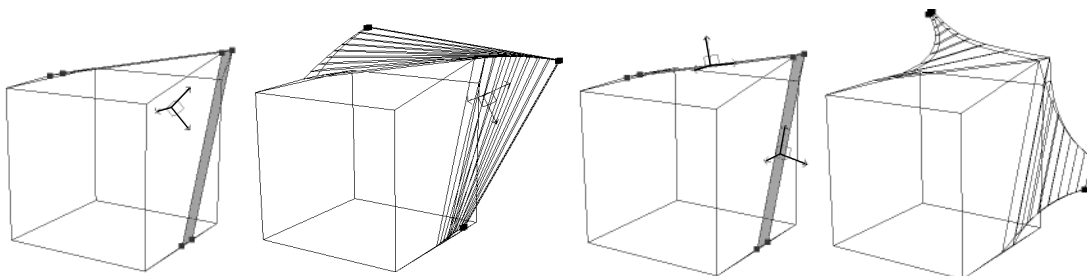
To automate the process, you can use either **Duplicate → Repetition** or a simple process of extruding the selected polygon along a curve. An example of each method is provided on page 79 and page 81.

Try it out!

1. Select the POL manipulation mode.
2. Select the object, then select the desired polygon(s) using the g Supra key.
3. To make one copy, choose the **Duplicate → Immediate** command, or use the d Supra key and pick the polygon to be duplicated.

4. To create multiple copies, choose **Duplicate → Repetition**. The Repetitive Duplication dialogue box appears, in which you can specify the number of duplications and successive transformations to be applied to each of the duplicates.

You can now manipulate and transform them separately.



Adjacent polygons are transformed according to the shared centre. To grow a fin, duplication with repetition was used (8 repetitions, scale z = 8, rotate z = -2, translate x = 0.6, translate y = 0.2, local mode, on a cube of length 4)

Deselect Merge Adjacent and then duplicate immediate to revert to separate centres. Grow two fins at once using duplicate with repetition (8 repetitions, scale x = 0.6, scale z = 0.8, rotate z = 4, translate y = 0.2).

Custom Effects for Breaking Up a Polygon Mesh

The **Polygon → Breakup** command in the Model module allows you to modify a polygon mesh object so that no vertex is referenced by more than one polygon. For example, a cube that has eight vertices and six polygons will have 24 vertices and six polygons after the Breakup effect is applied.

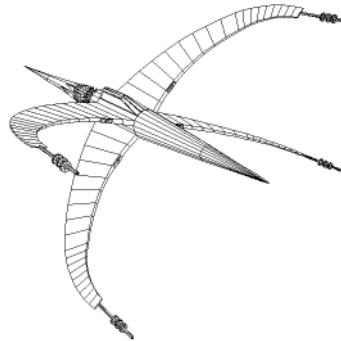
This allows you to manipulate individual polygons to “break up” the polygon mesh object. After a polygon mesh is broken up, you can transform the individual polygons. Normals at vertices are preserved so that the rendered image is the same after the Breakup effect is applied.

For more information on using this effect, see its command description in the *SOFTIMAGE 3D Reference Guide*.

Examples of Modelling with Polygons

Using Repetitive Transformations

This example is quick to model and lets you create a starship in a few minutes using some of the capabilities of polygon modelling.



The Body

1. Start with a primitive cylinder of:
 - Radius 1.0 and Height 5.0.
 - Longitude Step 22, Latitude Step 1, and Latitude Base 2
2. Rotate the cylinder 90 degrees in x and choose **Effect** → **Freeze Rotation**.

Setup

3. Make sure you are in POL mode and select the LCL translation and rotation modes.
4. Choose **Duplicate** → **Setup** and make sure that the **Merge Adjacent** option is selected in the Duplicate Setup dialogue box.

The Nose

5. In the Top window, select all polygons at the front of the cylinder (this is the end pointing downward in the Top window).
6. Choose **Duplicate** → **Immediate**. Then translate the selected polygons in y, and scale them in xyz to form a nose.
7. Deselect these polygons.

The Wings

8. In the Front window, select the polygons on the side of the cylinder at 2, 4, 8, and 10 o'clock.
9. Make sure you are in LCL transformation mode. Choose **Duplicate** ➔ **Repetition** and specify:
 - No of occurrences: 18
 - Scale: 0.9 in all axes
 - Rotation: 5 degrees in X
 - Translation: 1 unit in Y

The starship is starting to look like the real thing.

The Gun Turrets

At this point, the polygons are tagged at the end of the wings.

10. Duplicate once and scale in z to form a square shape.
11. Duplicate once more and translate in y. The result is gun turrets (or think up your own use for them!).
12. Repeat this procedure to add more detail.
13. When finished, deselect the polygons.

The Engine

14. In the Top window, select the end of the cylinder that hasn't been altered yet (this is the end pointing upward in the Top window).
15. Duplicate the selected polygons once (**Merge Adjacent** should still be selected in the Duplicate Setup dialogue box).
16. Translate them in y and then scale down in all axes.
17. Repeat this procedure to add more detail to the engine.
18. When finished, deselect the polygons.

The Canopy

19. In the Front window, select the polygons on the cylinder at 12 o'clock.
20. Duplicate once, translate in y, and then scale in x and z to form the canopy of the starship.

21. Translate the polygons in z to refine the shape and form the sloping windshield at the front of the canopy.

The Finished Product

And those are the basics of creating this model. Now try your hand at adding more detail on various parts of the ship.



When the desired modelling effect follows a complex path with random curvatures difficult to achieve through repetitive transformations, it is easier to extrude the polygon along a curve, as described in the following example.

Extruding Polygons along Curves

If you wanted to model a polygon along a twisting path, it is easier to draw the curve path, and then extrude the polygon along that curve, with or without adding transformations at each step of extrusion.



Relational modelling does not apply to polygon extrusion along a curve.

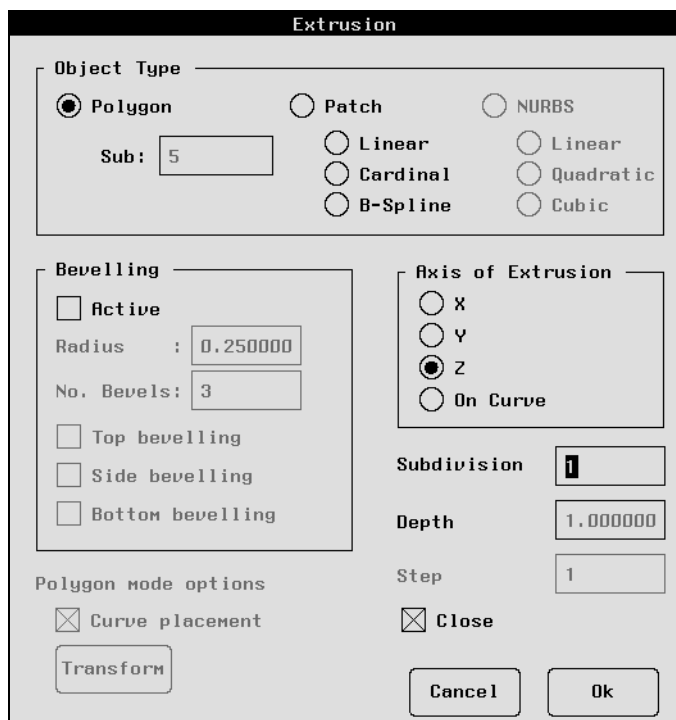
In general terms, this is the process you follow.

1. Draw the curve which serves as the extrusion path.
2. Select a polygon mesh object.
3. Select the polygon(s) to be extruded, using the **Polygon → Select by Raycasting** command or the **g Supra** key.
4. If you select non-adjacent polygons, they are extruded along the same curve path if **Curve Placement** is on.
5. If you select adjacent polygons, they can be transformed independently by choosing the **Duplicate → Setup** command and deselecting the **Merge Adjacent** option. For more information, see *Creating Independent Centres* on page 77.

Otherwise, adjacent polygons are transformed by default as one unit using a common coordinate system.

6. Switch to POL manipulation mode.

- Choose the **Surface** ➔ **Extrusion** command. The Extrusion dialogue box is displayed.



Only the options relevant to polygon extrusion can be selected (**Subdivision**, **Curve Placement**, and **Transform**)

- Set the **Subdivision** value. If you leave the default, the resulting extrusion appears “boxy” and rigid. By increasing the Subdivision value, the extrusion profile is smoother.
- If **Curve Placement** is selected, it doesn’t matter where you position the curve, just draw it going up the y-axis (Front window). The extrusion is performed as if the curve is lined up with the selected polygon(s) before extruding.

You can extrude several polygons along the same curve path. The curve is correctly reoriented for each extrusion.

- If **Curve Placement** is not selected, the extrusion takes place along the curve wherever the curve is positioned. For accurate results, you should draw the curve “in place,” directly out from the polygon.

You shouldn't extrude multiple polygons with this option deselected because the results would be unpredictable.

For more information and tips on drawing and placing the extrusion curve, see the next section, *Automatic Curve Placement*.

11. If you wish to add transformations in addition to extrusion (for example, to create a "twist and taper" effect at the same time), click the **Transform** button.
12. In the Extrusion Transformations dialogue box that appears, set the transformation values for the X, Y, Z local coordinates of the extruded polygon to add transformations at each step of the extrusion along the curve.
13. Click Ok to accept the transformation values and return to the Extrusion dialogue box.
14. Click Ok to apply the settings or Cancel to exit the dialogue box.
15. Pick the target curve.

The polygon is extruded along the curve path. Although the path is always that of the extruding curve, the result can look different depending on whether you selected **Curve Placement** or not. For more information, read on.

Automatic Curve Placement

You will usually want to extrude the polygon straight out from the object, using the polygon's own axes as the frame of reference and the curve as the path of extrusion. To achieve this, you would probably select **Curve Placement**.

The selected polygon(s) are extruded as if the target curve was first "placed" or lined up to the axes of the selected polygon(s) and then the extrusion is performed along the curve.

Here's how it works:

1. The beginning of the curve is lined up with the centre of the polygon.
2. The local XYZ axes of the curve are aligned with those of the selected polygon (the normal of the polygon is Y).
3. The polygon is extruded along the curve.

As long as you draw the curve going up the y-axis in the Front window, it is correctly placed for polygon extrusion. You can select several polygons at once and the curve is correctly reoriented for each extrusion.

Note

If you want to change the orientation of the extrusion, rotate the extrusion curve *before* extruding. This changes the relative orientation of the xz axes to the desired positions.

Manual Curve Placement

For greater control, you can deselect **Curve Placement**, but watch out – the curve’s coordinate system is not lined up to the local coordinate system of the polygon. The extrusion takes place along the curve wherever the curve is positioned, so you should draw the curve “in place” directly out from the polygon.

You cannot extrude multiple polygons with **Curve Placement** deselected because the results would be unpredictable.

Using a Reference Null

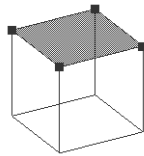
For greater accuracy in drawing a curve for this purpose, try using a reference null object with its coordinate system aligned with that of the selected polygon(s).

To do this, select the polygon. Switch to POL mode and choose the **Polygon → Reference Frame** command. A null appears on the polygon.

In the Right window, you can look down along the x-axis of the null (aligned with the x-axis of the polygon) and start drawing your curve out along the y-axis of the polygon.

Examples

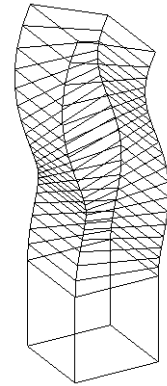
This illustrations shows extrusion of a polygon along a curve.



Selected polygon to be extruded

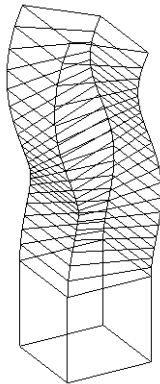


Extrusion curve
(NURBS curve drawn
in Front window)

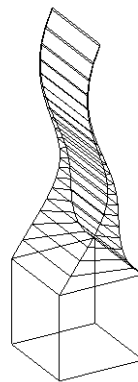


Resulting extrusion
(Step = 8, extruded on
curve, no transformations)

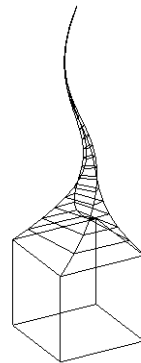
If you wish to add a “twist and taper” effect when extruding polygons, you could specify a rotation in Y (twist) and scale in X and Z (taper).



No transformations



Taper
(Scale x = 0.75)



Twist and taper
(Scale x = 0.75, Scale z = 0.75
Rotate y = 10 degrees)

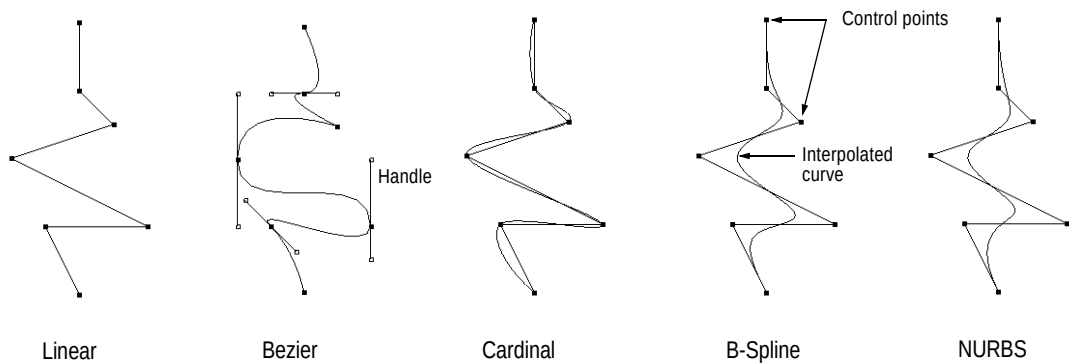
C H A P T E R F O U R

Curves

Introduction

In SOFTIMAGE 3D, there are five types of curves: Linear, Bezier, Cardinal, B-Spline, and NURBS.

This figure illustrates both the control points and the interpolated curves for each of them.



Linear

A Linear curve is made up of straight line segments between the control points. You need only two control points to create a curve segment.

Bezier

Bezier curve segments always pass through the control points. You need only two control points to create an interpolated curve segment.

A Bezier curve is unique because it has two tangent points connected by “handles” to the control point; manipulating the handles allows you to control the direction and the tension of the segment. The two tangent points are related by one of the three continuity types.

Bezier curves allow you to draw complex curve shapes with few control points because of the ability to manipulate the handles.

After drawing the curve, you can edit each point’s continuity, or change the interpolation of a selected segment from curved to linear.

Cardinal

A Cardinal curve is made up of smooth arc segments. A Cardinal curve passes through every control point. You need three control points to create an interpolated curve segment. Like B-Splines, two phantom control points are automatically generated when you draw the curve in order to achieve end point interpolation.

Cardinal curves have a parameter called tension, which can be edited in the Curve Info dialogue box (choose **Info** → **Selection**).

B-Spline

A B-Spline is made up of smooth arc segments. It is bounded and defined by the control points but does not pass through them. You need three control points to create an interpolated curve segment. To achieve end point interpolation, two additional points are generated when you draw the curve – one is added before the first drawn point and another after the last. These extra points are hidden by default, but can be visualized if necessary by editing a parameter in the Curve Info dialogue box.

NURBS

NURBS stands for *Non-Uniform Rational B-Splines*. NURBS have the greatest amount of capabilities of all curves in SOFTIMAGE 3D. Because of this, they can be described as the general category of all curves. Linear, Bezier, Cardinal, and B-Splines curves have only a subset of the capabilities of NURBS. For more details on NURBS, see *Some Basics about NURBS Curves* on page 94.

Working with Curves

All of these curve types, as well as Bezier and NURBS free form curves, can be drawn using the **Draw** → **Curve** commands in the Model module. With the **Get** → **Primitive** commands, you can select predefined curves, such as spirals, arcs, or circles.



Primitive curves: circle, arc, spiral, and square.

Having drawn a curve, you can edit it in a variety of ways using the commands available in the **Draw**, **Effect**, and **Edit** menu cells in the Model module.

Starting from a curve profile, you can create 3D objects using the **Surface** menu commands: Extrusion, Revolution, and Skinning, described starting on page 130.

Another way to use curves is to deform other objects along the shape of a curve, as described on page 300.

You can also animate objects along the curve: a simple process of associating a model with a curve path and saying “move the object along this path.” Path animation is described starting on page 67 of *Animating User’s Guide*.

Some Basics about Curves

Curves are defined by a set of control points. More precisely, they are actually a collection of curve segments defined by knots.

Curves can be *open* when the starting and ending points are at different locations (an arc), or *closed* when the starting and ending points are at the same location (a circle).

Once you have drawn a curve, you can perform various modelling operations on it using various commands in the **Draw**, **Edit**, and **Surface** menus. Some of the available modelling operations are:

- Edit the shape of the curve by moving, adding, or deleting points.
- Extract it, scale, rotate, or translate the extracted version.
- Make a skin over multiple curves and model the resulting surface.
- Extrude or revolve it to make a 3D object.
- Project closed curves onto 3D objects to create shapes out of the object; extrude the surface curves out of the objects, model the extrusions and so on.

Since curves are not 3D objects or 2D faces, they are not visible in the final rendering of the image. They can be seen in Shade view when orbiting, zooming, or dollying the camera. If you have selected the **Show Icons** option in the Shade View Setup dialogue box, they are always visible.

Control Points

The *control points* lie on or off the curve, depending on the curve type. Linear, Bezier, and Cardinal curves pass directly through the control points, whereas B-Splines and NURBS curves of anything other than linear degree usually do not. Depending on the curve type, the control points are interpolated differently to create the curve.

B-Splines and Cardinal curves do not have true end point interpolation. In SOFTIMAGE 3D, this is achieved through the use of phantom control points which are automatically added before the first and after the last control point.

Steps

The curve is drawn with small straight lines called *steps*. The higher the step value, the better the approximation to the real curve. When drawing a curve, the default step is 10, which can be edited later.



Step value: Bezier curve with step of 10, 5, and 3 (left to right).

The number of steps defines the resolution of curves and surfaces, both for viewing and rendering. However, it has no influence on the smoothness of movements when a curve is used as a path for animation. The number of steps can be modified using the dialogue boxes opened by choosing the **Info** → **Selection** command.

The Mathematics of Curves

In mathematical terms, curves are *piecewise*, *polynomial*, and *parametric*.

Piecewise means that curves are composed of segments, joined end-to-end (or knot to knot), rather than one uninterrupted curve. For this reason, each segment has more local independence when editing and makes it possible for different degrees of continuity of the curve to exist where the curve segments meet.

Polynomial is a term related to the powers of a variable. This means that the degree of the curve can vary: linear (degree 1), quadratic (degree 2), cubic (degree 3), and so on.

Note Models created externally with NURBS can be imported into SOFTIMAGE 3D as IGES files, and manipulated using existing tools. These imported NURBS can be of any degree.

Parametric is a term related to the knots. The parameterization of the curve indicates how you are moving along the curve segments, and determines the speed at which you move from knot to knot. For more information on the significance of different curve parameterizations, see *Parameterization* on page 97.

Some Basics about NURBS Curves

NURBS stands for *Non-Uniform Rational B-Splines*. NURBS have the greatest amount of capabilities of all curves in SOFTIMAGE 3D. Because of this, they can be described as the general category of all curves. Linear, Bezier, Cardinal, and B-Splines curves all have a subset of the capabilities associated with NURBS curves.

NURBS have several unique characteristics which allow greater control than other curve types:

- Their degree can be defined.
- Each control point can be weighted.
- The curve parameterization can be something other than uniform.
- You can control the precision of extracted and projected curve fitting.

Control Points

The control points for NURBS of linear degree lie on the curve. You only need to draw two points to create an interpolated curve segment representing a straight line at any slope.

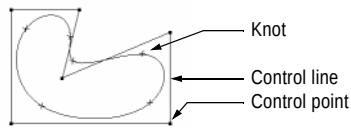
For quadratic NURBS curve, you need to draw at least three control points to create an interpolated curve; for cubic curves, at least four control points.

For quadratic and cubic NURBS curves, the control points lie off the curve, except in cases where there are double knots (for quadratic curves) or triple knots (for cubic curves). In this case, the control points lie on the curve at the knot locations.

Knots

Knots are points on the curve or surface where segments join. Unlike control points, knots cannot be seen in the windows or directly manipulated and are defined here only for purposes of understanding the behaviour of curves. To change the shape of the

curve, you move or edit the control points. In so doing, you edit the position of the knots on the curve.

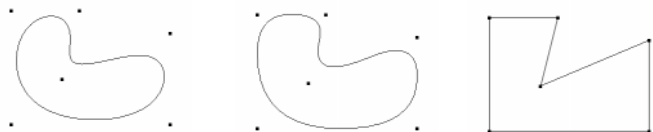


NURBS curve with position of knots indicated.

Degree and Continuity

Before drawing a NURBS curve, you can choose the *degree* of the curve (linear, quadratic, or cubic).

The degree determines the number of points you are required to draw to interpolate a segment of the curve, as well as the smoothness and continuity of the curve.



Degree of NURBS curve: Cubic, quadratic, and linear (left to right) with control points shown.

Continuity relates to the manner in which curve segments meet. In turn, it determines the relative smoothness of the curve interpolation at these points of meeting.

Note

The effects of different continuity are most evident when you edit the curve.

Linear

A linear type creates a NURBS curve of the first degree. You must draw two control points to get a curve segment.

The resulting curve has *positional* continuity, also called *zero-order* continuity or *C0*. This means that two curve segments meet at the same position and there is no break in the curve.

Quadratic

A quadratic type creates a NURBS curve of the second degree. You must draw at least three control points to get an interpolated curve segment.

The resulting curve has *positional* and *tangential* continuity, which means that two curve segments meet at the same position and the same angle. This is also called *first-order* continuity or *C1*.

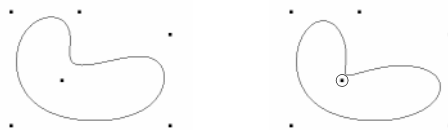
Cubic

A cubic type creates a NURBS curve of the third degree, which means you must draw at least four control points to get an interpolated curve segment.

The resulting curve has *positional*, *tangential*, and *curvature* continuity; this means that where two curve segments meet, the curvature of the two segments match. This also called *second-order* continuity or *C2*.

Weight

Each control point of a NURBS curve has four parameters: x, y, z, and a fourth parameter w (weight). *Weight* refers to the relative influence of each control point on the curve segments in its proximity. Weight is always greater than 0: the greater the weight of the control point, the closer the surface or curve moves towards it. A weight of less than 1 actually pushes the curve or surface away from the control point. For example, you can adjust this value for a control point to make a nose pointier.



Weight of a control point: Higher value increases influence (affected control point shown circled).

A NURB is truly *rational* when at least one of its control points has a weight factor other than 1.

For more information, see *Editing NURBS Weight* on page 121 and a description of the **Edit ➔ NURBS Weight** command on page 413 of the *Reference Guide*.

Parameterization

As already mentioned, knots lie on the curve and are not visible.

Linear, Bezier, Cardinal, and B-Splines all have *uniform parameterization*, in which the delta (difference) between successive knots is expressed in equal increments of 1.

By contrast, NURBS are *non-uniform* because the delta (difference) between successive knots does not need to be expressed in uniform increments of 1. It is this non-uniformity which distinguishes NURBS from other curve types and gives them a unique flexibility and localized control.

In theory, non-uniform parameterization simply means “anything goes” so any spacing is possible. In practice, the textbooks describe several different types of non-uniform parameterization, each of which attempts to define some relationship between the geometry of the curve and how you are travelling along it.

In SOFTIMAGE 3D, there are four knot parameterization possible for NURBS: Non-Uniform, Chord Length, Centripetal, and Uniform.

Chord Length

For chord length, the difference between successive knot values is related to the actual length of each curve segment. If you could see the knot values on the actual curve, these values might be unequal: 1, 3.2, 5.3, and so on. Contrast this with Uniform parameterization on page 98.

Chord length always strives to preserve the relation between the length of the curve segment and the spacing of the knot vector. When you change the geometry of the curve, the parameterization of the curve is completely recomputed to preserve the relation of length of segment/knot spacing.

As a result of this recomputation, when you move a control point, the edited segment jumps to a slightly different position when you release the mouse button.

Centripetal

This option is similar to chord length in that when you change the geometry of the curve, the curve parameterization is completely recomputed, with the same results as described for chord length.

One slight difference, however, is that the knot spacing is related to the square root of the length of each curve segment.

In some cases, this may make smoother curves; on surfaces of very unevenly spaced control points, it may give slightly better texture mapping results than chord length.

Non-Uniform

Non-uniform is the default value used for drawing a NURBS curve in SOFTIMAGE 3D. In this case, the first time the curve is drawn, the delta between successive knots is related to the length of each curve segment.

Unlike chord length or centripetal, when you change the geometry of the curve, the parameterization is not recomputed. Instead, it strives to preserve the exact shape of your edit and the curve “stays put” when you release the mouse.

For this reason, changes are more localized if you use non-uniform parameterization than if you use chord length and centripetal.

To demonstrate the difference, draw two NURBS, one with non-uniform and with one chord length parameterization. Try editing any of the control points on each and notice the slight difference in behaviour.

Uniform

For uniform parameterization, the delta between successive control points is expressed in equal incrementations of 1 without regard for the actual length of the curve segment. This choice may seem paradoxical for a NURBS, but just as whole numbers are a subset of the real number set, uniform parameterization can be seen as a subset of the generalized set of “non-uniform.”

It is useful to maintain uniform parameterization for the Skinning operation described on page 138, and the Merging operation described on page 111.

For more information, see *Joining Surfaces* on page 168.

Effects of Parameterization

The following section compares the effects of non-uniform types of parameterization (Non-Uniform, Chord Length, and Centripetal) with Uniform parameterization.

Control of Curve Shape

You can edit, subdivide, open, close, or merge NURBS curves without changing the shape of the whole curve. This provides better local control of curve shape, and is the major difference between curves of non-uniform parameterization (NURBS) and curves with only uniform parameterization (Linear, Bezier, Cardinal, and B-Spline).

There are slight differences in control of curve shape among Chord Length/Centripetal parameterizations and Non-Uniform parameterization, as described in the preceding pages.

Path Animation

SOFTIMAGE 3D ignores curve parameterization and creates an identical function curve for any curve used as an animation path. Initially, there is no difference in the travelling time of an object along any such curve, regardless of the curve parameterization.

But what if you want to customize the way or speed at which an object moves along the path? You edit the function curve as you normally do in the Fcurve window, as described in *Editing Function Curves* on page 30 of the *Animating User's Guide*.

Better Texture Mapping

Texture mapping applies to NURBS surfaces rather than curves, and is discussed on page 158.

Curve Fitting Formula

When you project curves onto NURBS surfaces or extract NURBS curves, you can determine the exactness of the fit of the resulting curve using the **Draw ➤ NURBS Curve Setup** command and its related dialogue box, shown on page 102.

Error Tolerance

You can select how closely you wish the resulting curve to fit the input data. By default, error tolerance is set low. When speed is more important than precision, as when working on a large project, you may wish to set it higher.

Angle of Discontinuity

You can also set the angle at which the sharp angles in the input data remain sharp or discontinuous in the resulting curve, and no rounding takes place. The default value is 60 degrees.

This means that angles greater than 60 degrees still experience some rounding. Angles less than 60 degrees would remain sharp (even if the curve is of quadratic or cubic degree).

If you set an angle of 5 degrees, a peak would have to be very sharp to make a discontinuous point; otherwise, there is some rounding.

Drawing Curves

The **Draw → Curve** commands in the Model module let you draw NURBS, Linear, Bezier, Cardinal, and B-Spline curves, as well as free-form curves (Bezier or NURBS). For NURBS curves, you can set various parameters before drawing.

Direction of Drawing

The direction in which you draw curves is important if you plan to create a 3D object from the curve. When you extrude, revolve, or skin curves, the direction of the normals of the resulting objects depends on the direction in which the original curve was drawn. This has an effect on the eventual rendering of the object.

NURBS curves can be drawn *clockwise*. When you draw a Linear, Bezier, Cardinal, or B-Spline curve to create a 3D object, enter points *counterclockwise*. By doing this, the object's surface normals point in the correct direction and the object is rendered correctly.



If you draw a curve in the wrong direction, you can easily invert it using the **Effect → Inverse** or the **Effect → Curve Controls** command, as discussed on page 128.

Default Step Value

When you first draw a curve in SOFTIMAGE 3D, the Step value used is 10. You can subsequently change this in the Curve Info dialogue box. Select the curve and choose **Info → Selection**.

Drawing Non-NURBS Curves

1. Choose **Draw → Curve** and one of the curve subcommands in the Model module.
2. Enter points one after the other by left-clicking.
For free form curves:
 - Choose the **Draw → Curve → Bezier Free form** command.
 - Click and drag to create the curve. When you release the mouse, it becomes a Bezier curve.
3. When drawing Bezier curves, you can drag the mouse slightly as you enter a point, in which case the characteristic “handles” of a Bezier control point appear. These also allow you to edit the point later.
4. Press Esc to end the drawing mode.

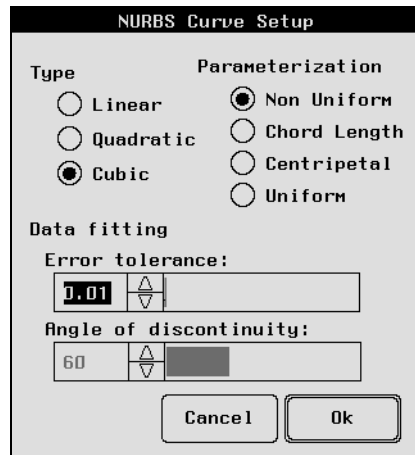
Drawing NURBS Curves

You can draw a NURBS curve as you do other curves, including a free-form NURBS curve.

Unlike other curves, you can (optionally) alter some setup options before you start drawing, by choosing the **Draw → Curve → NURBS Curve Setup** command.

1. Choose **Draw → Curve → NURBS Curve Setup**.

The NURBS Curve Setup dialogue box appears, displaying the options which control the degree and parameterization of the NURBS curve.



- Under **Type**, select the degree of the curve (Linear, Quadratic, or Cubic). These options are defined starting on page 95.
 - Select the curve **Parameterization** (Non-Uniform, Chord Length, Centripetal, or Uniform) described on page 97.
 - Under **Data Fitting**, use the sliders to set fitting parameters for projected or extracted curves. The options are described on page 100.
2. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.

If you simply choose the **Draw → Curve → NURBS** command, you apply the current setup parameters.

3. Choose the **Draw → Curve → NURBS** command. Enter points one after the other by clicking the left mouse button.

For free form curves:

- Choose the **Draw → Curve → NURBS Free form** command.
- Click and drag to create the curve. When you release the mouse, it becomes a NURBS curve.

By default, NURBS curves are drawn using a step value (resolution) of 10. To edit the curve resolution, select the curve, choose **Info → Selection**, and change the **Step** value in the dialogue box.

To subsequently modify the knot parameterization, choose the **Effect → Reparameterization** command.

Copying a Curve

There are several ways to make copies of curves:

- You can draw a curve and duplicate it.
- You can extract a selected curve from a face or surface.
- You can load an image (such as a logo) and copy a selected contour from it.

Duplicating a Curve

For more information on duplicating objects, see *Duplicating Objects* on page 32. Here's the basic steps:

1. Select the curve you wish to duplicate.
2. Choose one of the **Duplicate** menu commands.
3. Duplicate Immediate makes a copy of the curve and places it in the same position in the work space. Translate one in order to visualize and manipulate it.

Extracting a Curve from a Surface

The **Draw** ➔ **Extract** command in the Model module allows you to extract a copy of a curve from a face or a surface, leaving the original object untouched. The extracted curve is the same type as that used to create the object from which it is extracted.

This command is useful if you wish to copy a curve identical to one on an existing surface. Having extracted this curve, you can extrude, revolve, or skin it, or perform any other operation that can be done with a curve.

From Faces and Patch Surfaces

If you are working with a face or patch, you can only extract curves as displayed. The extraction is executed immediately and you must reselect the surface if you want to extract another curve.

From a NURBS Surface

If you are working with a NURBS surface, a related dialogue box allows you to extract a curve anywhere on the NURBS, not just at the curve lines displayed. You can also repeat the operation interactively, extracting as many curves as you want before closing the dialogue box.

You can also extract any curves that you have projected onto a NURBS surface, and use it for any modelling purposes. It may be particularly useful for creating a skin to seamlessly connect two NURBS surfaces, such as a head and an ear. For more information, see *Examples of Modelling with Projected Curves* on page 199.

Using Relational Modelling

You can use relational modelling concepts on the curve extracted using this command and thus on any objects created from that curve.

For example, you can extract a curve from a surface and then extrude the curve. By editing the original surface, you automatically modify the shape of the extracted curve, and in turn, the shape of the extrusion.

To use relational modelling, make sure that the **Preferences → Create Modelling Relation** command is activated first. Relational modelling is discussed on page 143.

Procedure

This is the basic procedure for extracting a curve:

1. Select a face or surface object.
2. Choose the **Draw → Extract** command.
3. Pick the face or any line on a surface. The curve flashes red to indicate that it is selected so you can transform it as desired.
4. If you are working with a patch surface:
 - Pick the curve on the object to extract on U or V. The extracted curve is now selected the patch object is deselected.
 - Right-click to end the mode.
 - To pick another curve, select the patch surface again and repeat the operation.
5. If you are working with a NURBS surface object, the NURBS Extract Curve dialogue box appears.
6. Select any **Location** in U or V to extract a curve. Select any projected or trim curves from the list. Click **Extract**. You can keep extracting without exiting the dialogue box.
7. When satisfied with the extraction(s), click **Exit** to exit the dialogue box.

Importing Curves for Editing

The **Get → Eps2soft** command in the Model module allows you to import an .eps file (for example, a logo from Freehand™ or Illustrator®). The result in SOFTIMAGE 3D is a Bezier curve which can be edited.

You can also load an image into SOFTIMAGE 3D and then copy one of its contours to create a curve using either the **Autotrace** menu cell in the Tools module, or the **contour** standalone from the command line, which allows batch processing.

The **Autotrace** menu cell processes a picture to extract its contours and creates Bezier curves from them. The main application is to extract curves from logos, providing an alternative to digitizing with a tablet.

First, the picture should be processed to create a high contrast picture. For this purpose, a threshold value is used to determine whether a pixel should be considered black or white. The threshold value represents the average value of the three RGB colour components and ranges from 0 to 1.

Once the high contrast representation has been completed, it is analyzed to extract the contours.

1. In the Tools module, choose the **Autotrace** menu cell. The database browser appears.
2. Select the image to be processed by double-clicking it. The Autotrace dialogue box appears.
3. Set the **Threshold** value. If you set the threshold to 0.3, no contour is extracted because both grey and black RGB averages are under the threshold value and are therefore considered similar in high-contrast representation. A threshold value of 0.001 to 0.2 is usually a good range of values (the default is 0.3).
4. Set the **Minimum number of points**, which sets the curve's resolution.
5. You can also set the **Depth search** value, which specifies the number of levels to search.
6. Click Ok to process the curves.

Using the contour Standalone

Like the **Autotrace** menu cell, the **contour** standalone extracts the contours of an image; however, you can run this standalone from a command line in a shell and use it for batch processing. It works well only with images that have good contrast between objects or faces.

You can specify the threshold on the command line. The threshold is the value of the smallest difference of intensity between two adjacent pixels for the standalone to determine that an edge exists.

Normally, pixels with an intensity difference lower than the threshold become black, and pixels with an intensity difference greater or equal to the threshold become white. The result can be inverted with the **-i** option.

The high-contrast output images are non-filtered.

This example extracts curves from the **shape.pic** file from frames 1 to 20 with a step of 1, and saves the results as **shape2**. The start threshold is set to be constant at 0.1 over the frames and the effect is inverted.

```
contour shape shape2 0.1 -1 -s 1 20 1
```

Opening and Closing Curves



Open NURBS curve.



Closed curve.

With the exception of Cardinal and B-Splines, curves are closed or opened at the starting point of the curve in the U direction. To visualize this, choose the **Show ➔ Edge Flags** command.

Operations Requiring Open or Closed Curves

In some cases, you must close or open a curve in preparation for performing a specific operation.

For example, you can only project a closed NURBS curve onto a surface, so you would first draw the curve and close it in preparation for projecting it or using it to trim a NURBS surface (see *Modelling NURBS Surfaces with Projected Curves* on page 185).



You cannot open or close a *trimmed* NURBS Surface. For more information, see the note and procedure on page 162.

You must close a curve before converting it to a face. Conversely, you can only merge two open curves or surfaces.

In some cases, the requirements depend on the type of curve. For example, you can only cut (**Draw ➔ Cut**) open Linear, Bezier, Cardinal, and B-Spline curves, whereas you can cut open or closed NURBS curves. Cutting a closed NURBS curve has the effect of opening it.

Opening or Closing Non-NURBS Curves

1. Select a curve other than a NURBS.
2. Choose the **Draw ➔ Open/Close** command.

The Linear, Bezier, Cardinal, or B-Spline curve is opened or closed automatically.

Opening or Closing NURBS Curves

When you close a NURBS curve, it remembers your selection for closing it. If you reopen the curve, it is reopened the same way it was closed.

1. Select a NURBS curve and choose the **Draw ➔ Open/Close** command. The NURBS Open/Close dialogue box appears.

2. Select the method of opening/closing the curve: **Normal**, **Preserve Shape**, or **Linear**.

When closing a NURBS curve, you can choose whether or not to preserve the shape of the whole curve; you also have more control over the shape of the closing segment using the **Start**, **Close**, and **End Extent** sliders.

3. If you choose **Preserve Shape**, the **Start**, **Close**, and **End Extent** sliders become available. As you adjust the sliders, the shape of the closing segment changes interactively.

You can change the method of closing the curve and see the results before you exit the box.

4. Click **Ok** to exit the dialogue box or **Cancel** to exit without accepting the settings.
5. If you then wish to reopen the curve and repeat the operation, choose **Draw → Open/Close** again, keeping in mind the following notes:
 - You can close the curve again, but it is not advisable to change the closure method because the results may be unpredictable. For example, if you close a curve with the **Normal** option, you can reopen it, but you shouldn't change to **Linear** or **Preserve Shape**. If you make a mistake, undo the operation and choose a different closure method.
 - **Draw → Open/Close** opens or closes a curve at point 0 of the curve. If necessary, you can use the **Effect → Curve Controls** command to shift this point, which shifts the point where the curve is opened or closed. This may also be useful for skinning or merging curves, or zipping surfaces. For more information, see *Shifting the Starting Point of a Curve* on page 127.

Merging Curves

The **Draw ➔ Merge Curves** command in the Model module allows you to combine any two open curves to make a new curve. When you choose this command, a different dialogue box appears according to whether you are joining non-NURBS or NURBS curves. Using these dialogue boxes, curves may be merged or blended, depending on the options you select.

Certain options preserve the shape of the original curves and add a new region between them. This operation is known as *blending* and the new region is called the *blend region*. Other options change the shape of the existing curves; this operation is known as *merging*.

For purposes of simplification, the general term *combining* is used in the following description.

- When you combine two curves, the original curves are preserved until you delete them. If you make a mistake in combining, delete the resulting curve and start again using the originals.
- The order of selection is significant; that is, the end of the first selected curve is joined to the start of the second selected curve. To help you see the curves' starting points, choose the **Show ➔ Edge Flags** command. You can use the **Effect ➔ Inverse** command to change the start and end points if necessary.

Merging Non-NURBS Curves

You can combine curves of different types, but the resulting curve is Bezier. For example, you can combine a Cardinal and a B-Spline, but the result is a Bezier curve.

When using the **Factor** option, the original curves must be of the same type; for example, you can combine two Cardinal curves using a Factor, but not a B-Spline and a Cardinal.

1. Select the first curve.
2. Choose the **Draw ➔ Merge Curves** command. When combining Linear, Bezier, Cardinal, or B-Spline curves, the Merge Splines dialogue box is displayed.
3. Select the desired option(s):
 - **Add linear (CO) segment between them** joins the end of the first and start of the second curve with a straight line.

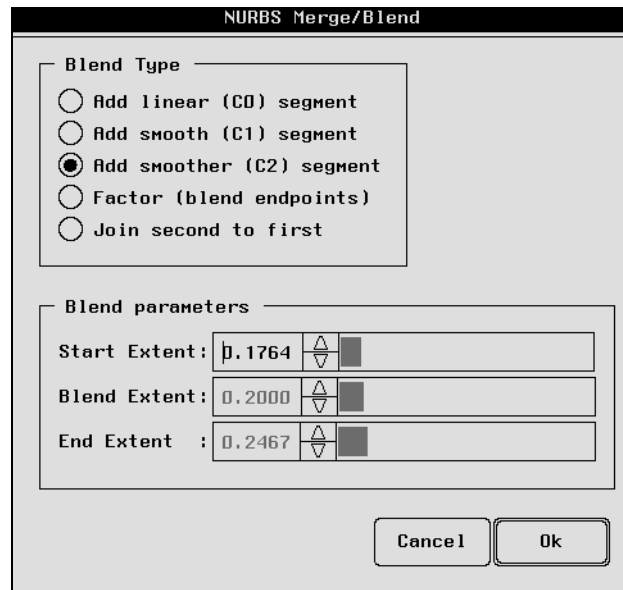
- **Add smooth (C1) segment between them** joins the end of the first and start of the second with a smooth curved line.
 - **Translate second one to join extremity** moves the second curve so that the first point of the second curve is attached to the ending point of the first curve.
 - **Factor** pairs and merges the last three points of one curve and the first three of another. You can also adjust the position of these (merged) points by moving the slider between 0 and 1. You can use this option only for Cardinal and B-Spline curves. For more information, see *Factor* on page 298 of the *Reference Guide*.
4. Click Ok to accept the settings and exit the dialogue box.
 5. Pick the second curve.

The two curves are combined.

Merging NURBS Curves

- You can only combine a NURBS curve with another NURBS curve, and the resulting curve is, you guessed it, also a NURBS curve.
 - Both NURBS curves must be of the same degree. For example, you can combine two NURBS of cubic degree, but not a quadratic and a cubic degree curve.
 - The blend region cannot be of a continuity higher than that of the curves you are merging. For example, if the curves are quadratic (C1), the blend region can be C0 or C1, but not C2.
 - SOFTIMAGE 3D allows you to import NURBS curves of a degree greater than 3 (cubic). However, these curves of a higher degree may not have all the merge/blend options available.
1. Select the first curve.
 2. Choose the **Draw → Merge Curves** command.

- Pick the second curve. The NURBS Merge/Blend dialogue box is displayed.



- Select the **Blend Type**. You can choose to create a C0, C1, or C2 blend which preserves the shape of the two original curves. Alternately, you can choose **Factor**, which interpolates (takes an average of) the start and end points of the curves and does not preserve the shape of the original curves.
- Set the **Blend parameters**. You cannot access all three **Extent** sliders unless you select **Add C2 smoother segment** or **Factor** as the **Blend Type**. The **Extent** sliders let you change the blending region along the curves.

For more details on these parameters, see page 299 of the *Reference Guide*.

The NURBS dialogue box is interactive, so you can apply the settings immediately to the combined curves and view their effects in the windows.

- Click Ok to accept the settings and exit the dialogue box.

Custom Effects for Merging Curves and Other Objects

The **Polygon → MergeGeometry** command in the Model module allows you to merge the geometry of any curve with that of an arbitrary polygon mesh object. You simply model the curve and the polygon mesh and then determine the projection plane in which the merge takes place – the points are added automatically.

The **Polygon → Stitch** command in the Model module allows you to create seamless joints between free or open edges of polygon meshes or curves by automatically creating joining triangles between nearby points of participating models. When stitching curves, all the points are automatically used. Difficult modelling tasks, such as joining a circle to a square, or connecting fingers smoothly to a hand, are easily performed using this effect.

The **Effects → Glue** command in the Motion module allows you to “glue” tagged points of one object to the tagged points of another surface object. You can apply the Glue effect to polygon meshes, patches, NURBS surfaces, or curves.

Glue is a motion effect, so either of the models can be animated, and the glued portions stick together. Falloff parameters allow the effect to be applied smoothly to nearby points.

For more information on using these effects, see their command descriptions in the *Reference Guide*.

Cutting a Curve

The **Draw → Cut** command in the Model module allows you to cut open or closed curves depending on the curve type. The cut section can then be translated as desired, while the original curve or surface is preserved.

You can continue cutting until you end the mode; at each step, the current selection is preserved. For example, cut a curve in half and the original (whole) curve is preserved. Cut the half curve in half, and the half curve is preserved as well as the quarter curve, which is currently selected.

Non-NURBS Curves

You can cut only an *open* Linear, Bezier, Cardinal, or B-Spline curve. B-Splines are cut at the knot position on the curve nearest to the mouse click. Linear, Bezier, and Cardinal curves are cut at the control point nearest to the mouse click.

1. Select the curve you want to cut.
2. Choose the **Draw → Cut** command.
3. Use the left or middle mouse buttons to cut the curve or surface.

You can continue cutting until you end the mode; at each step, the current selection is preserved.

4. Right-click to end the mode.

NURBS Curves

You can cut an open or closed NURBS curves. An interactive dialogue box allows you to choose an exact location for the cut. Cutting a closed NURBS curve has the effect of opening it.

1. Select the NURBS curve you want to cut.
2. Choose the **Draw → Cut** command. The NURBS Curve Cut dialogue box is displayed.
3. Use the **Location** slider to interactively position the magenta point on the curve.
4. Click **Cut** to perform the cut.

You can continue cutting until you end the mode; at each step, the current selection is preserved. Click **Exit** to close the dialogue box.

Adding Lines and Arcs

After drawing a curve, you can add lines or arcs to the end of it, or edit the shape of the last n segments on the curve.

The **Draw → Drafting** command in the Model module allows you to add straight lines and arcs to Linear, Bezier, Cardinal, and B-Spline curves and modify them interactively.

Drafting simulates 2D manual drafting and should be used in the Front window only.

The **Draw → Drafting** command converts Cardinal and B-Spline curves to Bezier curves. This command is not currently applicable for NURBS curves.

1. In the Front window, select a non-NURBS curve, such as a Bezier or B-Spline.
2. Choose the **Draw → Drafting** command. The Drafting 2D dialogue box appears.

The original curve is preserved; a new Bezier version of the curve is created in the same position and remains selected. Your choices are performed on the Bezier curve.

3. Select the type of segment to add: **Add Line** or **Add Arc**. The new arc or line may start from the last point of the existing curve or another point on the curve, depending on the order of selection.

Adding Lines

4. If you accept the default settings and click **Add Segment** immediately, a line is added onto the last point of the Bezier curve.
5. If you adjust the **Distance** or **Angle** sliders first, the line is added starting from the second to last point on the Bezier curve.

Adding Arcs

6. If you accept the default settings and click **Add Segment** immediately, an arc is added after the *last point* of your curve; the number of segments of the added section depends on the current **Step** value.

7. If you want to edit the existing curve segments, don't click **Add Segment**: use the sliders to edit the Arc parameters first (**Radius**, **Angle**, and **Step**).
 - In this case, **Step** value has a somewhat different (potentially surprising) effect. The new arc is considered to be n curve segments of the *existing* curve, where n is the value in the **Step** text box. Based on the default **Step** value of 8, the new arc is the last 8 curve segments of the existing curve.
 - To edit your curve sanely, first change the **Step** value to the desired number of curve segments you want to edit, then adjust the sliders, and finally click **Add Segment**.
8. You can continue adding and editing as many lines or arcs as desired using the **Add Segment** button without exiting the dialogue box. The results are shown interactively.
9. Click **Ok** to accept the additions to your curve or **Cancel** to exit the dialogue box without saving the additions.

For information on the parameters of this dialogue box, refer the **Draw → Drafting** command on page 313 of the *Reference Guide*.

Using Custom Effects for Adding Borders

The **Draw → Curve → Border** and **HrcBorder** commands in the Model module allow you to create borders around curves or faces, or a hierarchy of curves or faces. This tool is useful for creating mattes. You simply specify the width of the border to create a new curve with edges parallel to those of the original.

The **Draw → Curve → Pipe** command in the Model module is a simple effect for bordering an open curve by a given amount and joining it with the original curve to create a new, closed curve. This is useful for quickly creating outlines of uniform width for walls or letters.

For more information on using these effects, see their command descriptions in the *Reference Guide*.

Converting a Curve

You can convert curves to faces, to other types of curves, or to a polygon mesh object.

To a Face

The **Draw → Convert to Face** command converts a closed curve into a face, which is an object that can be rendered. You can also convert a face to a polygon mesh object using the **Effect → Convert** menu command. For information on creating faces, see *Creating Faces* on page 150.

1. Draw or select a curve, or use **Get → Primitive** to get a predefined primitive curve such as an arc or a circle.
2. To close the curve, choose the **Draw → Open/Close** command.
3. Choose the **Draw → Convert to Face** command.

The curve becomes a face.

Note

The original curve is deleted *unless* it is already being used as an animation path, a path of extrusion, or for curve deformation.

Custom Effects for Converting a Hierarchy

The **Draw → Splines2Faces** command in the Model module allows you to convert a hierarchy of curves into a hierarchy of faces in a single step. The Splines2Faces effect also recognizes curves within curves in the hierarchy and creates corresponding faces with holes in the result. This is typically used in conjunction with the Eps2Soft and Beveling effects. For more information on using this effect, see its command description in the *Reference Guide*.

To Another Curve Type

Depending on the modelling task, you may wish to draw a curve of a certain type, and subsequently convert it to another. For example, you may wish to draw a curve as a Bezier to take advantage of its modelling flexibility, but if you wanted to project the curve onto a surface, you would need to convert it to a NURBS curve.

The **Effect → Convert** command converts curves from one type to another by one of the following methods:

- Keep the control points, but do not preserve the shape of the curve. When you select a different curve type, the control points remain in the same spot but the shape of the curve changes.
- Preserve the shape of the curve while converting to a different type. In this case, the only conversion choices offered are those in which the curve shape can be maintained.



This command allows you to convert faces or surfaces into different object types, as described in *Converting a Face* on page 152 and *Converting Surface Objects* on page 166.

1. Select a curve.
2. Choose the **Effect → Convert** command. The Convert Curve dialogue box is displayed.
3. Select either the **Keep Control Points** or **Keep Curve** option and then select the new type of curve.

If you keep the same number of points, the shape of the curve may change as it is converted. Conversely, if you keep the curve's shape, the number of points on the curve changes.

4. If **Keep Curve**, **Linear** is selected, set the **Subdivision** value or accept the default value 10.



If you are satisfied with the shape of your original curve, then leave the **Subdivision** value fairly high. This provides sufficient sampling of the original curve to preserve its shape, even in a linear representation.

If **Bezier** is selected, set the **Maximum Error**.

5. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.

The curve is converted according to your settings. The original curve is preserved.

Editing a Curve

How can you edit a curve in SOFTIMAGE 3D? Let us count the ways. You can:

- Modify the curve shape locally or by using proportional editing.
- Add control points to and delete control points from the curve.
- Shift the starting point of a closed curve, thereby relocating the point at which the curve can be opened or closed.
- Invert the curve direction.
- Edit the curve resolution
- Edit the continuity of the curve segments.

Modifying the Curve's Shape

After drawing a curve, you may wish to change its shape, to make it “pointier” or flatter, for example.

More control points mean more local control since each point can be manipulated without greatly affecting the overall curve. If you wish to edit a curve segment locally, try drawing it initially with more control points in the area you may wish to edit later. When you add and delete points from a curve, the curve shape changes.

Each of the following operations offers a greater or lesser degree of local control over the curve shape. Some operations involve changing the continuity of a curve segment, and can be applied only on NURBS or Bezier curves.

Moving a Point

To move one point at a time, use the **m** Supra key or choose the **Edit** → **Move Point** command and translate the desired control point.

In the case of NURBS curves, changes in curve shape can vary depending on the parameterization of the curve: curves with chord length or centripetal parameterization are completely reinterpolated, but those with non-uniform parameterization are not. Most of the time, the change is only local.

For more information on moving points on all object types, including proportional editing, see *Moving Points* on page 44.

Modifying a Bezier Curve's Shape

You can use the handles of each Bezier control point to interactively edit the shape of a Bezier curve.

1. Select a Bezier curve.
2. Choose **Show ➔ Lines** to display the handles of the control points.
 - Handles can be blue, cyan, or magenta. Depending on the continuity of the point, its handles can be symmetrical, or of different lengths, or broken.
 - If the curve is a Bezier free-form, the points may be of different continuity (C0, C1, or G1) and thus the handles can be different colours.
3. Press the **m** Supra key and adjust the handles of the appropriate point to edit the curve.

Manipulating a Bezier handle does not change the continuity of the curve at that point. To do this, choose the **Info ➔ Selection** command and edit the continuity of Bezier points or change the shape of a curve segment in the related dialogue box, as described in the next section.

Editing Bezier Points

After you draw a Bezier curve, you can edit either the continuity of the curve at individual spots or change the interpolation between specific control points.

For instance if you wanted to make a pointy peak in an otherwise smooth Bezier curve, you could choose the point and assign it C0 continuity. As soon as you subsequently adjust the Bezier handles for that point, you notice that the handles are “broken,” allowing you to make the curve pointier at that spot.

1. Select the Bezier curve.
2. Choose either the **Info ➔ Selection** or the **Edit ➔ Coordinate** command. The Curve (Bezier) Info dialogue box appears.

In this dialogue box, “keys” refer to control points. Two parameters (**Continuity** and **Segment**) are repeated in the upper and lower portion of the dialogue box.

In the lower portion, you can cycle through the “keys” (control points) of the curve using the **Next** or **Previous** buttons and see the

selection highlighted in the windows. As you cycle through, **Continuity** and **Segment** display the current values for *existing* control points on the curve, which you can edit interactively.

In the upper portion, the **Continuity** and **Segment** parameters set the default for any *new* control points you may subsequently add.

3. To select a particular control point, click **Next** or **Previous**. As you cycle through the control points:
 - The current selection is highlighted in the window and the corresponding number of the point is displayed in **Key #**.
 - The continuity (**C0**, **C1**, or **G1**) of the current selection is shown.
4. For each point, you can select a different **Continuity**.
 - **C0** (discontinuity) lets you change the length and angle of one handle independently of the other handle. The handles turn cyan.
 - **C1** (mirror) is the default continuity. The angle and length of one handle are reflected in the other handle. The handles turn blue.
 - **G1** (colinear) lets you change the length of one handle independently of the other handle. The handles turn magenta.
5. To change the shape of a single curve segment, select the **Linear** option (under **Segment**).

The associated curve segment changes to a linear interpolation. This doesn't affect the continuity of handles for the associated control point.

6. Click **Ok** to accept the changes and exit the dialogue box.

For more information on these parameters, see *Curve Info (Bezier)* on page 858 of the *Reference Guide*.

Modifying a NURBS Curve's Shape

Editing NURBS Weight

NURBS are the only type of curve which allow you to edit the **Weight** value for one control point or several tagged points, thereby modifying the curve shape over one or more segments, depending on the degree of the curve. The **Weight** parameter for a NURBS curve is described on page 96.

Some curve modifications are done easier by editing the weight of NURBS points rather than by moving the control points. For

example, you might make a better NURBS belly button by adjusting the weight of a point rather than moving the points!

The default weight for NURBS control points is 1.0. The minimum weight allowed is 0.001. The maximum depends on such things as:

- What effect are you trying to achieve? The more extreme the effect, the higher the weight you may need to assign. If you are trying to achieve a sharp point in a circle, you may need to reweight to 1,000.
- What is the current weight value of the point and its neighbouring points?

Here's an extreme example: If you are starting with a default circle in which each point has a weight of 1.0, the point can be achieved using a weight of 1,000; if the current weight for all points were already 1,000 (an unlikely situation), you would have to assign a weight of 50,000 to achieve a pointier effect.

Method 1

The **Edit → NURBS Weight** command allows you to modify the NURBS curve's shape interactively, using the mouse to edit the weight of one or more control points. The modifications in curve shape can vary considerably, depending on whether you reweight one point or several at a time.

By tagging points, you can apply the same increment/decrement to more than one point simultaneously. If you tag points in different areas of the curve, the curve changes quite differently than if you tag two in proximity.

1. Select a NURBS curve and make sure **Show → Points** is selected.
2. Optionally, tag one or more points on the curve.
3. Choose the **Edit → NURBS Weight → Point** or **Tagged Points** command, depending on whether or not you have tagged points.
4. When editing a single point, you must pick the control point.

When editing tagged points, you can click and drag anywhere in the window.

5. Drag the mouse to modify the weight of the point(s):
 - Left mouse button modifies slowly.
 - Middle mouse button modifies moderately.
 - Right mouse button modifies quickly.

As you update the weight of the point(s), the current weight value is updated interactively in the status line and the curve shape is updated interactively in the windows.

Method 2

Another way of editing point weight is by editing the W (Weight) value in the **Edit → Coordinate** dialogue box.

1. Select a NURBS curve.
2. Choose the **Edit → Coordinate** command and pick a point on the curve. The Edit Coordinate dialogue box is displayed.
3. The point coordinates are displayed in the **Position** text boxes for each axis. For NURBS curves, the **W (Weight)** parameter can be edited (this is not available for other curve types).
4. Enter new value for **W (Weight)**.
5. Click **Modify** to modify the point weight. To edit another point, you don't need to exit the dialogue box: click **Next** or **Previous** and see the point highlighted in the window. Repeat the editing steps as necessary.
6. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.

Subdividing NURBS Curves

When you select a NURBS curve, you can choose one or more exact spots along the curve at which to subdivide. You can also specify the continuity at the subdivided spot.

Technically, you are really modifying the number of knots to achieve the specified continuity at that spot. As a result of these modifications, some control points are added and others may be moved.

Note

You cannot specify a continuity higher than that of the curve as a whole. For example, if the curve is quadratic (C1), you can specify C0 or C1, but not C2.

Regardless of the continuity you choose, the shape of the curve remains the same when subdivided. However, as you move the new control point, the continuity selected determines the shape and relative smoothness of the new curve segment.

The more points you add in one area, the more localized your control when editing. For example, if you add three control points in

one area, you can then move any of these new control points and the effect of the changes is limited to a smaller area.

1. Select a NURBS curve.
2. Choose the **Effect → Subdivision** command.

The NURBS Curve Subdivision interactive dialogue box appears.

3. Use the **Location** slider to choose the appropriate spot on the curve to subdivide.
4. Select the degree of **Continuity** for the new point. The maximum value of the slider depends on the degree of the curve: for linear, 0 only; for quadratic, 0 or 1; for cubic degree, 0, 1, or 2.

If you enter an invalid value (such as 4), it is ignored and the maximum value for the current curve is used instead.

5. Click **Subdivide** to apply the operation, or **Exit** to exit the dialogue box or **Cancel** to exit without accepting the settings.

Adding Points to a Curve

After creating a curve, you can add more control points to refine it: the more control points you have, the more local control of curve shape. You may also wish to extend the curve from the beginning or from the end point.

There are several ways to add control points to a curve, some of which change the shape of the curve.

Adding Points to any Curve

To add control points to any type of curve, you can use the **Edit → Add Point** command.

Note For NURBS curves, you can also use the **Effect → Subdivide** menu command, which adds control points *without* changing the original curve shape.

1. Choose the **Edit → Add Point** command.
2. Select a curve.
3. Pick the location for the new point in any window.
4. On closed curves, all mouse buttons add a point on the curve where you click.

On open curves:

- Left mouse button adds a point after the last point.
- Middle mouse button adds a point between the points.
- Right mouse button adds a point before the first point.

5. Press Esc to end the mode.

Adding Points to NURBS Curves

The **Effect → Subdivision** command allows you to divide one segment of a curve into several segments, adding control points without changing the shape of the curve.

Contrast this with the **Edit → Add Point** operation, which immediately changes the shape of the curve.

For information and procedures for subdividing NURBS curves, see *Subdividing NURBS Curves* on page 123.

Adding Points along a Plane

Using the **Effect → Plane Clipping** command, you can add points on a curve at each point where the curve crosses a defined plane. The resulting curve is converted to a Bezier.

For example, if you have a B-Spline curve, you can select the xy plane and specify 3.0 as the z-coordinate; control points are added at each location on the curve where the xy plane intersects the curve at that z-coordinate, but the resulting curve is now a Bezier.

Note

This operation is not available for NURBS curves; instead you can use **Effect → Subdivision** or **Edit → Add Point** commands.

1. Select any type of curve, except a NURBS.
2. Choose the **Show → Point** command to visualize existing points and any points that are added.
3. Choose the **Effects → Plane Clipping** command. The Intersection dialogue box is displayed.
4. Select the intersecting plane (XY, XZ, or YZ).
5. To specify the position for plane clipping in global coordinates, enter a value.

6. Click Ok to apply the settings and exit, or Cancel to exit the box without applying the settings.

The control points are added at the appropriate location(s) on the curve.

Note

Cardinal and B-Spline curves are converted to Bezier curves as the control points are added.

Deleting Points on Curves

To delete control points from any type of curve, you can use the **Edit → Delete Point** command.

Deleting control points changes the shape of the curve.

1. Choose the **Edit → Delete Point** command.
2. Choose the **Show → Point** command to visualize existing points on a selected curve.
3. Select a curve.
4. Use any mouse button to click on the control point you want to delete.

The curve changes shape when a control point is deleted.

Creating Clusters

Clusters are groups of tagged points which can be manipulated as a unit. A single object can have multiple clusters, each of which can be selected and transformed independently of the others.

You can create a cluster with or without a cluster centre.

For more information on creating a cluster using the **Tag → Set Cluster** command in the Model module, see *Creating Clusters* on page 43.

Clusters can be transformed differently, depending on whether or not they have a cluster centre. When you select a cluster centre, the points in the cluster are scaled, rotated, and translated in relation to the cluster centre rather than object's centre. You can also constrain objects to clusters, especially if the cluster has a centre.

Clusters and their use in modelling are described in more detail in *Deforming with Clusters of Points* on page 308.

Shifting the Starting Point of a Curve

When you close a curve, it is automatically closed at the starting point in U. For purposes of explanation, we'll call this "point 1."

It may be convenient to change this point, for example, if you have a closed curve and you wish to open it at a different point. Using the **Effect → Curve Controls** command, you can select a closed curve and offset this starting point using a slider, then open the curve at a new point.

1. Select a curve.
2. Choose the **Effect → Curve Controls** command. The Curve Controls dialogue box is displayed.

For curves, only the U parameters in this dialogue box are available.

3. Use the **Shift U** slider to offset the point at which the curve can be opened/closed.

The results are displayed interactively.

4. Click Ok to accept the settings or Cancel to exit the dialogue box.

For Skinning, Merging, Zipping

When you skin, merge, or zip things together, these operations match points on one curve/surface to those on another to create the skin; that is, point 1 on one curve to point 1 on the other, and so on.

For a smooth match, the curves need to start in the same location; in practice, they often do not and the result is a "twist" effect.

If you skinned the curves with modelling relations on, you can straighten out the skin as follows:

1. Among the generator curves, select the twisted one.
2. Choose the **Effect → Curve Controls** command.
3. Shift the starting point of the curve until the points line up as required and the skin is no longer twisted.

Inverting the Direction of a Curve

If you are using the curve as an animation path, you can use the **Effect → Curve Controls** command to invert the direction of the curve, thereby reversing the direction in which the object travels along that path.

1. Select a curve. Make sure **Show → Edge Flags** is selected so that the starting point of the curve is highlighted in red in the windows.
2. Choose the **Effect → Curve Controls** command. The Curve Controls dialogue box is displayed.

For curves, only the U parameters are available.

3. If the curve is open, you can select the **Inverse U** option to change the direction of the curve.

The results are displayed interactively.

4. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.

Editing Curve Resolution

The resolution of a curve is related to the **Step** value, which can be edited in the Curve Info dialogue box. By default, the **Step** value when creating a curve is 10.

1. Select the curve.
2. Choose the **Info → Selection** command. The Curve Info dialogue box appears.
3. To change the curve resolution, enter a new value in **Step**.
4. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.

Editing the Continuity of a Curve

You can edit the continuity of points on Bezier or NURBS curves. On NURBS curves, you are modifying the number of knots on the curve at a given spot, but since these are not visible to you, it is done by interacting with the curve control points.

For more information on editing Bezier continuity, see *Editing Bezier Points* on page 120. Editing the continuity of NURBS curves is described on page 123.

Note

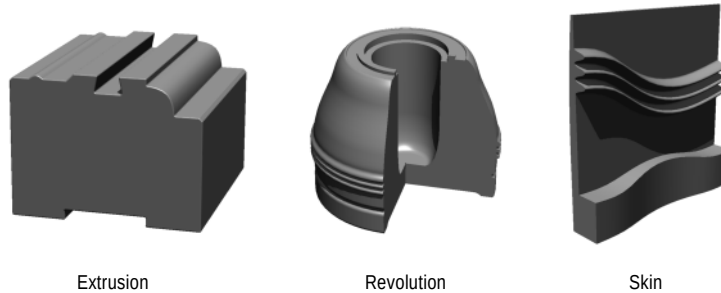
You can also change curve continuity in the Spreadsheet window.

Custom Effect for Creating Fractals

The **Effect → Flake** command in the Model module is a curve-based fractal curve generator. It repeatedly subdivides and adds features to an initial curve according to the edge pattern in a fractal “generator” curve. For more information on using this effect, see its command description in the *Reference Guide*.

Making 3D Objects from Curves

Using some of the commands in the **Surface** menu in the Model module, you can create a curve and make a 3D object from it. The resulting object can be a polygon mesh, or a patch or NURBS surface.



You can also extrude or revolve a face to make a 3D object; a face is simply a closed curve which already has been converted into a 2D planar surface. For information on creating faces, see *Creating Faces* on page 150.

Using Relational Modelling

If the **Preferences → Create Modelling Relation** command is selected, you can create a 3D object from a curve profile, then use relational modelling in editing the resulting object.

If a modelling relation exists, you can affect the generated object by editing or animating the generator curve.

Two of the operations (extrusion and revolution) allow you to extrude or revolve one curve around another curve. In these cases, there can be a three-way modelling relation, between the two curves and the generated object.

Restrictions on Generated Objects

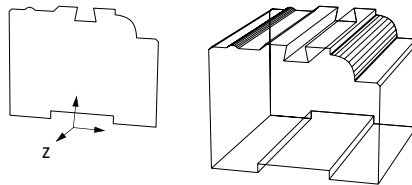
The only operations you can perform directly on the generated object are: transformations (**Scale**, **Rotate**, or **Translate**) or inversion of surface normals using **Effect → Inverse**. This last command is actually a modelling relation: the generated object is in modelling relation with itself, which allows you to invert it independently of the generator.

If you wish to separate the curve from the object for viewing purposes, select the generated object and translate it away from the generator curve. You cannot translate the generator curve away from the object because of the modelling relation.

If you wish to reparameterize a NURBS generated surface, you would have to reparameterize the generator curve. However, the parameterization is only updated at the next refresh or the next time you edit the generator curve.

Extruding a Curve

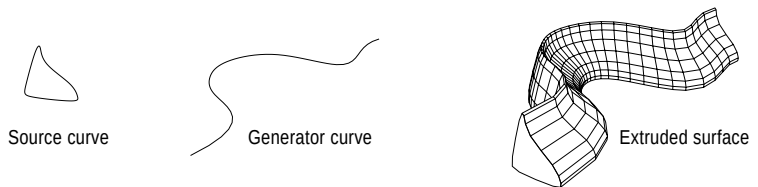
The **Surface → Extrusion** command in the Model module lets you create a 3D object out of a curve (open or closed). You can extrude the selected curve (or face) on any axis or along another curve.



Source curve

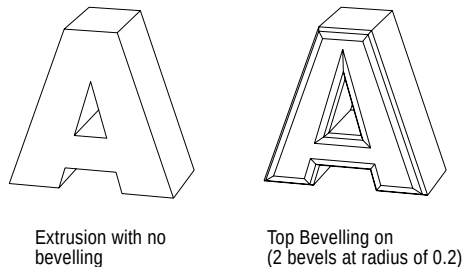
Extrusion on z-axis

The following illustrates extrusion along a path.



Extrusion along a curve

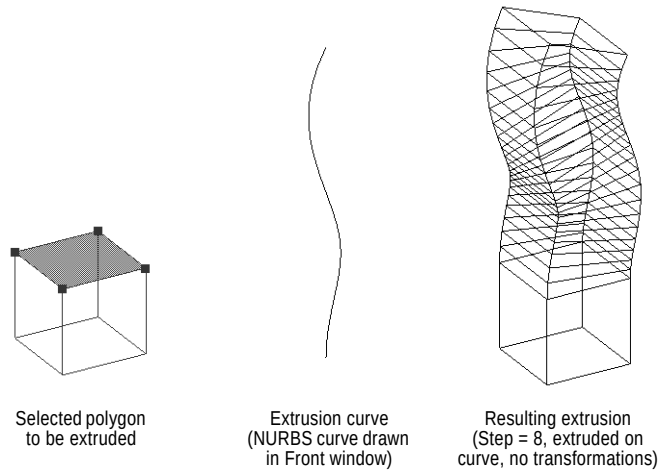
If the resulting extruded object is a polygon mesh, it can also be bevelled. The following illustrates the effects of beveling of a face.



If you activate the **Preferences → Create Modelling Relation** command first, you can use relational modelling concepts on most of the objects created using this command, (except for polygons extruded along a curve). For tips on how this affects operations in the **Surface** menu overall, see *Using Relational Modelling* on page 130 of this guide. Relational modelling is discussed in detail starting on page 143.

Extruding Polygons along a Curve

You can also extrude selected polygons of a polygon mesh object along a curve, with results that can look like this.



You can add a “twist and taper” effect when extruding polygons by specifying a rotation in y (twist) and scale in x and z (taper).

Note Relation modelling does not apply when extruding polygons along a curve.

This process is described in detail in *Extruding Polygons along Curves* on page 81.

Drawing the Curve

The window in which to draw depends on the axis of extrusion. For example, the Right window looks at the x-axis; if you wish to extrude the curve along x, draw the curve in the Right window. Similarly, if you wish to extrude the curve along y, draw the curve in the Top

window, and for extrusion along z, draw the curve in the Front window.

If you use Linear, Bezier, Cardinal, and B-Spline curves, you should draw the curve *counterclockwise* (or bottom to top, depending on the orientation you want). NURBS curves follow the more intuitive *clockwise* drawing direction (or top to bottom).

Extruding the Curve

1. Select the curve or face to be extruded.
2. If you wish to extrude along a curve, select the curve which serves as the extrusion path.

Note

You cannot extrude non-NURBS curves and faces along a NURBS curve. There are no restrictions to extruding NURBS curves and faces along a path of any curve type.

3. Choose the **Surface → Extrusion** command. The Extrusion dialogue box appears.

The image shows the 'Extrusion' dialog box with the following settings:

- Object Type:**
 - ☒ Polygon (Sub: 5)
 - ☐ Patch
 - ☐ NURBS
 - ☐ Linear
 - ☐ Cardinal
 - ☐ B-Spline
 - ☐ Linear
 - ☐ Quadratic
 - ☐ Cubic
- Beveling:**
 - ☐ Active
 - Radius: 0.250000
 - No. Bevels: 3
 - ☐ Top beveling
 - ☐ Side beveling
 - ☐ Bottom beveling
- Axis of Extrusion:**
 - ☐ X
 - ☐ Y
 - ☒ Z
 - ☐ On Curve
- Subdivision:** 1
- Depth:** 1.000000
- Step:** 1
- Polygon mode options:**
 - ☒ Curve placement
 - ☐ Transform
- Close:** ☒
- Buttons:** Cancel, Ok

4. Select Polygon, Patch, or NURBS as the resulting **Object Type**, as well as their associated parameters.
5. Select the desired **Axis of Extrusion**.
If you select **On Curve**, you cannot extrude a non-NURBS curve or face along a NURBS curve.
6. Fill in the **Subdivision**, **Depth**, and **Step** values.
7. Select **Close** to close the resulting polygon or surface object.
8. If you selected **Polygon** as the object type, you can fill in **Bevelling** values as required. To apply bevelling, the curve must be closed and the **Close** option must be selected.
9. Click Ok to apply the settings or Cancel to exit the dialogue box.
10. If you selected **On Curve**, pick the target curve.

The result is a 3D object of the type that you chose.

The original curve remains displayed in the window, overlaid by the new extruded object, and can be used again to create more objects.

If there is a relational modelling in effect and:

- If you have extruded the curve along another curve, both the generator curve and the curve of extrusion are in a modelling relation with the extruded object.
- You wish to separate the curve from the object for viewing purposes, select the generated object and translate it away from the generator curve. You cannot translate the generator curve away from the object because of the modelling relation.
- If you bevelled the extrusion, the normals are calculated using a non-averaging method. For more information, see *Keep Exactly Computed Normals* on page 880 of the *Reference Guide*.

Creating Extrusions with Custom Effects

There are several custom effects which also let you extrude curves to create 3D objects.

The **Surface** ➔ **GuidedExtrude** command in the Model module is an advanced modelling tool that allows you to extrude curves to create patch objects. You simply create a cross-sectional curve and then two additional curves, which act as guides or rails for the cross-section. The cross-sectional curve is automatically sized to conform to the guides during extrusion.

The **Path → Sweep** command in the Motion module allows you to create complex models by sweeping one curve along another. Unlike the standard extrusion on curve, Sweep allows you to animate the cross-sectional curve so that the resulting model reflects that animation. For example, a simple keyframed rotation applied to the cross-section curve automatically creates a surface with a twist in it.

In cases where the sweep path is closed and three-dimensional, Sweep adjusts the “twist” in the cross-section so that the beginning and end join smoothly.

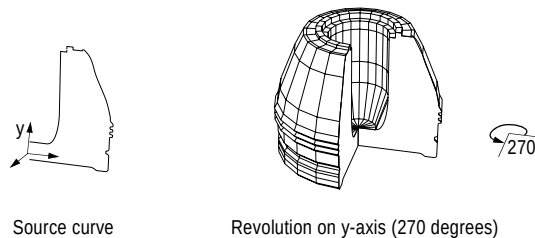
The **Surface → TBranch** command in the Model module performs branched extrusions to create polygon mesh objects. TBranch accepts a hierarchy of curves and an extrusion cross-section. The ends of each curve are checked to see if they lie near any other curves. If so, a T- junction is formed in the extrusion, with smooth curves connecting the stalk to the cross-piece.

The **Surface → Bevel** command in the Model module allows you to extrude curves or faces with user-defined bevels to create polygon mesh objects. You simply model the curve or face to extrude and the profile of the desired bevel. The effect automatically generates clean bevels with no crossed polygons.

For more information on using these effects, see their command descriptions in the *Reference Guide*.

Revolving a Curve about an Axis

The **Surface → Revolution** command in the Model module creates a 3D object out of a curve (open or closed). You can revolve the selected curve on any global axis or along another curve.



You can use relational modelling concepts on the objects created using this command. Always make sure that the **Preferences → Create Modelling Relation** command is activated first.

If **Preferences → Create Modelling Relation** is activated, the generated object is in a modelling relationship with *both* the original curve and the one you selected as the axis of revolution.

For tips on how this affects operations in the **Surface** menu overall, see *Using Relational Modelling* on page 130. For more details on using relational modelling, see *Relational Modelling* on page 143.

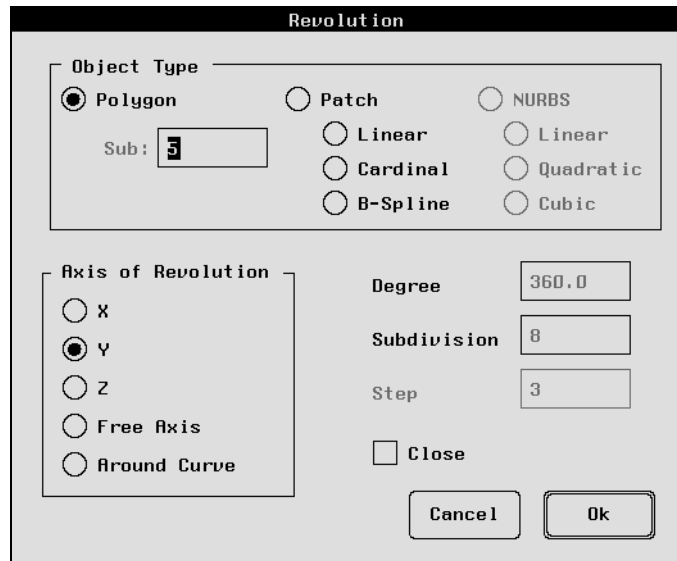
Drawing the Curve

The window in which to draw depends on the axis of revolution. For example, if you wish to revolve the curve around y, draw the curve in the Front or Right window.

If you're using Linear, Bezier, Cardinal, and B-Spline curves, draw the curve in a *counterclockwise* direction (or bottom to top, depending on the orientation you wish). When using NURBS curves, draw the curve in a *clockwise* direction (or top to bottom).

Revolving the Curve

1. Select a curve or face.
2. Choose the **Surface → Revolution** command. The Revolution dialogue box appears.



3. Choose the **Object Type** you wish to create by selecting Polygon, Patch, or NURBS and their associated parameters.

4. Select the **Axis of Revolution**, as described in the next section, *Selecting the Axis of Revolution*.
5. Set the degree of revolution in the **Degree** text box (range is 0 to 360°).
6. For all object types, set the number of **Subdivisions**.
7. If you have selected a patch or NURBS surface as the result, set the resolution of the resulting surface object in the **Step** text box.
8. Select **Close** to close the resulting polygon or surface object.
9. Click **Ok** to create the object or **Cancel** to exit the dialogue box.
10. If you selected **Around Curve**, pick the target curve.

Selecting the Axis of Revolution

You can revolve the curve around any of the global x, y, or z axes, around another curve, around itself, or around a “free axis.”

When you revolve around a **Free Axis** or **Around Curve**, the axis of revolution is based on the first two (visible) points of the curve selected as the axis of revolution. Only modifications made to the first two points of the generator affect the generated object.

When you choose **Around Curve**, you choose the curve of revolution after you exit the box. You could also revolve the selected curve around itself with quite different results.

When you choose **Free Axis**, SOFTIMAGE 3D creates a new default two-point Linear curve and displays it at the global origin, where it forms a 45° angle in the xy plane.

This two-point Linear curve is also displayed as a separate curve in the Schematic window. If there is a modelling relation in effect and you manipulate the free axis curve, you can edit the orientation of the generated object.

Results of a Revolution

The result of a revolution is a 3D object of the type that you chose. The original curve remains displayed in the window, overlaid by the new revolved object, and can be used again to create more objects.

If there is a relational modelling in effect and:

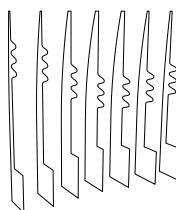
- If you have revolved the curve along another curve, both the generator curve and the curve of revolution are in a modelling relation with the extruded object.

For example, if you draw Curve 1, select **Around Curve**, and pick Curve 2, then both Curve 1 and 2 are now generators of the resulting revolved object.

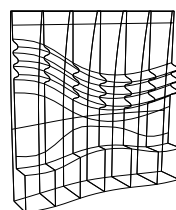
- If you are using relational modelling, you see the object simultaneously move and respond to your changes as you edit, transform, or scale the generator curve. The only operations that can be performed independently on the generated object are transformations and inversion of the surface normals.

Skinning Over Curves

The **Surface → Skin** command in the Model module allows you to create a 3D object from a series of curves or faces that define cross-sections of the object.



Source curves



Result of skinning operation



Skinning is also a good technique for joining objects together, such as joining an ear to a head in a realistic way (see *Skinning between Surfaces* on page 172).

You can use relational modelling concepts on the objects created using this command. Always make sure that the **Preferences → Create Modelling Relation** command is activated first.

For tips on how this affects operations in the **Surface** menu overall, see *Using Relational Modelling* on page 130. For more details on using relational modelling, see *Relational Modelling* on page 143.

Order of Selection

For all curve types, the order of selection in skinning affects the final result. For example, say you are skinning between four curves: c1, c2, c3, and c4.

If you select in this order:

c1 c2 c3 c4

the result is somewhat different than if you select in this order:

c2 c1 c3 c4

If you make a mistake in the picking order, middle-click on a curve to “unpick” all subsequent curves, something like a “clear all starting from this curve.”

Skinning Non-NURBS Curves

When skinning Linear, Bezier, Cardinal, or B-Spline curves:

- Curves must be either all open or all closed.
- The curves must have the same number of control points. It is easiest if you duplicate the first curve, move the duplicate to where you want to position the imaginary cross-section, then edit (without adding or removing) the points to build the new shape.
- The resulting object can be a polygon or patch, but not a NURBS surface.

Skinning NURBS Curves

- The resulting object can be a polygon mesh or a NURBS surface, but not a patch.
- Curves must be either all open or all closed and of the same degree (linear, quadratic, or cubic).
- You may successfully skin NURBS curves with a different number of control points and different parameterizations.

Note

Although you can skin over different number of control points, in an attempt to impose a common parameterization and number of control points, SOFTIMAGE 3D makes certain adjustments in the resulting skin.

Vertices and ribs are added, which may result in a cluttered appearance. For example, if you skin between three NURBS curves with 7, 8, and 11 control points respectively, the resulting NURBS surface has 18 points in the U direction, which may look cluttered.

Parameterization Options

You can select a parameterization for the resulting NURBS surface which is different from the parameterization(s) of the component curves.

If you want to preserve the interpolation of the curves, select Non-uniform parameterization for the skin surface (in the dialogue box discussed in next section).

If you want a specific parameterization for the skin surface, you can do so; however, the curves may not interpolate in the same way. So to avoid overcomplexity in the curves, apply Uniform parameterization to all curves and count the number of control points as before.

Procedure for Skinning

1. Draw a series of curves or faces representing the object to be reproduced at different depths (cross-sectional stages).
 - You cannot skin a mix of open and closed curves.
 - If you use Linear, Bezier, Cardinal, and B-Splines as generator curves, draw the curve *counterclockwise* (or bottom to top, depending on the orientation you wish).

If you make a mistake in the drawing direction of the curves and wish to reverse the direction of the surface normals, select the resulting skin and use the **Effect → Inverse** command.

2. Choose the **Surface → Skin** command.
3. Left-click to pick each curve in the desired order. As each curve is selected, it is highlighted.
4. If you make a mistake in picking:
 - Left-click to unpick the current curve.
 - Middle-click on any curve to clear all selections from that point.
5. When you have selected all the curves you wish, right-mouse click to display the Skinning dialogue box.
6. Select the resulting **Object Type** (polygon, patch, or NURBS) and the parameters for each one.
7. Set the other parameters:
 - **Step** is available for patch and NURBS surfaces, but not for polygon mesh objects.

- **Sub** and **Cap** are only relevant to polygon mesh objects.
- **Close** is available for any resulting object type.

8. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.

Result of Skinning

The result of skinning is a 3D object of the type you selected, either open or closed. The curves remain displayed on the screen after the new skinned object has been created, and can be used again to create more objects. Notice that the original curves and the skinned object are on the same spot.

If you are using relational modelling, you see the object simultaneously move and respond to your changes as you edit, transform, or scale the generator curve. The only operations that can be performed independently on the generated object are transformations and inversion of the object's surface normals.

Using Border Curves

The **Surface** → **Four-Sided** command allows you to create a patch or NURBS surface from four border curves.

If you use NURBS curves, the result is a NURBS surface. If you use Linear, Cardinal, Bezier, or B-Spline curves, the result is a patch surface of the same type as the curve.

You can use relational modelling on the resulting object. Always make sure that the **Preferences** → **Create Modelling Relation** command is activated first.

For tips on how this affects operations in the **Surface** menu overall, see *Using Relational Modelling* on page 130. For more details on using relational modelling, see *Relational Modelling* on page 143.

1. Draw two pairs of curves. Do not mix NURBS and non-NURBS curves.

Pairs of non-NURBS curves should be:

- Drawn *counterclockwise* (or bottom to top).
- The same type and have the same number of points.

Pairs of NURBS curves, each pair:

- Being of the same degree.
- Can have different number of points.

2. Position each pair (top-bottom, left-right) opposite the other.
The result is more predictable if the ends of the curves are close to each other.
3. Choose the **Surface ► Four-Sided** command.
4. If you are using NURBS curves, the Four Sided Setup dialogue box appears, in which you can indicate the parameterization of the resulting NURBS surface (non-uniform, chord length, centripetal, or uniform).
5. Pick curves one at a time:
 - Pick non-NURBS curves in any order.
 - Pick NURBS curves in a clockwise direction.

Each curve is highlighted as it is picked.

The result is a patch or NURBS surface.

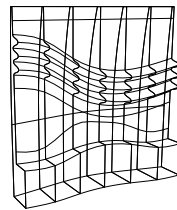
The curves remain displayed on the screen after the new four-sided object has been created, and can be used again to create more objects.

If you are using relational modelling, you see the object simultaneously move and respond to your changes as you edit, transform, or scale the generator curves. The only operations that can be performed independently on the generated object are transformations and inversion of the object's surface normals.

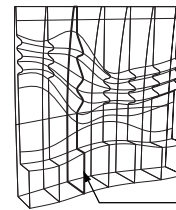
Relational Modelling

In relational modelling, there is a direct relationship between generators and an object which is generated from them. A *generator* can be a curve, a surface object, or a face, (a polygon mesh object, when you use **Effect → Inverse** on it).

When you modify a generator, such as by scaling or moving points, the changes are automatically reflected on the generated object. This makes editing models quick and easy, eliminating many time-consuming editing steps. Editing shape animation with relational modelling is also simpler.



Skin surface with source curves shown.



Modify one of the source curves.

Surface changes as any of the source curves are modified.

You can break the modelling relationship between the generators and the generated object at any time, and from that point manipulate each independently.

To use relational modelling, you must choose the **Preferences → Create Modelling Relation** command.

Which Operations Can Use a Modelling Relationship?

In the **Surface** menu, the **Extrusion**, **Revolution**, **Skin**, and **Four Sided** can create a modelling relation between the original curves and the object it generates.

In the **Draw** menu, the **Extract**, **Zip Surfaces**, **Project on NURBS Surface**, and **Trim NURBS Surface** operations can use modelling relations.

The **Effect → Inverse** command creates a modelling relation between the generated object and itself, which allows you to invert the object independently of the generator curve.

Some operations allow a “three-way” modelling relation. For example, if you revolve one curve around another curve to create a

3D object, then both curves are in a modelling relationship with the resulting objects' changes to either curve affect the resulting object.

Similarly, when you use a curve to trim a NURBS surface, the result is a new (trimmed) object. Both the curve and the original (untrimmed) surface are in a modelling relation with the trimmed surface.

Creating the Modelling Relation

1. Choose the **Preferences → Create Modelling Relation** command.
2. Draw a curve or get a primitive circle. The window in which to draw depends on the axis of extrusion, for example, if you want to create an extrusion. For example, if you wish to extrude the curve along y, draw the curve in the Top window.
3. As an example, apply any of the **Surface** commands to it, such as **Extrusion**.

The generator curve remains displayed in the window after the new extruded object has been generated, and can be used again to generate more objects.

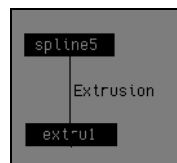
In the Schematic window (Model mode), a link between the generator curve and the generated object indicates that a modelling relation exists. If you freeze the relationship using **Effect → Freeze Modelling Relation**, the link disappears.

If you extruded the curve along another curve, both curves are linked to the resulting object in a modelling relationship. Any changes to either curve are reflected in the extrusion.

Viewing Modelling Relationships

To see the links between the generator(s) and the resulting object, open a Schematic window and select the Model mode.

The link between the generated object and its generator(s) is displayed and labelled, for example, such as an Extrusion.



Editing the Generated Object

Editing the Generator

Almost any operation that you perform on the generator is reflected on the generated object.

1. In the Schematic window, select any generator curve(s).
2. In any other window, edit the generator(s) or transform them (scale, rotate, translate) and you see the extruded object simultaneously respond to your changes.

This does not include operations which create new objects. For example, if you do a **Draw** ➔ **Cut** operation on a generator curve, the result is two new curves; this operation does not affect objects already generated from the curve.

Editing the Generated Object Directly

As long as a modelling relation exists between the generator(s) and the generated object, the only operations that can be performed independently on the generated object are:

- Transformations (scaling, rotation, and translation).
- Inverting the surface normals using **Effect** ➔ **Inverse**. This creates a modelling relation between the generated object and itself, and allows you to invert the object independently of the generator curve(s).

Anything else you do to the object is lost at the first screen refresh. For example, if you modify the generated object and then use the Supra key **r** to refresh or **u** for Undo, *only* its transformations (scaling, rotation, and translation) are retained.

If you wish to move a generator curve away from the generated object for viewing purposes, select the generated object and translate it away from the generator. You cannot translate the generator away from the object because of the modelling relation. If you do not need to edit the generator(s), you can simply hide them (see *Displaying and Hiding Elements* on page 21).

Creating Shape Animation

You can easily animate an object with relational modelling by animating its generator(s) instead of the object itself, and saving keyframes using the **SaveKey → Object → Shape** command. By animating the generator(s), you can avoid steps such as moving the object's points individually or creating clusters of points and then saving keyframes for the clusters. Using this method, creating animations such as winding snakes and rippling ribbons are less time-consuming. Simply extrude a curve, then move points on the generator and save keyframes. When you play back your animation, you can see the winding or rippling action on the extruded object.

Once you have completed your animation, you can remove the modelling relationship between the generator(s) and the generated object. You can also delete the generator(s) and save the animation to your object, as described in the next section, *Removing a Modelling Relationship*.

Removing a Modelling Relationship

You can break the modelling relationship between the generator(s) and the generated object at any time using the **Effect → Freeze → Modelling Relation** command. Once the relationship is frozen, modifications made to the generator(s) no longer affect the generated object and you can manipulate each independently.

This freeze affects only the selected object and does not remove all existing modelling relationships, nor does it undo any previous modelling steps you have performed on the related objects.

 To temporarily deactivate relational modelling for selected objects, you can also use the DopeSheet window. Select the generated object, open the DopeSheet, and click the green rectangle beside the object's name. The rectangle turns red to show its deactive. Now when you modify the generator, this object is not updated.

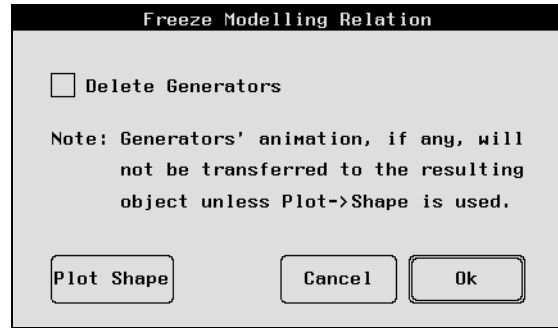
By contrast, if you deactivate the **Preferences → Create Modelling Relation** command, this deactivates the relational modelling mode but has no effect on existing modelling relations.

1. Select the generated object.
2. Choose **Effect → Freeze → Modelling Relation**.



This procedure is not reversible.

The Freeze Modelling Relation dialogue box appears.



3. Click Ok to remove the modelling relationship and exit the dialogue box.

See below for information on deleting generators.

Deleting a Generator

If you no longer need a generator, it can be deleted and its animation can be copied to the object if desired. You cannot undo the deletion.

If you delete a generator which is shape animated, you lose the animation on the generated object; however, if you select the **Plot Shape** button in the Freeze Modelling Relation dialogue box, the generated object keeps the animation of its generator even after the latter is deleted.

Before you delete, be aware of the following:

- You cannot undo the deletion.
- By deleting the generator(s), you also delete the modelling relationship.
- You cannot delete a generator which is in a hierarchy or which is used in another modelling relation.
- You cannot use this option to delete a generator curve if it was used to generate more than one object and at least one of those objects is still linked to it. In this case, you can use the **Delete → Selection** command to delete the generator, but this removes modelling relations for *all* its generated objects.

Procedure

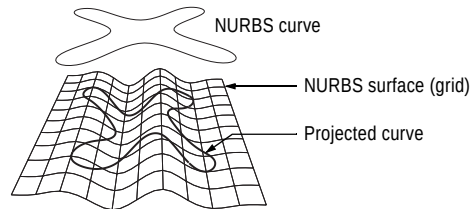
1. Select the generated object.
2. Choose the **Effect → Freeze → Modelling Relation** command. The Freeze Modelling Relation dialogue box is displayed, as shown previously.
3. Select **Delete Generators**.
4. If you are deleting an animated generator and want the animation copied to the object so that its animation is retained, click the **Plot Shape** button. If you delete the generator *without* selecting **Plot Shape**, the object is no longer animated. (For more information on shape plotting, see *Plotting an Animation* on page 75 in the *Animating User's Guide*.)

The Plot Shape dialogue box is displayed.

5. Enter the **Start** and **End** frames of the animation.
6. Choose the copy method: **Frame step** or **Number of copies** and set the value for either.
7. Click **Ok** to accept the settings and exit the Plot Shape dialogue box.
8. Click **Ok** to remove the modeling relationship and exit the Freeze Modelling Relation dialogue box.

Projecting Curves onto Surfaces

You can project a NURBS curve onto a NURBS surface then model with that curve, with or without affecting the underlying surface.



NURBS curve projected onto a NURBS grid.

There are two very similar commands for doing this: **Draw → Project on NURBS Surface** and **Draw → Trim NURBS Surface**.

The process of projecting curves onto NURBS surfaces is essentially that of projecting trim curves onto NURBS surfaces.

- Projecting curves allows you to fit a curve onto the surface and then model with that curve, without affecting the underlying surface at all.
- By contrast, trimming a NURBS surface allows you to sculpt the underlying surface by trimming it.

Modelling with Projected or Trim Curves

Using a combination of modelling techniques on projected curves and surfaces, you can achieve some quick and interesting results. For example, after projecting a curve onto a surface, you can extract the curve from the surface, extrude it, and modify the generated objects using relational modelling.

For information and detailed procedures related to projected curves and trim curves, see *Modelling NURBS Surfaces with Projected Curves* on page 185.

Creating Faces

A face is a 2D object made of curves. A face with holes in it is made up of an exterior curve and interior curves (the holes). As a 2D object, faces can be rendered; they can also be extruded or revolved to make 3D objects.

It may seem unnecessary to use planar 2D objects in 3D space, but faces are quick to render and have many applications; for example, all text fonts in SOFTIMAGE 3D are faces (see *Creating Text* on page 153).

Making Faces!

You cannot draw a face object as you do a curve, or “get” a default face as you do a polygon mesh or surface object. Instead, you start by drawing a curve, closing it, and then converting it to a face.

1. Draw or select a curve, or use **Get → Primitive** to get a predefined primitive curve such as an arc or a circle.
2. To close the curve, choose the **Draw → Open/Close** command.
3. Choose the **Draw → Convert to Face** command.

The curve becomes a face.



The original curve is deleted *unless* it is already being used as an animation path, a path of extrusion, or for curve deformation.

In turn, faces can be converted to a 2D polygon mesh or to a Linear or NURBS surface using the **Effect → Convert** command.

Making Holes in a Face

The **Draw → Attach Hole** command creates holes in a face (ouch!). Starting with closed curves, you can convert one to a face and attach the other(s) as a hole(s). The curve you attached is preserved.

Faces with holes can also be extruded and revolved.

When attaching a hole, the following rules and tips apply:

- The curve which represents the hole must be inside the face. The hole cannot partially intersect the surface or be located outside the surface. If you do so, a message advises you of this error.

- For this operation, do not mix NURBS with other types of curves (Linear, Bezier, B-Spline, or Cardinal).

Procedure

1. Create the face to which you want to attach one or more holes.
2. Draw one or more curves you want to define as holes in the face and close each one.
3. Position the hole curves in relation to the face, making sure that:
 - The hole does not intersect the face boundaries, as described in the previous paragraphs.
 - The hole is in the same plane.
4. Select the face.
5. Choose the **Draw → Attach Hole** command.
6. In sequence, pick all curves which are to be holes.

The face and attached holes remain highlighted as more holes are added on the face. If you subsequently access the **Info → Selection** for the face, its related dialogue box indicates that it has holes.

After the Attach operation, the original curve is preserved and remains in the same spot as the newly attached hole. This superimposition may cause confusion later when you try to delete or detach the hole in the face. You may unknowingly select the original curve instead of the hole, and get an error message “You have to pick a hole.”

To avoid this problem, you should move or delete the original curve (use the Schematic window to select it) before performing operations on the face.

Deleting or Detaching Holes

The **Draw → Detach/Delete Hole** command allows you to detach and (optionally) delete holes you previously made in faces using the **Draw → Attach Hole** command.

1. Select the face which has a hole you want to delete or detach.
2. Choose the **Draw → Detach/Delete Hole** command.
3. Middle-click on the hole to delete it entirely, or left-click on the hole to detach it from the face.

When you detach a hole, a new curve is created and remains in the same spot on the face from which you detached it. In the Schematic window, another curve appears.

4. Right-click to end the mode.

Extruding or Revolving a Face

Since a face is simply a closed curve converted to a 2D planar object, you can extrude or revolve a face to make a 3D object (such as turning 2D text into 3D text) using the same process as that of extruding or revolving a curve.

For information and procedures, see *Extruding a Curve* on page 131, and see *Revolving a Curve about an Axis* on page 135.

Converting a Face

Using the **Effect → Convert** command, you can convert a face to:

- A 2D polygon mesh
- A face made from a Linear type curve
- A NURBS face



If you start with a NURBS face of quadratic or cubic degree and choose the Linear face option, the result is a NURBS face of linear degree.

You can control the resolution of the newly created object, both the exterior contour and any holes in the object.

After a conversion, the original face is preserved.

1. Select a face.
2. Choose the **Effect → Convert** command. The Convert Face dialogue box is displayed.
3. Select either **Polygon**, **Linear Face**, or **NURBS**.
 - If you select a B-Spline face, for example, you can convert it to a cubic NURBS face or a face made of a Linear curve.
 - If you select a NURBS face of higher degree (quadratic or cubic), and choose the **Linear Face** option, the NURBS face is converted to a NURBS face of linear degree.
 - If you select the **Polygon** option, the face is converted to polygon mesh face.

4. To change the resolution of a curve (either a **Hole** or the **Exterior** curve), select the curve in the scroll box and enter a new **Step** value. The **Step** value applies only to the selected curve. To modify the **Step** value for all curves, you need to repeat the conversion procedure.

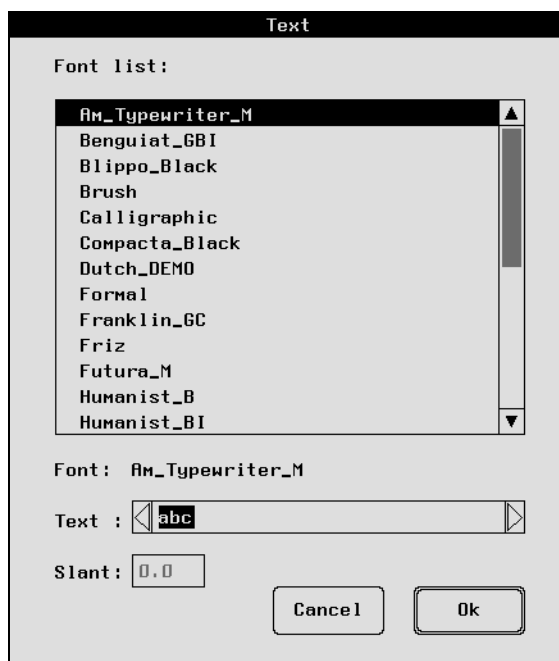
If you are satisfied with the shape of all curves of your original face, then leave the **Step** value near the default of 10.

5. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.

Creating Text

When you create text, each letter is actually a 2D face which can be extruded into 3D text using the **Surface → Extrusion** command.

1. Choose the **Get → Text** command. The Text dialogue box appears.



2. Select the font you wish to use from the **Font list**.
3. You can also enter the **Text** string you wish to create. For special characters, see the *Sample Character Set* in the *Reference Guide*.
4. To create italicized text, enter a value such as 10 in **Slant**.

5. Click Ok to exit the dialogue box or Cancel.

Your string of text appears using the font you selected.

The process of extruding a face (such as text) is the same as that of extruding a curve, which is described on page 131. The related dialogue box offers you the option of bevelling the extruded letters.

Using Custom Effects to Create a Font

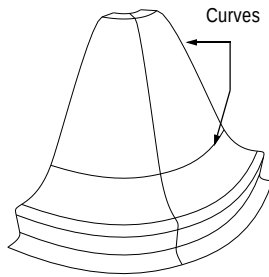
These three effects allow you to create, store, and use your own character fonts. Once a font is designed and saved with the **Save → SaveEffectsFont** command in the Model module, you can immediately produce text in SOFTIMAGE 3D with the **Get → EffectsText** command in the Model module. Fonts can also be manipulated and saved again with **Get → GetEffectsFont** command in the Model module. A default font is supplied.

For more information on using these effects, see their command descriptions in the *Reference Guide*.

C H A P T E R F I V E

Patch and NURBS Surface Objects

Introduction



NURBS surface object

A surface is made up of smaller surfaces bounded by parametric curves. With patch and NURBS surface objects, points are joined by curves to create a curved surface.

Points on surfaces have u and v parameters or coordinates, which measure along the surface itself. As a result, surfaces have both u and v directions, which can be opened or closed, cut, inverted, have rows added to them, and so on. The u direction is analogous to longitude and the v direction is analogous to latitude.

Surfaces are ideal for modelling round or curved objects because fewer points need to be defined than using a polygon mesh. The objects look smoother and do not have a fixed definition, and you can change their resolution to either increase detail or reduce rendering time. Texture mapping on a surface object is also more flexible on surface objects because they allow uv mapping.

Tasks which apply to both patch and NURBS surface objects are described in this section. Tasks which can be performed only with NURBS surfaces are described in *Modelling NURBS Surfaces with Projected Curves* on page 185.

Patch Surfaces

A patch surface can be made of either Linear, Bezier, Cardinal, or B-Spline curves in either the u or v direction. This means that you can have a patch surface which is Linear in the u direction and Cardinal in the v direction, for example.

NURBS Surfaces

NURBS surfaces are made of NURBS curves (linear, quadratic, or cubic degree) in either direction. A NURBS surface can be of different degrees in the u and v directions, such as quadratic degree in u and cubic degree in v .

However, you cannot mix NURBS and non-NURBS curves (Linear, Bezier, Cardinal, and B-Spline curves) on a surface: a surface object is by definition either a patch or NURBS surface.

NURBS surfaces have the same properties and behaviour as NURBS curves, described in detail in the *Curves* chapter, including the choice of different parameterizations.

Parameterization

NURBS' parameterization is discussed in detail in *Parameterization* on page 97. The options are Non-Uniform, Chord Length, Centripetal, and Uniform.

If you select a predefined primitive object such as a sphere or torus and define it as a NURBS, it has uniform parameterization by default.

If you create a NURBS surface by revolving a NURBS curve, it has the same parameterization as the curve used to build it.

In either case, you can change the parameterization using the **Effect** → **Reparameterization** command.

Effects of Parameterization

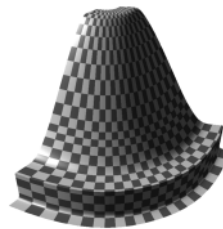
Any non-uniform parameterization (Non-Uniform, Chord Length, or Centripetal) offers the following advantages over uniform parameterization.

Better Texture Mapping

As an example, say that an object's surface has Uniform parameterization, but the knots are unequally spaced so that the curve segments are unequal in length.



Uniform parameterization



Chord length parameterization

When you apply a texture map, the texture is mapped equally to surface segments which are actually unequal in size, resulting in the texture on the larger segments appearing stretched. Each surface segment is treated equally and given an equal “piece” of texture, but the piece of texture has to fit over a surface segment which is larger.

By contrast, an object with Chord Length or Centripetal parameterization permits more flexible texture mapping, in which each piece of the surface is given a “piece” of the texture according to

its relative size – bigger segment, bigger piece – such that the texture looks more uniform.

NURBS surfaces are based on NURBS curves, so information pertaining to NURBS curves is also valid for NURBS surfaces. For more information, see *Some Basics about NURBS Curves* on page 94.

Control of Surface Shape

In SOFTIMAGE 3D, you can edit and subdivide NURBS surfaces without changing the shape of the whole surface. This provides more (localized) control, and is the major difference between NURBS surfaces and patch surfaces.

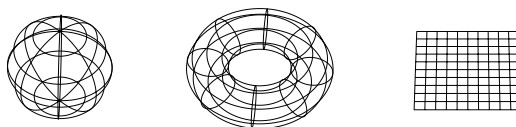
There are slight differences in control of surface shape among Chord Length/Centripetal parameterizations and Non-Uniform parameterization. For more information, see *Parameterization* on page 97.

Creating Surface

There are two basic ways of creating a patch or NURBS surface object: by choosing a predefined primitive object or by creating a curve and making a surface object from it.

Selecting a Predefined Primitive

You can create a primitive Sphere, Torus, or Grid as a patch or NURBS surface object.



Primitive patch or NURBS surfaces: sphere, torus, and grid.

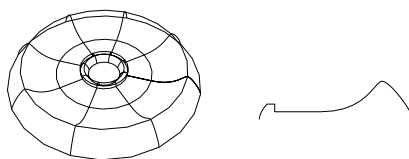
1. Choose **Get → Primitive** and the primitive type, such as Sphere. The related dialogue box appears, with one or more parameters to set the initial size or step value of the object.
2. Select Patch or NURBS as the object type.
3. Fill in the parameters or accept the default values.
4. Click Ok to accept the settings or Cancel to exit the dialogue box.

Making a Curve into a Surface

Circle, Arc, Square, and Spiral primitives are defined as curves, but when they are extruded or revolved, the resulting object can be defined as a patch or NURBS surface.

You can also create your own custom 3D shapes, starting with curve profiles.

- Draw a curve profile using the **Draw → Curve** commands, then use the **Surface** commands to extrude or revolve the curve into a 3D surface.



NURBS surface from revolution

Source curve

- Draw several curve profiles and use the **Surface** → **Skin** or **Four Sided** commands to create a surface from them.

Note

Starting with a NURBS curve, the resulting 3D object can be a NURBS or a polygon mesh.

Starting with a non-NURBS curve, the resulting 3D object can be a patch or a polygon mesh.

For more information and procedures, see *Making 3D Objects from Curves* on page 130.

Creating Patches with Custom Effects

You can also create a patch object from scratch using the **Get** → **Primitive** → **PatchFormula** command in the Model module. This effect creates a patch according to precise, user-defined formulas. You enter a system of equations defining the x, y, and z coordinates of the patch's points as functions of their patch coordinates u and v. These equations are composed of normal mathematical symbols and functions such as +, -, *, SIN, COS, PI, TAN, SQRT, and RAND.

For more information on this effect, see its command description in the *Reference Guide*.

Opening and Closing Surfaces

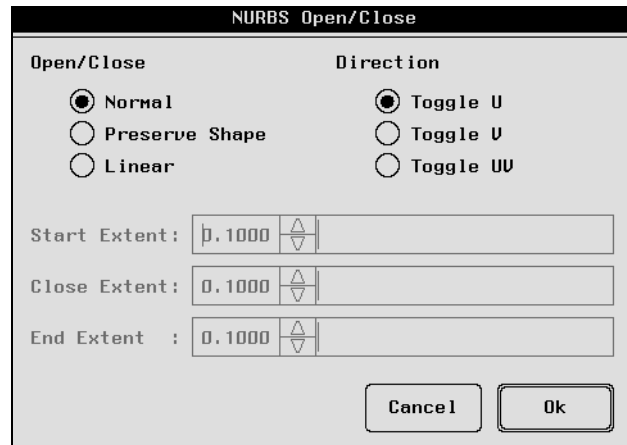
In general, patch and NURBS spheres are open in v and closed in u upon creation, toruses are closed in both directions, and grids are open in both directions.

In some cases, you must open or close a surface before you can perform some other operation. For example, you can only merge two surfaces when they're open in u.

1. Select a patch or NURBS surface.
2. Choose the **Draw** ➔ **Open/Close** command.
3. If you selected a patch surface, the Patch Open/Close dialogue box is displayed.

You can only choose whether the operation is performed in the U or the V direction.

4. If you selected a NURBS surface, the NURBS Open/Close dialogue box appears.



5. If you are closing the surface, you can choose the shape of the closing segment, or whether to reinterpolate the whole surface or preserve the shape of the unaffected portions. You can **Toggle** the direction (u or v) in which the surface is opened or closed.
6. If you select **Preserve Shape**, the **Start**, **Close** and **End Extent** sliders become available. As you adjust the sliders, the shape of the closing segment changes interactively.

You can change the method of closing the surface and see the results, before you exit the box.

7. Click Ok to exit the dialogue box or Cancel to exit without accepting the settings.

Tips and Restrictions

You can close or reopen the surface, but it is not advisable to change the closure method (Normal, Preserve Shape, Linear) because the results may be unpredictable.

For example, if you close a surface with the **Normal** selection, you can reopen it, but you shouldn't change to **Linear** or **Preserve Shape** because the shape of the original surface will likely change unpredictably.

You cannot open or close a *trimmed* NURBS surface. If you wish to do so:

1. Delete the trim curves from the surface using **Draw → Delete Trim Curves**. If you have deleted the original curve used to trim, you may extract the trim curve before deleting (using **Draw → Extract**) and reuse it for projection.
2. Open or close the NURBS surface.
3. Trim the NURBS surface again.

You can also close surfaces in u or v using the **Effect → Curve Controls** command, which allows you to shift the point of opening or closing at the same time for a somewhat different result than using **Draw → Open/Close**. For more information see the next section, *Shifting the Point of Opening or Closing a Surface*.

Result

If you later choose **Info → Selection** or **Effect → Curve Controls** for that surface, the dialogue boxes indicate whether the surface is open or closed in u or v.

To modify point at which the surface opens, choose **Effect → Curve Controls** as described in the next section.

See also the **Draw → Cut** command for a way of slicing off a piece of a surface, but preserving the original.

Shifting the Point of Opening or Closing a Surface

You can choose the **Effect → Curve Controls** command and use the **Shift U** or **Shift V** slider to shift the “starting point” of a surface, thereby shifting the point at which the surface opens.

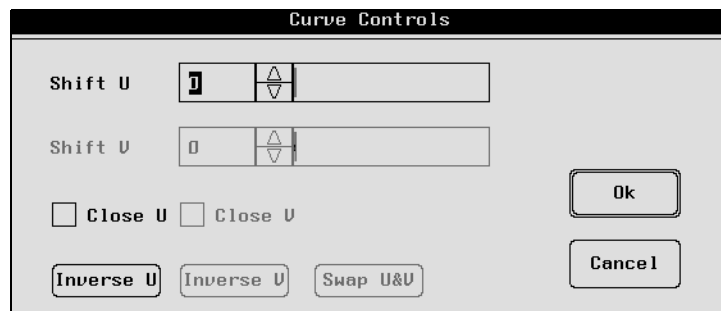
This may be useful in such operations as merging open surfaces, if the open segments of the objects do not match up.

You can also invert the direction of the surface curves to invert surface normals.

Procedure

1. Select a surface.
2. Select the **Effect → Curve Controls** command.

The Curve Controls dialogue box appears.



On the selected surface, the U is highlighted by the red edge flag on the object; the V is highlighted by the green edge flag.

- To open or close in U or V, select or deselect the **Close U** or **Close V** options.
- To shift the joining point in u or v, use the **Shift** sliders.
- To reverse the direction of the surface normals, use the **Inverse U** or **Inverse V** options.
- To exchange the U and V directions, click **Swap U&V**.

The results are displayed interactively.

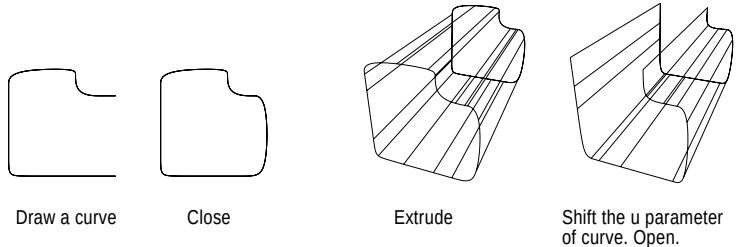
3. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.

Example

Let's say you extruded a 3D object from a closed curve. Now you want to open or close the extruded object at a different point.

1. Turn on relational modelling (**Preferences → Create Modelling Relation**).
2. Draw a curve and close it.
3. Extrude the curve to create a 3D object (**Surface → Extrusion**).
4. If you want to close or open the extruded object at a different point, select the generator curve and choose **Effect → Curve Controls**.
5. In the Curve Controls dialogue box, use the **Shift U** slider to offset the open/close point on the curve. If you deselect the **Close U** option, you can open the curve at this new opening point.
6. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box

The object created by the extrusion is also opened at the desired location.



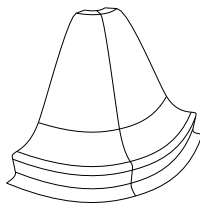
Converting Surface Objects

Modelling with surface objects allows you to apply operations not available with polygon mesh objects, such as merging objects, extracting curves from the surface and modelling with those curves, trimming, texture mapping, and so on.

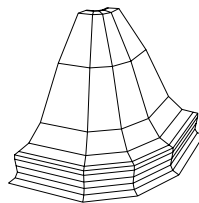
On the other hand, some useful options are available only for polygon mesh objects, such as if you wish to:

- Use your model in games development, on platforms which support only polygon mesh models.
- Optimize complex models using the Polygon Reduction effects.

Using the **Effect → Convert** command, you can convert a NURBS surface to a polygon mesh, or convert a patch to a NURBS or a polygon mesh.



NURBS surface



Converted to polygon mesh object

1. Select a patch or NURBS surface.
2. Choose **Effect → Convert**.
3. If you select a NURBS surface, it is converted immediately to a polygon mesh.

If you select a patch surface, the Convert Patch dialogue box appears in which you can choose to convert it to either a polygon mesh or a NURBS surface.



It is not possible to convert polygon mesh objects to surface objects except by using the **Effect → Web** or the **Deformation → Shrinkwrap** command, as described in the *Reference Guide*.

Custom Effect for Converting Surfaces to Polygon Meshes

The **Shape → Patch2Mesh** command in the Motion module lets you convert patches to polygon mesh objects. This effect is similar to the **Effect → Convert** command, but can be used on animated patches to obtain an animated duplicate polygon mesh. For more information on using this effect, see its command description in the *Reference Guide*.

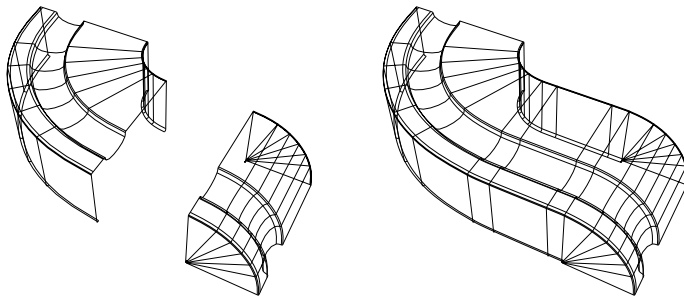
Joining Surfaces

Surfaces can be joined using a number of techniques:

- Merging surfaces creates a single new object which is the combined result of the originals.
- Zipping two surfaces does not join the two objects, but simply maps points on one surface to points on the other.
- If you create a profile of curves between two surfaces allows, you can Skin curve profiles between two surfaces, such as an ear and a head to make a smoother connection.
- Various custom effects.

Merging Surfaces

The **Draw → Merge Surfaces** command in the Model module allows you to join two patch or NURBS surfaces in order to create a new (merged) surface.



Two NURBS surfaces.

Resulting merged NURBS surface.

Depending on the type of surfaces to merge, the operation is executed immediately or opens up an interactive dialogue box.

Some options in these dialogue boxes preserve the shape of the original surfaces and add a new region between them. This operation is known as *blending* and the new region is called the *blend region*.

Others change the existing surfaces to *merge* them. For purposes of simplification, the general term *combining* is used in the following descriptions.

Contrast this command with **Draw ➔ Zip Surfaces**, which zips points on one surface to points on another, but does not create a new object.

For All Surfaces

- You can only combine two surfaces at a time and the merge must be along *open* edges.
- You can combine two patch surfaces or two NURBS surfaces, but not one of each.
- The curves of both surfaces to combine must be of the same degree in U and in V. For example, you can combine two NURBS which are both cubic in U and linear in V, or you can combine two patches which are both Cardinal in U and B-Spline in V.

For Individual Surface Types

- Patch surfaces must have the same number of control points in the direction of the merge.
- NURBS surfaces can successfully be combined with the different number of control points.
- SOFTIMAGE 3D allows you to import NURBS curves of a degree greater than 3 (cubic). These curves of higher degree may not have all the options available.
- Linear and Bezier type surfaces are combined immediately. NURBS, Cardinal, and B-Spline surfaces require that you fill in the relevant dialogue box. In the case of NURBS, the dialogue box is interactive.

Procedure

1. If the surfaces are closed, open them in the same direction (either U or V) using the **Draw ➔ Open/Close** command.
1. Position the surfaces as desired, so that open edges are appropriately positioned.
2. Choose the **Draw ➔ Merge Surfaces** command.
3. Pick an edge of the first surface. The edge is highlighted.
4. Pick an edge of the second surface.

- If you are merging two patch surfaces, the Merge Patches dialogue box is displayed. Select the **Last & first row**, or the **Number of rows** with the **Row interpolation factor**, and click Ok.
- If you are merging two NURBS surfaces, the NURBS Merge/Blend dialogue box is displayed. Set the **Blend Type** (C0, C1, C2, Factor, or Join second to first) and the **Blend** parameters (the **Start**, **Blend**, and **End Extent** sliders).

In the case of NURBS, the dialogue box is interactive.

For a description of the parameters of each dialogue box, refer to the **Draw → Merge Surfaces** command in the *Reference Guide*.

5. Click Ok to accept the settings or Cancel to exit the dialogue box.

The original surfaces remain as they are, and a set of combined surfaces is created.

Zippping Surfaces

The **Draw → Zip Surfaces** command in the Model module allows you to “zip” or fit one surface to the shape of another (support) surface by mapping points on the zip surface to points on the support surface.

Both surfaces must be of the same type: patch to patch or NURBS to NURBS. Within each type, variants can be zipped; for example, you can zip a B-Spline patch and a Linear patch or a quadratic NURBS and a cubic NURBS.

Zip Surfaces does not create a new object. Instead, the support surface is unchanged, but the points of the zip surface are zipped to points on the support surface.

Contrast this with **Draw → Merge Surfaces**, which creates a new (merged) object and leaves both source objects unchanged.

Tips

- For best results, the U and V lines should have the same orientation and relative position on both objects; for example, the V lines are in the same direction and the green highlight marking the beginning of the V line has to be at the top or bottom on both objects.
- Normals on both objects must be in the same direction.
- It is best to handle these adjustments *before* trying to zip.

You can use relational modelling concepts on the zipped object. Relational modelling is discussed on page 143.

Procedure

1. Select the two surfaces you want to zip together (use **Multi** menu cell or branch/tree select for patches in a hierarchy).
2. Choose the **Draw → Zip Surfaces** command.
3. Pick a point on the *supporting* surface using the left or middle mouse button. This point defines the origin of the “zipper” and, along with other points in the same row, is highlighted.
4. The Zipper Direction dialogue box is displayed, with two options that allow you to set the **zipper direction** in either the U or V direction. Select the desired zipper direction for the supporting surface.
5. Click Ok to continue or Cancel to return to step 3.
6. Pick a point on the surface you want to modify using the left or middle mouse button.

This highlighted point and others in the same row define the origin of the “zipper.” The Zipper Direction dialogue box is displayed again.

7. Select the desired zipper direction in which the surface is to be modified.
8. Click Ok to continue or Cancel to return to step 6. The Zip Patches dialogue box is displayed.
9. Set the parameters to define the number of points to zip together:
 - **Rows up and down** defines the number of rows zipped (on the zip surface) in the direction *opposite* to the direction picked on the support surface.
 - **Points up and down** defines the zipping in the *same* direction as the direction picked on the *zipped* surface.

The zipper is displayed interactively.

10. Click Ok when you are satisfied with the zipper, or Cancel to exit the dialogue box.

Skinning between Surfaces

You can take two surfaces, extract curves from each surface, place the extracted curves between the surfaces and skin the contour you have created, so as to join the two surfaces.

For example, you could create a head and an ear and then blend the two surfaces together using the properties of NURBS curves and surfaces and several different modelling operations. For details of the process used, see *Examples of Modelling with Projected Curves* on page 199.

Using Custom Effects to Join Surfaces

The **Effects** → **Shoe** command in the Motion module allows you to glue one entire hierarchy to another hierarchy. This can be useful for filling in gaps between models in a hierarchy, such as the fingers and palm of a hand.

The **Effects** → **Glue** command in the Motion module allows you to “glue” tagged points of one object to the tagged points of another surface object. You can apply the Glue effect to polygon meshes, patches, NURBS surfaces, or curves.

Glue is a motion effect, so either of the models can be animated, and the glued portions will stick together. Falloff parameters enable the effect to be applied smoothly to nearby points.

For more information on using these effects, see their command descriptions in the *Reference Guide*.

Extracting a Curve from a Surface

The **Draw → Extract** command extracts a curve from a surface. The extracted curve will be the same type as that used to create the surface object from which it is extracted.

If you want to model a shape extending out from a surface, or attach two parts of a large model, such as attaching an ear to a head in a smooth, realistic fashion, extracting a curve from the surface object is often the first step in your task.

An extracted curve can be extruded, revolved, transformed, skinned, or modelled with any other operation that you can do with a curve.

For a modelling example which uses extracted curves, projected curves, and several other modelling operations, see *Examples of Modelling with Projected Curves* on page 199.

1. Select a surface object.
2. Choose the **Draw → Extract** command.
3. Pick any u or v line on a surface object.
4. If you are working with a patch surface, you can only extract one of the u or v lines on the surface.
 - Pick the curve to extract on u or v. Your selected curve is highlighted (in white) and the patch is deselected.
 - Right-click to exit the mode.
 - To pick another curve, reselect the patch surface and repeat the operation.
5. If you are working with a NURBS surface, the NURBS Extract Curve dialogue appears.
 - Using the **Location** slider, you can extract a curve in the **U direction** or the **V direction** anywhere on the NURBS surface, not just at the existing u or v curves displayed.
 - Using the **Trim and projected curves** listed in the scroll box, you can extract a trim or projected curve if there are any on the NURBS surface.
6. Click **Extract**. You can repeat the process extract as many curves as you want before clicking **Exit** to exit the dialogue box.

Using Relational Modelling on Extracted Curves

You can also use relational modelling concepts on the extracted curve and thus on any objects created from that curve.

When an extracted curve is in a modelling relation with the surface from which it is extracted, you automatically modify the shape of the extracted curve by editing the original surface.

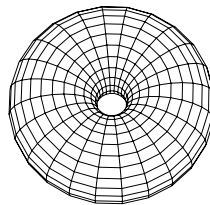
To use relation modelling, make sure that the **Preferences → Create Modelling Relation** command is activated first. Relational modelling is discussed on page 143.

Cutting a Surface

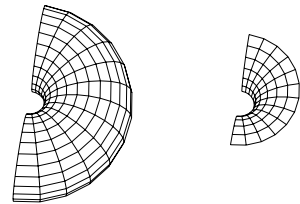
Here's an easy way to take a piece out of an object without changing the original object.

1. Using the **Draw → Cut** command, you can:
 - Cut a NURBS surface at an exact position.
 - Cut open patch surfaces at a u or v line or at the control point closest to where you click the mouse.
2. You can continue cutting until you end the mode.

At each iteration, the current selection is preserved. For example, cut a torus in half in U and the original (whole) torus is preserved. Cut the half torus in half, and the whole and half toruses are preserved as well as the quarter torus, which is currently selected. A good way to halve your donut, and keep it too!



NURBS torus



Results of successive cuts.

By contrast, you could use the **Draw → Open/Close** command to open the torus in u for example; in that case, however, you would be left with a “bite” out of the donut since the entire segment is removed.

1. Select the surface you want to cut.
2. Choose the **Draw → Cut** command.
3. For patch surfaces:
 - Use the left or middle mouse buttons to cut the surface.
 - Right-click to end the mode.
4. When you select a NURBS surface, the NURBS Surface Cut dialogue box appears, allowing you to choose an exact location for the cut.
 - Use the **Location** slider(s) to interactively position the magenta point or curve in the **U** or **V** direction.

- Click **Cut** to perform the cut or **Exit** to close the dialogue box.
- 5. The original surface is preserved, in the same position as the cut segment. To see the results, translate one away from the other.

Editing a Surface

In general, all the editing operations for curves apply to surfaces as well. You can:

- Add or delete rows of control points to the surface.
- Modify the shape of the surface, locally or by using proportional editing.
- Shift the starting point of a closed curve, thereby relocating the point at which the curve can be opened or closed.
- Shift the beginning point of the surface.
- Invert the surface direction.
- Edit the surface resolution.
- Cap the surface

Adding Rows of Points

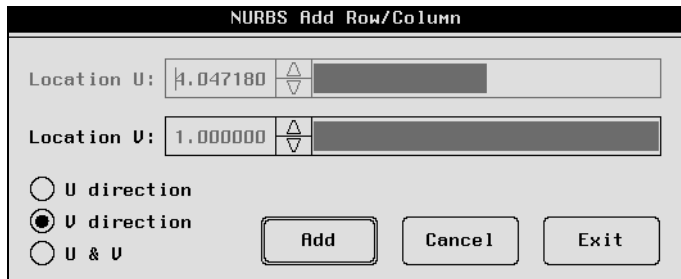
If you pick a location between two points on a line in U and choose **Edit → Add Point**, you add a row of points in U between those two points.

For patches, rows of points are added halfway between two selected rows. For NURBS surfaces, such rows can be added and positioned at an exact location on the surface, using an interactive dialogue box.

Procedure

1. Select the surface object.
2. Choose the **Edit → Add Point** command.
3. If you select a patch:
 - Pick the location on a line in the direction (U or V) in which you want to add the row of points.
 - Click any mouse button to add the row of points.

4. If you select a NURBS surface, the NURBS Add Row/Column dialogue box appears.



Note This dialogue box is similar to the NURBS Surface Subdivision dialogue box; the sole difference is that the subdivision process allows you to select a degree of continuity for the new row which may be different than the rest of the surface curves.

A magenta curve on the surface highlights your current position in U or V.

5. Select the **direction** (U, V, or both) in which you want to add the new row of points. For example, to travel along the U direction, select the **U direction** option and use the **Location U** slider to choose exactly the surface position in which to add the row of points. In response, the magenta guideline moves along the object's u direction to mark the currently selected spot.
6. Click **Add** to add the row of points at the desired spot.

You can continue adding rows in this fashion until you exit the dialogue box.

Example

This vase was created using a Cardinal curve of five points and revolving it using the **Surface → Revolution** command. The resulting object is a patch surface.



Patch surface.



Two rows added.

It was easy to create an object using a curve of only a few points, but you may wish to increase the resolution to be able to model in finer detail. In this case, choose **Edit → Add Point** and middle-click to add two rows in *v* to increase the resolution of the vase, compared to that of the curve profile used to generate it.

Since it is a patch vase, the profile of the vase is slightly changed (around the “waist”) by adding the row.

Subdividing a NURBS Surface

Another way of adding a row of control points to a NURBS surface is using the **Effect → Subdivision** command. This differs from using **Edit → Add Point** only in that it allows you to select a degree of continuity for the new row which may be different than the rest of the surface curves.

Continuity

Technically, you are really editing the continuity of knots on the surface, but since these are not visible to you, it is done by interacting with the surface control points.

The points in the new row cannot have a higher degree of continuity that the surface has as a whole: for example, if the surface is composed of quadratic NURBS (C1) in *U*, then the new row of control points can have C0 or C1, but not C2.

Whatever type of continuity you choose, the shape of the surface remains the same when you add the row of points. However, as you move any point in the new row, the degree of continuity you selected determines the shape and relative smoothness of the surface at that point.

Procedure

The more rows you add in one area, the more localized your control when editing.

1. Select a NURBS surface.
2. Choose the **Effect → Subdivision** command. The NURBS Surface Subdivision interactive dialogue box appears.
3. Select the **direction** (*U*, *V* or both) along which to subdivide the surface. A magenta curve appears, showing your choice.
4. Use the **Location** slider to position the curve along a surface.

5. Select the degree of **Continuity** for the new control point. The maximum value of the slider depends on the degree of the surface: for linear, 0 only; for quadratic, 0 or 1; for cubic degree, 0, 1, or 2. If you enter an invalid value (such as 4), it is ignored and the maximum value for the current surface curve is used instead.
6. Click **Subdivide** to accept the settings. Click **Exit** to exit the dialogue box or **Cancel** to exit without accepting the settings.

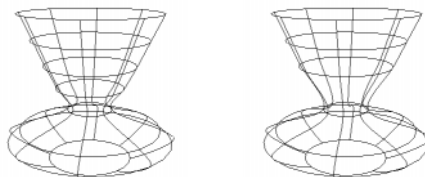
Deleting Rows

By deleting rows on a surface, you change the shape of the surface on a more global scale than by deleting points here and there.

1. Select a surface object.
2. Choose the **Edit → Delete Point** command.
3. Use any mouse button to pick a point to be deleted.
4. Either the Patch or NURBS Delete Row/Column dialogue box appears, depending on your selection of a patch or NURBS surface object.
5. Select the **direction** (U or V) in which you want to delete the row of points and click **Ok**.
6. On patch surfaces, if you delete a row made of a Bezier curve, you need to delete three such rows to maintain patch integrity. A message appears in the status line telling you how many rows to delete.
7. Press **Esc** to end the mode.

Example

In the case of our vase, when we added two rows in U, the waist appears a little too “nipped in.” To delete a row at the waist, choose **Edit → Delete Point** and click on the row that was added at the waist to delete it.



Before

After deleting one row.

Modifying the Shape of Surfaces

The most common methods of editing points on surfaces are:

- Moving individual points: use the **m Supra** key and drag the point to a different location. As you drag, the selected point is highlighted in green.
- Using proportional editing for a more subtle result.
- Tagging points and creating clusters, then transforming the cluster as a group.

These methods are discussed in detail in *Editing Points* on page 42.

Example

For our vase, you could tag the points around the lip of the vase, create a cluster, and scale the cluster down to make the vase look like this.



Tag cluster and scale.

To take advantage of some NURBS surface editing techniques, use **Effect → Convert** to convert the patch to a NURBS. For procedures, see *Converting Surface Objects* on page 166.

Now tag the row of points above the waist and choose **Edit → NURBS Weight**. Drag the mouse to interactively change the weight of the tagged points to give the vase a thicker waist.



Adjust weighting of points around waist.

Procedures for editing the weight of NURBS points are discussed in *Editing NURBS Weight* on page 121.

Inverting the Surface Curves

You can usually see by the appearance of your object in the Shade view whether the normals are pointing in the correct direction for rendering: normals facing the camera are illuminated. When the object's normals do not give the desired appearance, you can use the **Effect → Inverse** command.

If there is a modelling relation between a curve and an object, and you wish to inverse the direction of the generated object's normals, you do not need to inverse the generator curve. Instead, simply select the object itself and choose **Effect → Inverse**.

Effect → Inverse creates a modelling relation between the generated object and itself, which allows you to invert the object independently of the generator curve.

1. Choose the **Show → Normal** command to see the current direction of the normals.
2. Select a surface object.
3. Choose the **Effect → Inverse** command.

The direction of the normals is reversed.

Editing the Surface Resolution

The resolution of a surface is related to its **Step** value, which can be edited in the Info dialogue box.

1. Select the surface object.
2. Choose the **Info → Selection** command.

The Info dialogue box for that object (Patch or NURBS) is displayed.

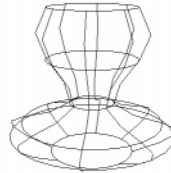
3. To change the surface resolution, enter a new value in **Step** for U and/or V.
4. Click **Ok** to accept the settings and **Cancel** to exit the dialogue box.



You can also use the Info dialogue box to access the Cluster List dialogue box, or view curve information for any projected or trim curves on NURBS surfaces.

Example

In the case of the vase, if you decreased the **Step** value in **V** to 2, it would look like this.



Lower Step value.

Surface Resolution in the Shade View

To specify the “surface step” for all patch and NURBS surfaces displayed in the Shade view, you can set the **Surface Precision** value in the Shade View Setup dialogue box. This setting does not affect the actual step values of the objects in your scene. For example, you can set high step values in the Info dialogue box (for all renderable objects) for high-quality rendered images while setting a low surface precision in the Shade view for a faster display. A surface precision value of 0 uses whatever value is specified in the Info dialogue box.

Capping a Surface

When you select a predefined surface object, such as a patch or NURBS sphere, or when you create one by revolving a curve, the surface can be open in **U** at the top or bottom.

When you choose **Info** → **Selection** on a surface to open the Patch or NURBS Surface Info dialogue box, the **Capping** options allow you to group the points on the first or last row of the surface with a kind of magnet effect.

The points are not merged, but they can now moved and manipulated as one, which can be especially handy for speeding up point editing.

When you uncapped the points, they do not return to their former position, but they can be moved individually.



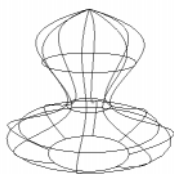
When capping, **Bottom** refers to the start of curve in the specified direction and **Top** refers to the end of the curve in that direction.

If you make a mistake, the **History → Undo** command may reverse the process and return the points to their position before the capping operation, if it isn't too late.

Capping is not the same as closing the surface. In fact, the two operations are mutually exclusive. If the surface is closed in U, the **Capping** options cannot be selected. Conversely, if you cap in one direction, you cannot close it.

Example

If we take the example of the lamp, you can select **Capping - Bottom** in the Info dialogue box. The vase now looks a little like a genie's lamp.



Capping enabled.

To close a surface in U or V, you can use **Effect → Curve Controls** or **Draw → Open/Close**.

Editing Surfaces using Custom Effects

After creating a surface object, use the **Info → ChangeInfo** command in the Model module to create a modified version of a model by editing information you usually access from the **Info → Selection** command of a selected model. This allows you to easily change the resolution of a complex model (such as a human body made of several surfaces) with a single mouse click. You can then use the different versions of your model (low, mid, high res) in conjunction with the **Get → Substitution** command.

For more information on this effect, see its command description in the *Reference Guide*.

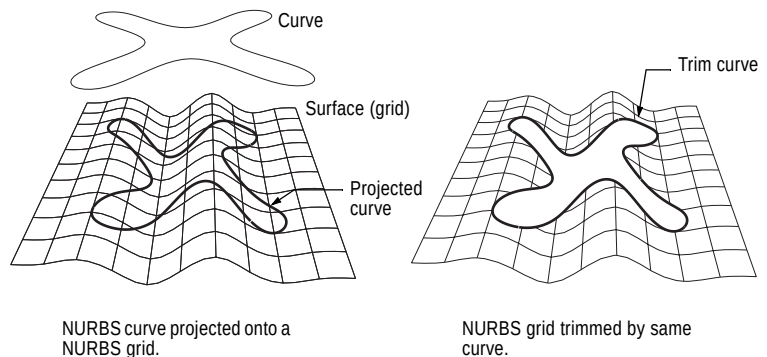
Modelling NURBS Surfaces with Projected Curves

Some important modelling operations are available only with NURBS curves and surfaces. For example, you can project a NURBS curve onto a NURBS surface, with or without “trimming” the underlying surface with the curve.

The two operations, projecting and trimming, are based on the same processes of curve projection and curve fitting, but the objectives and results can be very different.

How does Projecting Compare with Trimming?

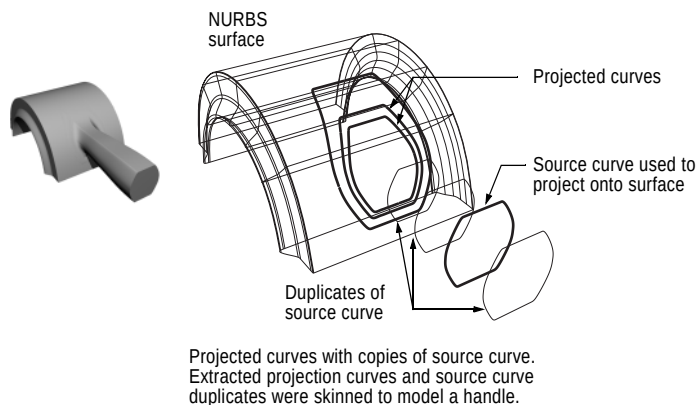
Projecting curves onto NURBS surfaces is essentially the same process as trimming NURBS surfaces. In both cases, you are projecting a curve onto a 3D surface.



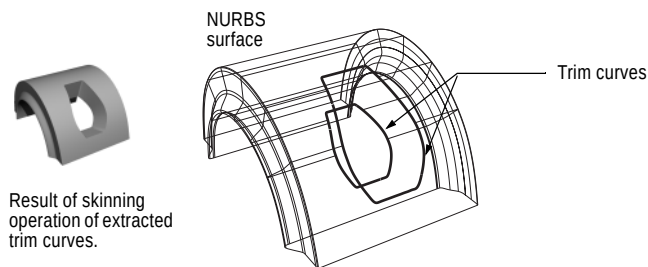
In both cases, you can apply similar modelling operations to the curve that is on the surface. For example, after projecting a curve onto a surface, you can extract the surface curve, extrude it, and modify the generated objects, and use relational modelling if you choose.

Different Results

Projected curves allow you to “fit” a curve on the surface and model with that curve, without affecting the underlying surface at all. The primary objective is to model with the new curve shape.



On the other hand, projected “trim” curves change the surface onto which they are projected. For this reason, the objective is to sculpt and model the surface itself using the projected curve as the “sculpting tool” – for instance, use a trim curve to take a bite out of an apple!



Using different trimming options, you can trim away most the surface, leaving different pieces to model with. The trimming process in SOFTIMAGE 3D is fast enough to see your results right away, which makes it considerably easier to experiment with.

For more information and examples, see *How Trimming Works* on page 194.

Combinations

You can combine projected curve and trim curves on the same surface, for example. Projected curves have no interaction with each other or with trim curves on the target surface. As a result, projected curves can overlap each other or trim curves.

Trim curves can also intersect, giving different results depending on the combination. For examples, see *Trimming Options* on page 195.

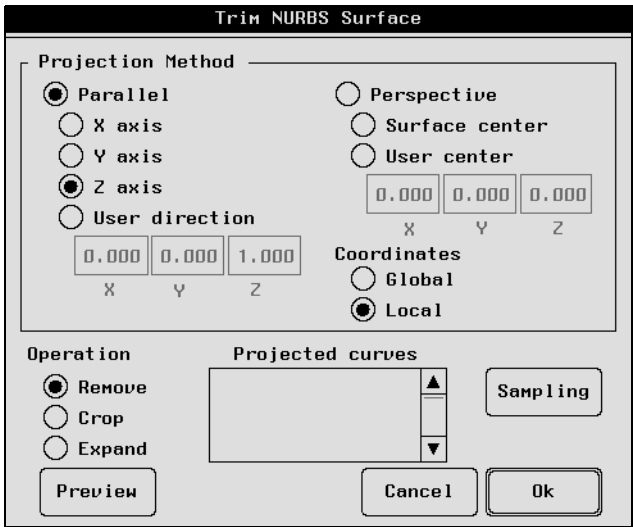
Procedure for Projecting or Trimming

Whether projecting or trimming, both the source curve and the target object must be NURBS. The curve must be closed.

1. Create your NURBS surface and projection curve.
2. Draw a NURBS curve and close it using the **Draw → Open/Close** command.
3. Position the curve in relation to the target surface; for example, translate it, scale it, or rotate it to the desired angle.
4. Select the target surface.
5. Choose either the **Draw → Project on NURBS Surface** or **Draw → Trim NURBS Surface** command and pick the source curve.

Depending on the command you chose, either the Trim NURBS Surface or Project on NURBS Surface dialogue box appears. They are

identical, except for the presence of the **Trimming Operations** in the Trim NURBS Surface dialogue box.



Detailed procedures for setting projection and trim parameters, changing sampling data, previewing and selecting the resulting curves are provided, starting on page 188.

You can also delete projected or trim curves, or extract them. For procedures and examples, see *Deleting Projected or Trim Curves* on page 201 or see *Extracting the Curve* on page 199.

The following procedure covers the basic use of either dialogue box.

Setting the Projection Parameters

Select the projection source or direction, the coordinate system to be used, and the coordinates of a user-defined centre or direction of projection.

Cyan lines indicate the source and direction of the projection ray. As you modify the settings, the direction of the lines changes interactively.

Note In perspective projection, these lines are projected from the point of projection to evenly spaced locations along the trim curve. The number of lines visible may vary in each window.

1. Select either the **Perspective** or **Parallel Projection Method** (see *Projection Methods* on page 191).

2. If you selected **Parallel** projection, choose a predefined direction (**X**, **Y**, or **Z axis**) or a user-defined one.

If you selected perspective projection, choose either the surface centre point or a **User centre** point, which can be inside or outside the surface.

3. Select the **Coordinate** system to be used (**Local** or **Global**).
 - When you select **Parallel** projection, local coordinates are with respect to the projected curve. When you select **Perspective** projection, local coordinates are with respect to the target surface.
 - **Global** selects the world coordinate system.
4. If you chose a user-defined direction or centre point in step 2, use the **Coordinate** text edit boxes to set the desired coordinates.

If you chose Surface centre or one of X, Y or Z axes, the coordinate text boxes are ghosted.

Setting the Trim Parameters

5. If you are in the Trim NURBS Surface dialogue box, select the desired Operation option:
 - **Remove** removes anything within the trim region, creating a hole in the surface.
 - **Expand** fills in the area falling within the trim region, creating a boundary.
 - **Crop** removes all regions outside the trim region, up to the next boundary encountered.

These options are the only ones that occur in the Trim NURBS Surface dialogue box, but not in the Project on NURBS Surface dialogue box. For examples of the results, see *Trimming Options* on page 195.

Setting the Sampling Parameters

The basic formula for sampling projected curves is:

“Number of control points multiplied by **Step** value of the curve”

6. If you wish to change the **Step** value from the default of 10, click **Sampling** (see *The Curve Fitting Process* on page 193).
7. The Curve Info dialogue box appears, in which you can increase or decrease the **Step** value of the curve.

By adjusting the **Step** value, you can get greater fidelity to the shape of the original curve, but it depends mostly on the results you want. Previewing the results is the easiest way to judge.

8. Click **Ok** to exit the Curve Info dialogue box and return to the Trim or Project dialogue box.

Previewing and Selecting Projected Curves

Most often, there are multiple projected or trim curves created on the surface along the projection direction, and you may not wish to accept them all.

You can preview all potential projected or trim curves, check for correct shape and location, or select specific regions for trimming.

9. Click **Preview**.

- In the windows, all the potential trim intersections are highlighted in magenta on the target surface.
- In the Projected Curves scroll box, the potential projected curves and/ or trim curves on the surface are listed by number.

For example, if you are trimming a surface, you may choose to accept trim region number 1, number 4, and number 5.



The number sequence does not necessarily correspond to the apparent sequence on the surface. Just use the highlighted curve on the target surface to make you are selecting the region you want to trim.

10. Click to select a curve in the box (or Shift-click to select more than one) and highlight the desired curves on the target surface.
11. Click **Ok** to accept the settings. Clicking only **Ok** without previewing applies all potential projected or trim curves.

If you are using projected curves, the surface remains unchanged, but the projected curves are now fitted to the surface.

If you are trimming, creates the selected holes or boundaries are created on the surface.

You can now extract the curves and start modelling with them.

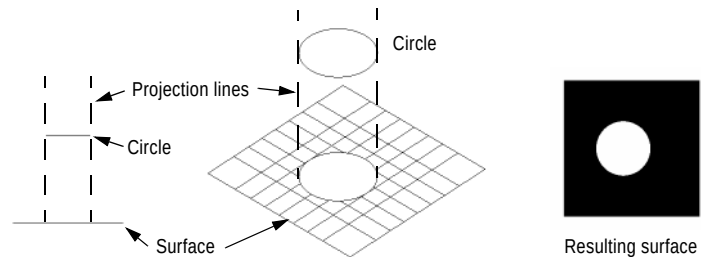
Projection Methods

The projection method and relative position and orientation of the source curve and the NURBS surface determine the size and shape of the curve projected on the NURBS surface. There are two basic methods of projection: perspective and parallel.

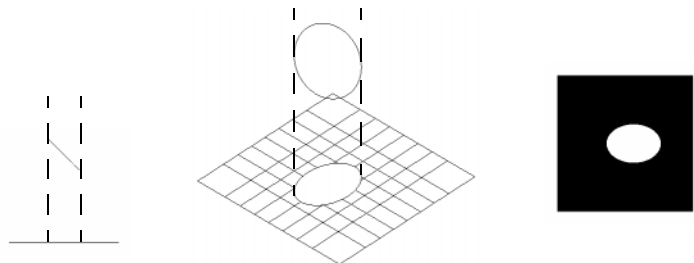
Parallel Projection

With parallel projection, the projection source is infinity and as a result, the rays remain parallel. You need only specify the direction of the ray. The size of the projected curve on the surface depends on the position and shape of the surface.

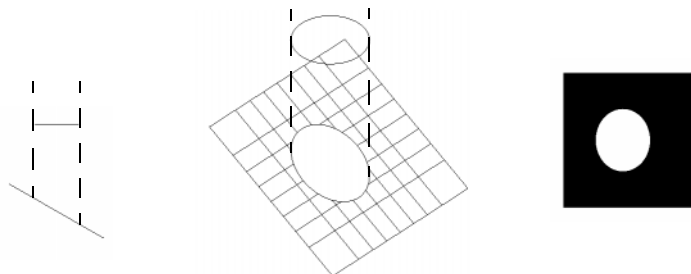
This example shows parallel projection perpendicular to both projected curve (circle) and surface. Result is a circle on the surface.



This shows parallel projection perpendicular to the surface, but not to the projected curve (circle); the projection is using global coordinates and the circle is at an angle. The result is something like a squashed ellipse on the surface.



This shows parallel projection perpendicular to the circle, but not to the surface; the surface is at an angle. The result is something like an elongated ellipse shape on the surface.



Perspective Projection

With perspective projection, the curve is projected onto the surface by a ray originating from either:

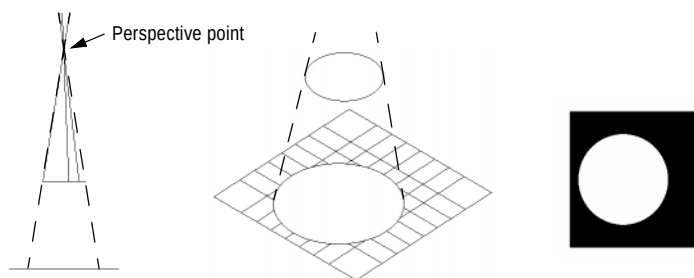
- The surface centre.

or

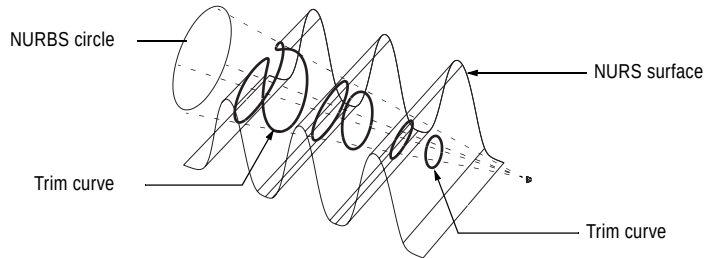
- A user-defined point inside or outside the surface, and using either coordinates local to the surface or global coordinates.

Following the rules of perspective, the curve is projected onto the surface at a different size according to the relative distance and angle of the projection source.

This example shows perspective projection from outside the surface; surface and circle are parallel. The result is a bigger circle on the surface.



If you think of the curve as being extruded through the surface, it likely intersects the surface more than once along the projection direction.



Multiple intersections of a circle trimming a surface.

You can preview and select which projected curves you wish to accept and which you wish to ignore, allowing greater control over the results (see *Previewing and Selecting Projected Curves* on page 190).

The Curve Fitting Process

How does the curve you start with compare to the result on the surface?

Projecting the source curve to the 3D surface requires sufficient sampling data. By default, the sampling formula is approximately the number of control points of the curve multiplied by **Step** value.

For example, a closed curve of 6 control points and the default **Step** value of 10 gives about 60 sampling points. This can be changed prior to projection by clicking the **Sampling** button in the Projection or Trim dialogue box.

By a process which is transparent to you, the source curve is projected into the uv domain of the surface (not the same as the 3D representation in the windows). The result (which you don't see) is a linear degree curve about 60 points.

The Result on the Surface

In turn, this linear curve on the uv domain is fitted with a curve of the same degree as the original, and with the *fewest* number of control points necessary to maintain the curve shape.

On the 3D surface as it appears in the window, you see a curve of the corresponding shape and degree as that in the uv domain. For

example, if you started with a cubic degree curve, you would end up with a cubic degree curve on the surface. You see only this result.

Different Number of Points

As already mentioned, the curve fitting process actually samples many more points than you used in drawing the curve. This may result in an apparent discrepancy when you look at the **Point** value for the surface curve using **Info ▶ Selection**. For example, when projected onto the surface, a curve of 6 points may now be a curve of 16 points.

The curve fitting process tries to use the smallest number of points required to accurately produce a curve of the same degree and shape as the curve in the uv domain. Any apparent discrepancy you notice in the **Info** dialogue box should not be significant.

Degree and Step

The degree and step of the resulting curve on the surface are the same as the original projected curve. For example, a cubic projected curve with a step of 10 results in a cubic curve on the surface with the same step.

However, when trim curves intersect, the result uses the highest curve step and degree. For more information and techniques for preserving curve shape, see *When Trim Regions Intersect* on page 196.

Projected curves have no interaction with each other or with trim curves on the target surface, so their degree is not affected.

How Trimming Works

Trimming can be compared to using a cookie cutter: cut out what you want and use the different pieces that you create.

If you remove the region within the trim curve, the target object remains with a hole cut out of it. Alternatively, you may choose to crop or expand the trim region, in which case the solid “cookie” remains and all or part of the surface is cut away. You can combine the pieces in various ways, examples of which are shown starting on page 196.

On the NURBS surface, the area lying inside this projected curve can define either a hole in the surface or a boundary.

Trimming Options

You can remove the area within the trim curve, crop it or expand it.

Remove removes anything within the trim curve, creating a hole in the surface.

Expand fills in the area falling within the trim region, creating a boundary. Thus, if you projected a trim curve onto an existing hole in the surface, **Expand** would fill in any region enclosed inside it on the surface. By the same token, if you projected it onto a solid region of the surface, **Expand** would have no effect.

Crop removes all regions outside the trim region, but *only* up to the next boundary encountered.

Depending on what you want to do, you can create an “island” floating within a hole within the surface. Holes and islands can be nested within each other to any level.



The surface



Two holes



Islands within holes



Floating islands



Nested holes
(holes within islands)

When Trim Regions Intersect

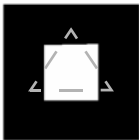
If a new trim curve intersects an existing one, the results might be compared to Boolean union, difference, or intersection operations as in the following examples.

Example 1

The existing trim region is a hole in the surface. If you project a new trim curve and **Remove** it, the result is a bigger hole which includes both the first and the second holes.



Original surface



"Remove" triangle



Resulting surface

Example 2

The existing trim region is a hole in the surface. If you project a new trim curve and **Expand** it, any empty areas within the trim intersection are filled (creating a boundary) and the result looks like this.



Original surface



"Expand" triangle



Resulting surface

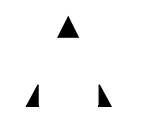
Note that in the same situation, if you **Crop** the second trim curve, this removes all areas outside the intersection area up to the next boundary encountered, which in this case is the edge of the surface. The result would look like this:



Original surface



"Crop" triangle



Resulting surface

Example 3

The existing trim region is a boundary within a hole on the surface. You choose to **Remove** the new trim curve. The result looks like this:



Original surface



"Remove" triangle



Resulting surface

Example 4

The existing trim region is an island within a hole in the surface. You choose to **Crop** the new trim intersection. Again, **Crop** removes the areas outside the intersection area up to the next boundary encountered, which in this case is the edge of the hole.



Original surface



"Crop" triangle



Resulting surface

What Degree is the Resulting Trim?

When trim curves intersect, the result is a single curve which uses the *highest* step and degree of the intersecting curves.

For example, if a linear trim intersects with a cubic trim, the result is cubic, and this resulting curve is likely to be smoother than the linear curve version you projected.



If you want to preserve the “boxier” shape of a linear curve, increase the **Step** value before projecting it. The more sampling information, the more faithful representation of the shape of the curve.

Restrictions and Tips For Trim Curves

If you are trimming the surface, no self-intersections are allowed in either the source curve or the target surface. Furthermore, to open, close, or reparameterize a NURBS surface with a curve on it, the following conditions apply:

- If you are using relational modelling, you can perform these operations on the generator curve or surface, and thus modify the generated objects.
- Otherwise, you must:
 - Delete the trim curves. However, if you have already deleted or moved your initial projected curve, you should extract the trim curves first, if you wish to reproject them.
 - Open, close, or reparameterize the NURBS surface.
 - Trim the NURBS surface again.

Trimming entails a significant amount of computing overhead. If you only want to “draw” a curve on the surface and then model with that curve, and if holes in the surface are not necessary to the final result, use the **Draw → Project on NURBS Surface** command.

If your rendered trimmed surface seems incorrect, even though the wireframe is correct and there were no error messages during trimming process, then the problem may be (yet again) inadequate sampling on the surface or the trim curves.

First, try increase the sampling of the surface:

1. Select the surface and choose **Info → Selection**.
2. Increase the **Step** value in the dialogue box to increase the resolution.

If that does not help, try increasing the sampling on the individual trim curves:

1. Select the surface and choose **Info → Selection**. In the Info dialogue box:
 - Select a curve in the **Trim and projected curves** scroll box.
 - Click the **Trim** button to display an Info box on the selected trim curve.
 - Increase its **Step** value and click Ok to exit.
2. If the **Step** value for a trim curve seems very high, it may be appropriate to reduce it.

If neither of these methods work, remove the trim curves and trim the surface again using the original curve(s), but increase their **Sampling** values.

Modelling with Projected Curves

Having projected these curves to the surface or trimmed your surface with them, now what?

Extracting the Curve

You cannot interact directly with the projected curve on the surface, so the first step to modelling with these curves that now lie on the surface is usually to extract them using the **Draw → Extract** command. This allows you to copy it, scale it, extrude it out the surface, and so on. For procedures, see *Extracting a Curve from a Surface* on page 104.

Extracting projected curves from the surface involves another curve fitting process, so extracted curves have a different number of points than the original projected curve.

This apparent discrepancy is not significant. The extracted curve has the fewest possible points possible to maintain the shape of the curve on the surface.

Examples of Modelling with Projected Curves

You may simply wish to take the new curve shape that results from fitting the curve to a 3D surface, and use it for modelling.

Alternately, you may wish model the surface itself using the curve; for example, extract the curve, and extrude it out to model a spout on a jug.

You could also have a head and ear which you can join in a seamless fashion. This makes use of a variety of modelling operations, including relational modelling.

1. Before you begin, choose the **Preferences → Create Modelling Relation** command.
2. Extract the curve on the inside of the ear, using the **Draw → Extract** command. There is a modelling relation between the curve on the ear and the extracted curve.
3. Before you project the extracted curve onto the head, deactivate the **Preferences → Create Modelling Relation** command.

If you don't do this, you won't be able to "untwist" the skin later if you need to because the curve on the head will rotate as well.

4. Project the extracted curve onto the head using **Draw → Project on NURBS Surface**. There is now a matching curve on the ear and the head.
 5. Reactivate relational modelling with the **Preferences → Create Modelling Relation** command.
 6. To be able to work with the curve that is now projected on the head, you need to extract it using **Draw → Extract**.
 7. Duplicate the extracted curve once.
 8. Translate the duplicate curve to an appropriate position between the head and ear.
 9. Freeze the modelling relation on the duplicate curve using **Effect → Freeze → Modelling Relation**, and translate the curve's centre to the actual centre of the curve before scaling it.
 10. Scale the translated curve.
 11. Duplicate and scale again as required to create a sufficient number of curves for the curve profile.
 12. Cover the curve profile using the **Surface → Skin** command.
-  If the resulting skin is twisted, use **Effect → Curve Controls** on rotate the starting point of the curve on the ear as required to “untwist” the skin.
13. Make sure to add the middle curves to the hierarchy, so that they move with the head.

Using Relational Modelling

By activating **Preference → Create Modelling Relation** before the projection operation, you can apply relational modelling on the resulting object and curves.

For example, get a primitive sphere, draw a curve and use it to trim the sphere. The original (untrimmed) sphere is preserved, as well as the newly trimmed sphere.

From now on, if you edit either of the generator objects (the original curve or the original sphere), the changes are reflected in the generated objects (the trimmed sphere or the trimmed region on its surface).

To clarify what you're working on, translate the generated sphere so you can see both spheres and the original curve.

Note

Don't translate the original sphere! Because of the modelling relation, this changes the relative position of generator and generated objects, so your generated object would no longer have its projected curves.

You can:

- Transform, edit, open, close, or reparameterize the original curve. See the changes reflected in the corresponding projected curve on the generated sphere.
- Edit, open, close, or reparameterize the original sphere and see how the changes modifies the generated sphere accordingly.

For example, you could select a trim curve on the surface, extract that curve, and then extrude it. By editing the original curve used for trimming, you can see the ripple effects on the generated object (the curve extruded from the surface).

To cut this modelling relation at any point, choose **Effect → Freeze → Modelling Relation**.

Tips

Relational modelling on trimmed NURBS can slow down modelling operations considerably, particularly when using complex surfaces or trim curves with many control points.

To improve the performance somewhat, deselect the **Preferences → Interactive Update** command or press the **n** Supra key while translating the generator curve or surface

You cannot use two Supra keys at once (such as **m** and **n**). To translate a point, use **Edit → Move Point** instead of the **m** Supra key, so you can use **n** Supra key while translating the desired point.

Deleting Projected or Trim Curves

You can delete selected curves from the target surface. When you delete projected curves (as opposed to trim curves), it does not affect the surface in any way. If you delete all trim curves, the surface returns to its state before you trimmed it.

1. Select a NURBS surface which has projected or trim curves.

2. To delete projected curves, choose **Draw → Delete Projected Curves**. To delete trim curves, choose **Draw → Delete Trim Curve**.
3. On the surface, pick a curve to delete. Your selection is highlighted in magenta and the Delete projection curve or Delete trim curves dialogue box appears, depending on your chosen operation. The two dialogue boxes are identical.

Your selection is highlighted in the scroll box, but you can select another if you wish.



Make sure you are selecting the correct curve by referring to the highlighted (magenta) curve on the target surface.

4. Click **Delete**.

The dialogue box remains displayed and you can continue to undo and redo deletions, select other curves, and delete them.

5. Click **Ok** to accept the current settings and exit the dialogue box or click **Cancel** to exit without accepting the current settings.

Error Messages and Possible Solutions

If you get either unexpected results or no results at all, the problem is usually in an incorrect projection direction or insufficient sampling value used for the projected curve. Here are some possible error messages and what to do.

Ignoring projected curve with point count less than curve order.
Ignoring projected curve containing less than 3 samples.

The projected curve does not contain enough points to form a valid curve. Make sure the curve projection direction was correct or increase the sampling.

Unable to fit high degree curve to trim curve. Using linear projection curve.

The curve fitting process failed, possibly due to large gaps in samples of the projected curve. Try increasing the sampling when projecting the curve.

Ignoring projected curve with no area.

The bounding box of the projected curve has no area. Check whether the curve is projected in the proper direction.

Ignoring projected curve containing self intersections.

Either the original curve or the projected curve on the surface intersects itself (such as a figure 8 curve). Make sure the source curve contains no self intersection. If not, check for valid projection direction to make sure projected curve does not contain self intersection.

Loop detected in building composite curve.

Loop detected in projected samples.

These two messages mean that it was impossible to piece together a valid trim curve from the samples. Check the projection direction, in particular make sure part of the surface is not parallel to projection direction.

This problem may also be caused by inadequate sampling, in which case project the curve again, increasing the sampling.

Detected portion of surface parallel to projection direction.

Error projecting Ray.

These messages mean that all or part of surface is parallel to the projection direction. If the surface is flat, it is possible for one of the parallel projection rays to miss part of the surface entirely. Check the projection direction first.

If this is correct, modify the surface slightly (for example, make it curve it slightly or pull a few points up) to add projection information. After a successful projection, return the surface to its original shape.

Unable to determine proper curve orientation.

The projection was unable to determine the orientation of projected curve (hole or boundary). This is usually caused by inadequate sampling. Increase the sampling and project again.

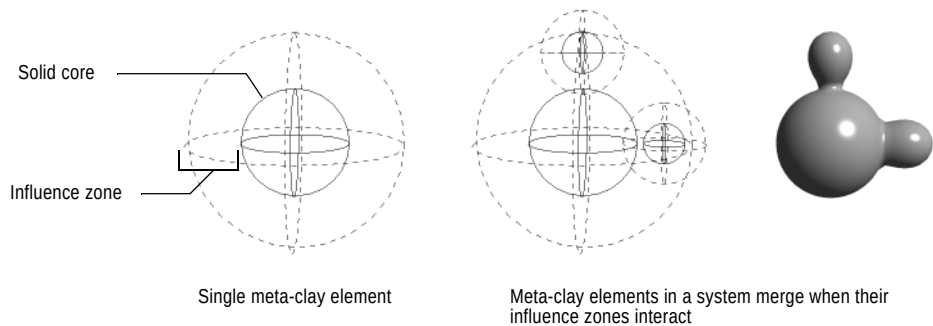
C H A P T E R S I X

Meta-clay Objects

Some Basics of Meta-Clay

Meta-clay objects are used to create models that require smooth and varying curves. They are based on density rather than on geometry. They are ideally suited to modelling smooth, soft-edged, organic objects such as the human (or alien!) bodies.

A *meta-clay element* is a field of matter consisting of a solid area, called a *core*, surrounded by a semi-solid area of influence called the *influence zone*. The solid core represents the actual physical object for a single meta-clay element; its outer boundary is shown with a solid line.



The influence zone is a zone of weighted density surrounding the core. Density decreases with distance from the centre until it reaches its defined edge, where it can be considered to have zero density. Since the influence zone surrounding the core does not contain enough density to create a solid object, its outer boundary is shown as a dotted line.

A meta-clay element is by default a sphere, but can also be defined as an ellipsoid, diamond, octahedron, rounded cube, or any shape in between. This is called the meta-clay element's *superquadric exponent*.

As you move meta-clay elements in proximity to each other, they start to blend so that individual meta-clay elements blob together. A *meta-clay system* contains any number of these meta-clay elements. It is a simple hierarchy of the system (parent) and its elements (children).

By definition, there is no blending between different meta-clay systems, only between meta-clay elements within the same system.



For example, a hand would be made from one meta-clay system made up many meta-clay elements. Otherwise, the parts of the hand would not blend, such as the fingers and the palm.

You may wonder “How do I stop elements from blending if I don’t like the result?” For example, how do you avoid the “webbed” effect between adjacent fingers on a hand, or prevent the hand from becoming a big blob when the fingers curl into a fist? Of course, you can use unusual blending techniques to your advantage to create strange new life forms!

To avoid these problems, you can control the degree of blending, create groups of meta-clay elements within the same system, and define which groups are allowed to blend with which.

For information on this and other techniques for controlling the degree of blending, see *Controlling How Meta-clay Elements Blend* on page 215.

Creating Meta-clay Elements and Systems

You can either start by creating elements and build the system, or create the (empty) system and add the elements into it.

Starting with individual elements from which to build a system:

1. Choose the **Meta-Clay → Add Element** command in the Model module.

As you create the first meta-clay element, a meta-clay system is automatically created as the parent of the meta-clay element.

2. To add more meta-clay elements to the system, keep choosing the **Meta-Clay → Add Element** command.

Tip Middle-click on the **Meta-Clay** menu cell to repeat the last selection.

3. You can also use the **Duplicate → Immediate** or **Duplicate → Repetition** commands to copy an existing meta-clay element.

Repeat this step until you have the desired number of meta-clay elements. Of course, you can always change the contents of a meta-clay system.

Alternately, you can create a system first, and add elements to it:

1. If you choose the **Meta-Clay → Add System** command, it creates a meta-clay system labelled “Meta *n*” in the Schematic window.

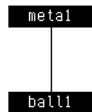
Note The meta-clay system is represented only in the Schematic window; it has no graphical representation in any other window.

2. Choose the **Meta-Clay → Add Element** command.

In the window, a meta-clay element appears. The default shape is a ball, but this can be changed. In the Schematic window, the meta-clay system is now the parent of the meta-clay element.

3. Repeat this step to create the desired number of elements in the system. Each meta-clay element is added on top of the previous ones. Translate them away from each other to start modelling and building your meta-clay model.

You can also incorporate other objects into a meta-clay system, such as a torus, whose shape might not be easy to model with meta-clay. However, these objects will not blend with meta-clay elements.



Importing a Metaball File

Meta Corporation provides a library of predefined metaball objects. You can import a Metaball file using the **Meta-Clay → Read Metaball File** command, described on page 996 of the *Reference Guide*.

Tips for Modelling with Meta-Clay

Meta-clay models can be slow to manipulate and render. To increase the speed of these operations, you can edit the Wireframe Slice Size and the Rendering Slice Size in the Meta-Clay Setup dialogue box. See *Setting the Resolution* on page 213 for more information.

Viewing Meta-clay Elements

Displaying and Hiding Surfaces

You can display or hide a wireframe representation of the blended surfaces of selected meta-clay elements or meta-clay systems in the Wireframe view. This representation approximates the surfaces you would see in the Shade view.

1. Select any element in a meta-clay system.
2. To show the surfaces in Wireframe view, choose the **Meta-Clay → Turn ON Surfaces** command.
3. To hide the surfaces, choose the **Meta-Clay → Turn OFF Surfaces** command. In this case, all elements in the selected system are displayed in their default form (spherical core and influence zone).

Note

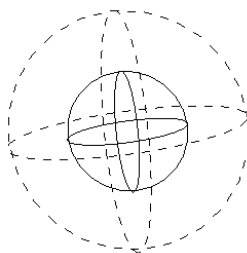
You can get the same effect by selecting or deselecting the **Surface Display** option in the Meta-Clay System Settings dialogue box, accessed by choosing the **Meta-Clay → Setup** command.

Displaying and Hiding Influence Zones

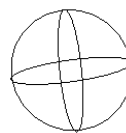
You can show or hide the influence zone of any selected meta-clay element or system. The influence zone surrounds the core of the meta-clay element and its boundary is shown as a dashed line.

By default, the influence zone is displayed.

1. Select a meta-clay element or meta-clay system.
2. To toggle the display status, choose the **Meta-Clay → Show Influence** command. The dashed line representing the influence zone is hidden.
3. Choose the command again to display the influence zone.



Show Influence on



Show Influence off

Editing Meta-clay Systems

Editing an Element's Shape and Influence Zone

In the Meta Element Info dialogue box, you can change the shape of a selected meta-clay element, which is by default a sphere shape.

You can also edit the size and density of the area of influence, and thus change the way elements “attract” and blend with others. Changing the influence zone does not change the size of the core radius.

The changes are limited to the selected meta-clay element, and are not global to the meta-clay system. For information on changing global settings for a meta-clay system, see the **Meta-Clay → Setup** command on page 976 of the *Reference Guide*.

1. Select a meta-clay element.
2. Choose the **Info → Selection** command.

The Meta Element Info dialogue box appears.

3. To edit the shape, use the **Superquadric Exponent** sliders.

These allow you to create shapes that vary between octahedron, ellipsoid, and a rounded square or rectangle. Values for each range from 1 to 8, and the default value is 2.

Try out different combinations of each value to attain a variety of shapes. To see the changes interactively, without exiting the dialogue box, select the **Interactive Refresh** option.

4. To edit the area of influence, change any of the sliders under **Blending Influence**. All three values (**Weight**, **Core Ration**, and **Influence Zone**) are correlated, so when you change one value, the other two also change correspondingly. If **Interactive Refresh** is selected, you can view your changes interactively.
5. If you assign negative weight values to a meta-clay element, the meta-clay element does not have any solid material, but will subtract it from other elements. It can then be used to create holes and depressions in meta-clay models.

Setting the Resolution

You can determine the final resolution of the selected meta-clay system for purposes of viewing and moving in the Wireframe view mode, rendering, or converting to polygon mesh objects.

1. Select the meta-clay hierarchy system (tree).
2. Choose the **Meta-Clay ➔ Setup** command. The Meta-Clay System Settings dialogue box appears.
3. To modify the resolution of the meta-clay system in the Wireframe view mode, enter a value for the **Wireframe Slice Size**. This is similar to setting the Step value for objects which have geometry. The default value is 0.5 SOFTIMAGE units. The smaller the size of the slices, the more refined the wireframe surface will be.
4. To modify the resolution of the meta-clay object in the Shade view, as well as in the SOFTIMAGE and mental ray renderers, enter a new value for the **Rendering Slice Size**.
5. Click Ok to accept the settings or Cancel to exit the dialogue box.

Tips

For faster manipulation in a scene, keep the **Wireframe** and **Rendering Slice Size** values near the default. If necessary, you can make the **Slice** values smaller (to increase the resolution) just before previewing and rendering.

You should test that the model can be rendered on your machine, since it may not have sufficient memory to render meta-clay elements. Try rendering at least one frame.

Making the **Rendering Slice Size** too small can consume a large amount of memory, especially with complex meta-clay models. For more information on this parameter, see the **Meta-Clay ➔ Setup** command in the *Reference Guide*.

The **Wireframe Slice Size** also affects the conversion of meta-clay systems to polygon mesh objects. Setting the **Wireframe Slice Size** smaller than the default can create very heavy polygon mesh objects during conversion.

Applying Materials and Textures

You can only apply a material and a texture to the entire meta-clay system, but not to individual meta-clay elements. The easiest way to do this is in the Schematic window.

1. Select the meta-clay system in the Schematic window.
2. In the Matter mode, follow the procedures described for applying 2D or 3D materials and textures as you would for any object.

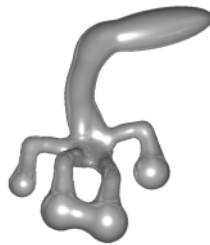
See the *Defining Materials and Textures User's Guide* for more information on doing this.

Controlling How Meta-clay Elements Blend

There is no blending between meta-clay elements in different meta-clay systems: only between elements in the same system.

To prevent unwanted blending of elements in the same system, you can use the **Meta-Clay → Grouping** command to group meta-clay elements into named groups and specify which groups should not blend with which, regardless of proximity. This lets you create complex objects such as a hand using a single meta-clay system.

Let's take this example of a “hand” made from a single meta-clay system. By grouping the meta-clay elements and limiting the blending between, you manipulate the finger positions individually, so as to close the hand or touch the thumb to the index finger without the fingers blending.



System blending

Fingers merge when they are close together because they belong to the same system.



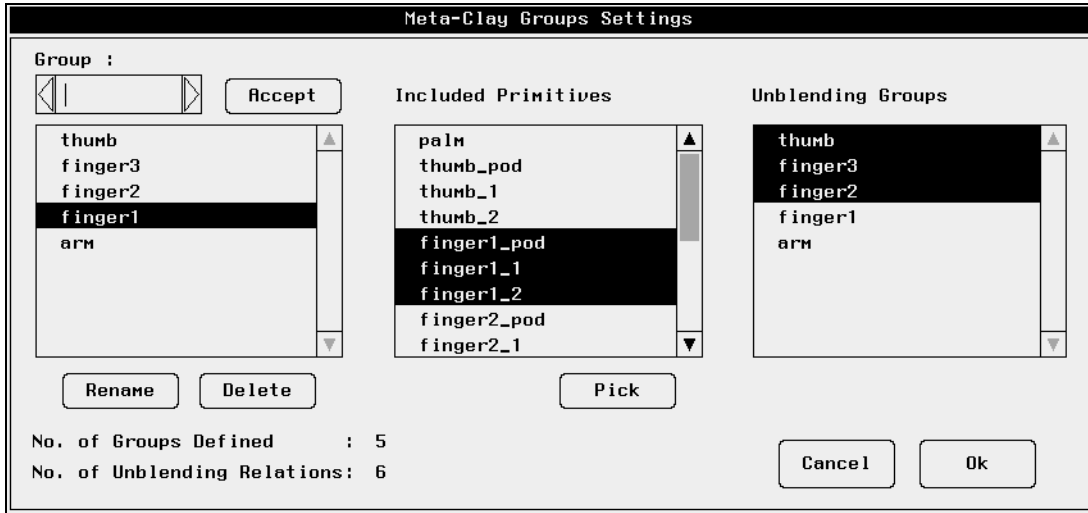
Unblended groups

You can define each finger as a group and then isolate them from each other with unblend.

Defining Groups

1. Select a meta-clay system or a meta-clay element within a system.
2. Choose the **Meta-Clay → Grouping** command. The Meta-Clay Groups Settings dialogue box appears.

In this case, the dialogue box already contains group definitions for the hand that was created on the previous page.



3. Enter a name such as “arm” in the **Group** text box and click **Accept**.

The name “arm” is added to the **Group** scroll box and also the **Unblending Groups** scroll box, and the **No. of Groups Defined** is increased by one.

You can define all the group names before defining their contents.

Defining the Group Contents

4. Select a name in the **Group** scroll box.
5. In the **Included Primitives** scroll box, select the meta-clay elements you wish to include in the current group.

If you have not given the meta-clay elements meaningful names and need to identify the correct element, click the **Pick** button. The dialogue box disappears and you can pick the meta-clay

elements from the windows. When you pick one, the dialogue box reappears.

6. Repeat the previous steps.

A meta-clay element cannot belong to more than one group. For example, say you select the meta-clay elements 1, 2, 3, and 4 and define them as group_a, and subsequently assign meta-clay element 4 to group_b. When you reselect group_a in the **Group** scroll box, you see that it includes only meta-clay elements 1, 2, and 3. Meta-clay Element 4 was automatically subtracted from group_a as soon as you selected it for group_b.

Defining Unblending Relations

7. Select a group name in the **Group** scroll box. The meta-clay elements included in it are highlighted.
8. In **Unblending Groups**, select any group(s) with which you do not wish the current group to blend. For example, for the group “arm,” you could choose any groups except “arm” in **Unblending Groups**.

Each time you select an **Unblending Group**, notice that the **No. of Unblending Relations** increases by one.

9. Select another name in the **Group** scroll box and repeat the steps until all unblending relationships are defined.
10. Click Ok to accept the settings or Cancel to exit the dialogue box.

Animating Meta-Clay System Shapes

To modify a meta-clay system and animate these changes, you can use the typical keyframing approach or create a metamorphosis from one system shape to another at different keyframes and save the changes, similar to shape animation.

Keyframing

You can modify a meta-clay system's transformations as well as modifications to shape and influence zone, and keyframe these modifications.

Meta-clay objects generate function curves for scaling, rotation, translation, weight, and all superquadric exponents.

Keyframing Transformations Only

To keyframe only transformations, you can simply use the **SaveKey** → **Object** → **Scaling, Rotation, Explicit Translation**, or **Transformation** commands used with all object types.

Keyframing Meta-Clay Parameters Only

If you wish to animate only the meta-clay superquadric exponents and core influence (make the system change shape without moving or transforming):

1. Choose the **Info** → **Selection** command. The Meta Element Info dialogue box appears.
2. Click the **Key** button to create a keyframe for the selected element.

This button is interactive with the time line, so you can move the time line and keep on creating keyframes without closing the dialogue box.

Keyframing All Information

If you wish to keyframe all information for the meta-clay element or system – the core influence, superquadric exponents, and transformations – do this:

1. Select the meta-clay element or meta-clay system.
2. Choose the **Meta-Clay** → **Save 'System Shape'** command.

See the next section to find out how to create a metamorphosis with this information.

Creating a Metamorphosis

Say you have several meta-clay systems and you want one to turn into another over time. Starting with one or more *source* meta-clay systems, you can construct a *target* meta-clay system which you want to animate.

At selected keyframes, you save the shape of a selected source system to the target system (the one you are constructing and animating), creating a metamorphosis from one system shape to another at different keyframes.

This technique involves two commands:

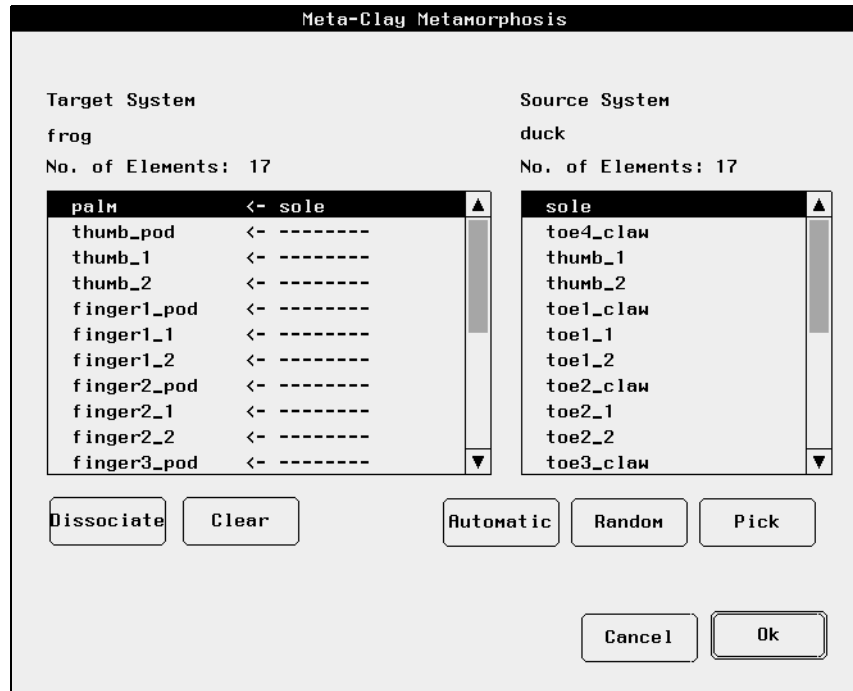
- **Meta-Clay → Pick ‘System Shape’**, which copies the relative position and scaling of individual elements in the source meta-clay system to individual elements in the target meta-clay system, but it does *not* copy the transformations of the entire source system.
- **Meta-Clay → Save ‘System Shape’**, which saves a keyframe for the target system, including superquadric exponents and weight information, as well as transformations.

You can still save keyframes for transformations of the target system.

Procedure

1. Save all source system shapes using the **Meta-Clay → ‘Save System Shape’** command.
2. Select the target system shape (the one you wish to modify).
3. Choose the **Meta-Clay → Pick ‘System Shape’** command.
4. Pick a source system shape.
 - If you left-click to pick the source system shape, this temporarily copies the shape of the source system to the target system.

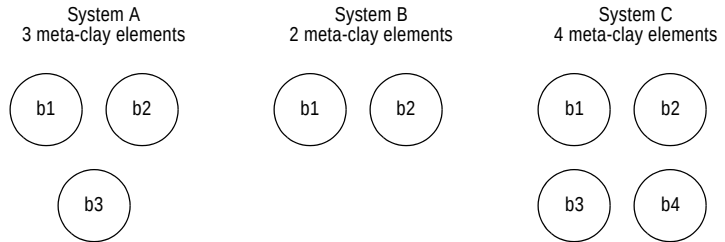
- If you middle-click on the source system, the Metamorphosis dialogue box appears.



5. You can associate the elements you want from the **Source System** to the appropriate element in the **Target System**:
 - Left-click on an element in the **Target System** scroll box to select it.
 - Double-click on an element in the **Source System** scroll box to associate it to the selected target element.
6. If you make a mistake, click **Dissociate** to remove one link or **Clear** to remove all links.
7. To create links automatically, based on relative proximity of source and target elements, click **Automatic**. To create links randomly, click **Random**.
8. Choose the **Meta-Clay** → **Save 'System Shape'** command to save your choices from the previous step.
9. Move to another frame and repeat these steps so that your system blends from one shape to another over time.
10. Play your animation to view the results.

Example of a Metamorphosis

The following are three system “shapes”.



System A will be animated. At different frames, it will assume the shapes of B and C. Here's how.

1. Frame 1: Choose **Meta-Clay** → **Save System Shape**. This saves scaling, rotation, and a path translation of all the meta-clay elements in System A.
2. Frame 10: Choose **Meta-Clay** → **Pick System Shape** and pick System B. This temporarily positions all of the System A's meta-clay elements in the positions of System B. Since there is one less meta-clay element in B than in A, one of the elements in A is scaled to zero.
3. Choose **Meta-Clay** → **Save System Shape**. At this frame, A is an exact copy of B.
4. Frame 20: Choose **Meta-Clay** → **Pick System Shape** and pick system C. This temporarily positions all of System A's meta-clay elements in the positions of System C. Since there is one more element in C compared to A, an element is added to System A. This newly added meta-clay element is invisible to the renderer until that point in the animation; the software animates its core ratio down to zero, with the result of making it invisible.
5. Choose **Meta-Clay** → **Save System Shape**. At this frame, A is an exact copy of C.

Using Meta-clay Elements as Envelopes

In the Actor module, you can create skeleton chains, then give them a “skin” which is called an *envelope*. One of the types of envelopes you can create is a rigid envelope, using the **Skin → Rigid Envelope** command in the Actor module.

Rigid envelopes take advantage of the characteristic blending of meta-clay objects. Furthermore, you can weight them to more than one joint. While they can be made from any kind of object, rigid envelopes are especially useful when using meta-clay objects because the motion is more realistic.

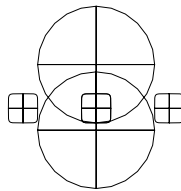
Using a meta-clay system as an envelope for a skeleton chain is an easy way of animating meta-clay elements.

For more information, see *Assigning a Rigid Envelope* on page 253, and also the **Skin → Rigid Envelope** command on page 1322 of the *Reference Guide*.

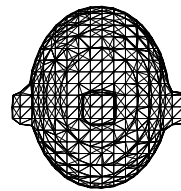
Converting a Meta-Clay System

You can convert a meta-clay system to a polygon mesh object using the **Meta-Clay → Convert to Polygon** command. This allows you to take advantage of the smooth, organic modelling of meta-clay objects, but reduce some of the performance penalty associated with them.

This can also be useful for exporting models built using meta-clay to platforms which do not support meta-clay. You can optimize the resulting polygon mesh object using the Polygon Reduction effects, described in *Optimizing Models* on page 317.



Meta-clay system



Converted to polygon mesh

The relative complexity of the resulting polygon mesh object is affected by the **Wireframe Slice Size** value in the Meta-Clay System Settings dialogue box, accessed by choosing the **Meta-Clay → Setup** command. Setting the Wireframe Slice value smaller than the default can create very heavy polygon mesh objects during conversion.

1. Select a meta-clay system or a meta-clay element.



Check that the **Wireframe Slice Size** in Meta-Clay System Settings dialogue box is at the default value (0.5) or higher. Small values generate excessively high polygon counts!

2. Choose the **Meta-Clay → Convert to Polygon** command.

The active system is converted to a polygon mesh object.

3. To optimize the resulting polygon mesh object, use the Polygon Reduction effects as described in *Optimizing Models* on page 317.

C H A P T E R S E V E N

Skeletons

Introduction

Anatomy of a Skeleton

The human skeleton is made up of torso, limbs and articulation points, each of which can move to some extent independently of each other, and also in unison. Similarly, SOFTIMAGE 3D skeletons can include any number of articulated chains, which is a hierarchy made up of a chain root, joints, and an end effector.

These skeletons can also be composed entirely of articulated chains or of 3D objects, or a mixture of both objects and articulated chains.

Purposes

As with the human skeleton and skin, in SOFTIMAGE 3D you cover a skeleton with “skin,” and you want to use a skeleton to move it or change the skin’s shape in some way.

Skeletons serve both these purposes:

- Animation: articulated chains are the key to animation with inverse kinematics.
- Deformation: when grouped as skeletons, both chains and objects help to deform the surrounding skin. Objects in skeletons can give better results in local deformations of the skin (facial expressions, bulging muscles, and so on) than do articulated chains.

The section following discusses the building blocks of skeletons. To find out how to put them together, see *Defining a Skeleton* on page 235.

How Do You Visualize a Skeleton?

Skeletons are visible in the SOFTIMAGE 3D wireframe display, but invisible to the renderer, like other tools that are used to manipulate objects, such as curves and nulls.

To make it visible, you must add “skin,” which is called an *envelope* in SOFTIMAGE 3D.

When you move the skeleton, its envelope moves and stretches with it. You have a certain amount of control over the way the envelope is deformed by the skeleton’s movement. Together, the skeleton and its envelope(s) are known as an *actor*.

For more information, see *Envelopes* on page 243.

What Makes Those Bones Move?

In SOFTIMAGE 3D, you can use *inverse kinematics*, a simple animation method in which you only need to drag the end of a chain to a desired point.

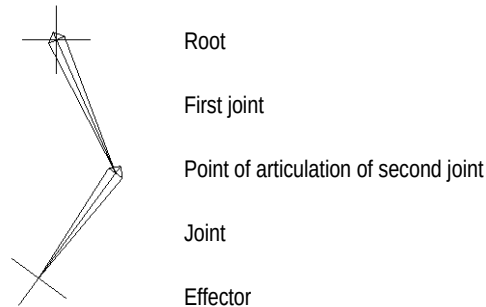
Inverse kinematics registers the new position of the extremity and automatically calculates the new positions of the intermediate chain joints based on that. As such, it is “bottom up” animation of the skeleton.

How parts of the skeleton move relative to each other is determined by the way your skeleton hierarchy is built, whether and how objects are constrained to each other, and if you have created any expressions to govern their motion.

This chapter is concerned with process of modelling skeletons. For more information on using skeletons for inverse kinematics, as well as techniques such as forward kinematics, constraints, and expressions, see the related chapters in the *Animating User's Guide*.

A Closer Look at Articulated Chains

A chain is made up a chain root, one or more joints, and an effector.



There is no limit to the number of joints that can be included in a chain. Chains can be 3D or 2D.

In a 3D chain, all joints behave like ball joints. They can be rotated about all three axes and can move in any plane.

In 2D chains, only the chain root and first joint rotate around three axes, like a ball joint. All other joints are constrained to stay on a single plane which is defined by the x and y-axes of the first joint of the chain. These joints in the chain can rotate about the local z-axis *only*, like a hinge joint rather than a ball joint.

You could use a 3D chain to create swaying seaweed which makes use of the random movement. However, to model a leg, a 2D chain would be more useful because a leg moves more like a hinge than a ball joint.

Chain Root

The chain root is the first point of a chain. Its position determines the orientation of the chain, and it often contains data associated with the whole chain. It is displayed as a null model that can be selected.

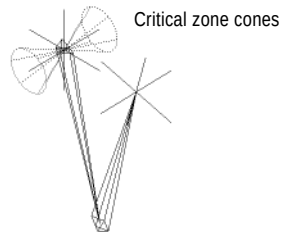
In skeletons with multiple chains, a chain root can be the child of another chain or object.

Critical Zone

There is a zone around the chain root in which the general orientation of a chain will flip if the effector is translated into the zone. It is called the *critical zone* and it is represented by a double-

cone shaped icon located at the point of articulation of the first joint of a chain. The chain will not flip in the plane perpendicular to the preferred axis of rotation.

To visualize it, use the **Skeleton → Show Critical Zone** command in the Actor module.



Joint

A joint is a link in the articulated chain. It is composed of a point of articulation and a stretched octahedron. Joints are connected end to end with the local x-axis pointing towards the next joint.

Joints can be scaled (to change the length of an actor's leg, for example), and the first joint has three translational degrees of freedom (x, y, and z) since the chain root can be moved anywhere in space. The default node name for a joint is `jnt1_1` or `jntn_n` as you can see in the Schematic window or by using the **Info → Selection** command.

Effector

The effector is the last point of a chain. Inverse kinematics is invoked when you move the effector. In the display windows, the effector is displayed as a null model that can be selected. In skeletons with multiple chains, the effector of one chain can be defined as the parent another chain.

Drawing Articulated Chains

Articulated chains are created by choosing the **Skeleton → Draw 2D Chain** or **Draw 3D Chain** command and entering a series of points. The first point entered defines the chain root, and the last point entered defines the effector.

1. Choose either the **Skeleton → Draw 2D Chain** or **Draw 3D Chain** command in the Actor module.

2. To create the first joint, click in any window.

A null object is created with its local z-axis pointing “out” from the window.

3. Move the mouse cursor to a new position and click again.

The null is now defined as the chain root, and the first joint is created.

4. Follow the instructions in the status line:

- To add joints, continue clicking to create as many joints as desired.



Each newly created joint can be rotated to a new position before the next joint is created, if you keep the mouse button held down and drag the joint to the desired position.

- To start a new chain, middle-click. Move your cursor and begin another chain. If you use the same window for drawing all chains, draw all joints in the window's plane.

5. To end the drawing mode, right-click.

Tips for Drawing Chains

When a 2D chain is drawn in a given window (such as the Front window), it is created in the local xy plane of the first joint defined and rotates about the local z-axis. The local x-axis always points to the point of articulation of the next joint.

A 3D articulated chain is also created in the local xy plane of the first joint, but it rotates in all three axes and can be moved in any plane.

You can manipulate any chain successfully in 3D by using constraints. Constraints allow you to constrain objects (models, nulls, chains, etc.) to another object. This makes it easy for you to animate an object using the movement of the constraining object.

For more information, see *Constraints* on page 167 of the *Animating User's Guide*.

Preferred Angles

When you draw a chain, this establishes the “preferred” angles of all joints and determines how the chain will bend when you move the effector.

If you create a straight chain aligned with the grid, you may not be able to predict how the chain will bend.

You can subsequently edit the joints by moving or rotating them; see *Editing Joints* on page 233. You can also refer to see *Controlling the Orientation of a Chain* on page 137 of the *Animating User's Guide*.

Direction of Drawing

Remember that the first point entered defines the chain root, and the last point entered defines the effector. For example, to draw a human skeleton, you would draw the spine upwards from base to neck so that the chain root is in the pelvis.



You wouldn't drag the “neck” effector to move a spine. Inverse kinematics do not work well on a spine because the spine could not bend backward or sideways, or twist. Instead, you would use forward kinematics (changing the rotation of each joint) or channel assignment.

When drawing legs, arms, or anything else on which you can use inverse kinematics, draw the skeleton in the way it will naturally bend. For example, if you draw a leg chain in the Right window, and you draw the chain root and effector in alignment, the knee joint should be bent forward a bit. This establishes the preferred angle for proper inverse kinematics.

Toggling between 2D and 3D Chains

You can change a selected chain from 2D to 3D, or vice versa, using the **Skeleton → Toggle 2D/3D** command.

1. Select a chain.
2. Choose the **Skeleton → Toggle 2D/3D** command in the Actor module.

The chain type is changed. A confirmation message is displayed in the status line.



Choose the **Info → Selection** command to verify the current type of chain with which you are working.

Editing Joints

After creating a chain, you may need to edit its look and shape. For example, the “knee” of your skeleton may be poking outside the envelope, or the “thigh bone” may be shorter than the “shin bone.”

You can translate joints or rotate joints on a chain to adjust the way the skeleton joints fit within the envelope, or the scale a joint and change its length respective to other joints.

You may also wish to change the preferred position and orientation of the joints, or scale them.

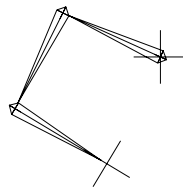
Repositioning and Stretching Joints

What if you want to reposition a specific joint or the last n joints in a chain? There are two commands in the Actor module that move selected joint(s) along one axis (the local x-axis of its parent) or in free translation.

Choosing the **Skeleton → Move Joint** command translates only the specific joint.

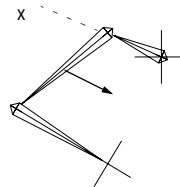
If you choose the **Skeleton → Move Branch** command, the selected joint and all joints beneath it are moved as a group.

You will see how translating a joint stretches out its neighbours.



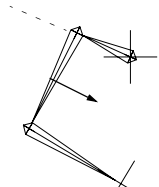
Move joint

Example shows middle joint moved along its parent's x-axis.



Move branch

Example shows middle joint (and below) moved along its parent's x-axis.



Note

Neither operation moves the chain root or the first joint.

You can move joints at any of these times:

- Before or after defining the chain as a skeleton. If you place the joints before defining the skeleton, this placement is part of the preferred position and orientation and this is considered.
- Before or after animating the chain. In the last case, this may affect the animation results which may need to be adjusted.

To move joints or branches:

1. Select the entire chain.
2. Choose the **Skeleton → Move Joint** or **Move Branch** command in the Actor module.
3. In either case, you can pick your selection by mouse or by name.
4. Follow the status bar instructions:
 - Left mouse button translates the joint according to the x-axis of the joint's parent.
 - Middle mouse button freely translates the joint on any axis (relative to the window plane).
 - Right mouse button exits the Move Joint mode.

You cannot move the first joint.

The next time you move the joint, the envelope may be deformed in unexpected ways. In this case, choose **Skin → Envelope Assignment → Reassign Automatically** to reassign vertices.

Changing the Preferred Angles

When you draw a chain, it is created with a “preferred position and orientation.” Depending on how you drew the chain, the “preferred angle” may not be correct. As a result, the knee may not bend at the point in 3D space or may not achieve the right angle.

To change this:

- Select the joint and rotate it in the z-axis; remember, you can only rotate a joint independently in the z-axis because inverse kinematics controls all other rotations. You can set the rotation for each joint separately to control bending at each point of articulation.
- Use the **Skeleton → Move Joint** or **Move Branch** command in free translation to put some rotation on the joint as well.

Defining a Skeleton

From Hierarchies to Skeletons

Typically, your final objective is not just to build a hierarchy of chains or objects and play with it. Instead, you want to apply a “skin” using one or more models, and to take advantage of inverse kinematics to move the actor around and have it deform naturally — or unnaturally, depending on your latest project!

To animate and deform the envelope using that hierarchy of chains or objects, the hierarchy must be defined as a skeleton.

An ordinary hierarchy of chains or objects becomes a skeleton hierarchy in either of these cases:

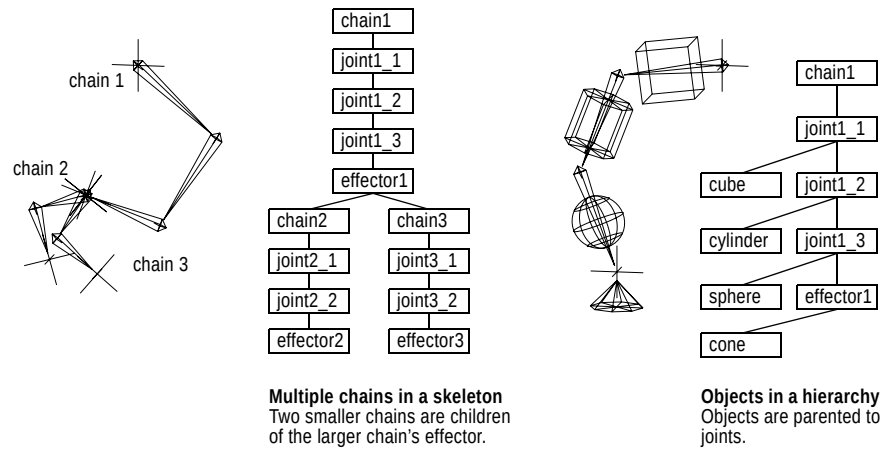
- As soon as you assign an envelope model to it, using the **Skin → Local Envelope, Global Envelope, Automatic Envelope, or Rigid Envelope** commands.
- If you select the hierarchy and choose **Skeleton → Group as Skeleton**. At that point, you can also assign certain envelopment details to the skeleton without yet assigning an envelope.

Creating the Skeleton Hierarchy

1. Draw one or more chains. Optionally, you may get or select any other object you wish to include in the skeleton.
2. To include more than one chain in a skeleton, parent one chain to another in the usual way (using the **Parent** menu cell) to form a hierarchy.

Normally, a skeleton has multiple chains, in which case you can make the effector of one chain the parent of the chain roots of other chains, and so on.

3. To include an object as part of a skeleton, make the object the child of the desired joint.



Defining It as a Skeleton

4. Select the hierarchy.

- If you have an envelope, you can assign it now. For information, see *Assigning an Envelope* on page 247.
- If you do not want to assign an envelope yet, choose **Skeleton → Group as Skeleton** and follow the status line instructions:
Left mouse button creates a skeleton from the selected hierarchy.
Middle mouse button undoes the creation operation.
Right mouse button exits the mode.

Results

As soon as you define a skeleton, there are several immediate effects.

In the Schematic window, all skeleton components (root, chain, models) are identified by the label “Skel.”

After you assign an envelope, the skeleton root model has the new label “SkRt.”

In all windows, skeleton components are displayed with their default colour coding.

The initial position of the skeleton is saved, although there is no visual feedback of the save. After you move the skeleton, you can reset the skeleton to the initial position at any point by choosing the **Skeleton → Reset Actor** command.

Levels of Hierarchy

Skeleton hierarchies have multiple levels. Each chain root, each joint, the effector, as well as any models used are all separate entities and “levels” in the skeleton. Thus, every component corresponds to a separate level in the hierarchy.

Bounding Models

As the name suggests, a bounding model describes the area of influence around the joint/element or limits the assignment of envelope vertices. For example, using a bounding model, you can make sure that points on a human torso “stay put” and do not follow the movement of the arm as it is raised.

You can create a bounding model for a skeleton without assigning an envelope, but these models are obviously more useful with an envelope being visible. For more information, see *Creating a Bounding Area for Envelope Vertices* on page 261.

Resetting the Skeleton

The **Skeleton → Reset Actor** command in the Actor module returns the skeleton (and its envelope, if any) to its initial position.

Note

The initial position is saved as soon as a skeleton is created, whether or not you have assigned an envelope.

For example, let’s say you create a chain of two joints with the “knee” bent at a 45-degree angle chain; if you then group this chain as a skeleton and alter that angle to 90 degrees, **Skeleton → Reset Actor** returns the actor to the original 45-degree angle.

Tip

It helps to reset the actor before doing a manual reassignment of envelope vertices.

1. If you select the envelope, the entire actor is reset. If there is a global envelope, you can reset individual skeleton branches if you wish. To do this, select the skeleton branch in branch selection mode.
2. Choose the **Skeleton → Reset Actor** command.

The skeleton (and any associated envelope) immediately returns to its initial position. If you selected a branch of a skeleton according to the conditions described in the procedure, only the selected branch returns to its initial position.

Using Objects in Skeletons

Using objects in a skeleton framework offers greater control of local deformation. It lets biceps ripple under the skin, or lets a cartoon character's knee “throb” up and down in an exaggerated fashion independently of the forward motion.

There are many techniques for shape deformation, such as lattices, cluster centres, deforming by surface or curve, and so on (see *Deforming an Object's Shape* on page 291). Several of these techniques offer local control of deformation, but all points are affected equally by the deformation.

By contrast, when you use objects as skeleton elements and assign an envelope, you can take advantage of:

- The way envelope assignment and weighting allows deformation over an area with subtler, diminishing effects.
- The ability to create and combine bounding models on skeleton elements.
- Using expressions to define relationships between parts of skeletons and parts of their envelopes, such as their related transformations and interpolations. For instructions and examples on using expressions, refer to the *Expressions* chapter in the *Animating User's Guide*.
- The ability to combine it with inverse kinematics (if skeletons combine objects and articulated chains). You can use expressions to relate the local deformation around the object with the inverse kinematics of the chains.

What Kind of Objects?

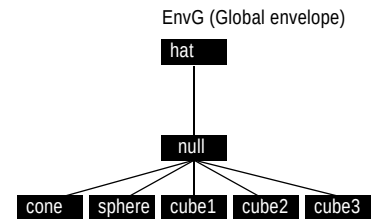
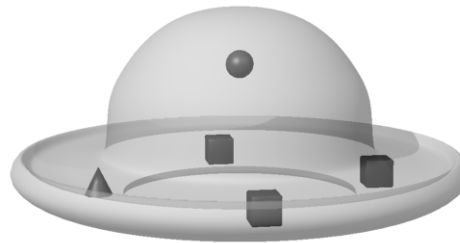
You can use any kind of object in a skeleton. The geometry of the skeleton object is really not significant because the vertices of the envelope will be assigned to the skeleton object's centre. Thus whether you use a cube, a sphere, or a null in the skeleton, all transformations of the surrounding skin are in relative size, orientation, and position to the deforming object's centre, not its geometry. The results will look exactly the same.

Null objects do not introduce extra geometry into the scene, but objects with geometry are sometimes easier to visualize and interact with. Be aware that if the skeleton object somehow overlaps the

envelope, it shows in the rendering, and you might have an unexpected lump on the envelope model's head!

Examples

In the following example, the skeleton is composed of three shapes: three cubes, a cone, and a sphere under a null parent. Nulls would work equally well, and the example shows you that the object geometry is not significant.

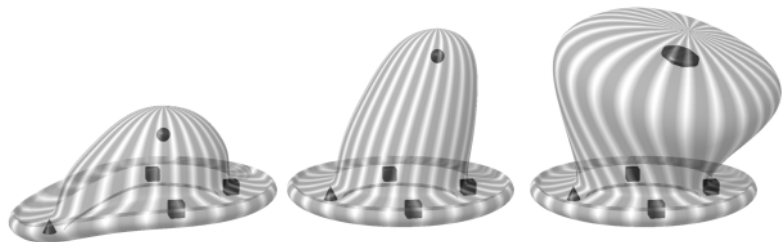


Hat as global envelope

Hat is a NURBS surface defined as a global envelope to the hierarchy of sphere, cone, and three cubes.

Hierarchy of objects as skeleton

Sphere, cone and cubes were used to illustrate the example. Nulls would produce equivalent results.



Deform the hat by altering the skeleton.

Translate cone (left); translate sphere (centre); scale, rotate, and translate sphere (right).

A skeleton of an arm could be composed of a chain and a sphere, scaled to the shape of a bicep muscle, positioned, and then constrained to joint_1 of the chain. To animate the deformation, a simple expression relates the scaling of the sphere (bicep) with the rotation of the forearm joint: as the arm flexes, the sphere is scaled to make a swelling bicep under the skin.

You could also have a face with a null object at the cheek level. Some envelope vertices are automatically assigned to the null. To give the character a nervous twitch without affecting the whole face, you

could apply an expression using a random time-related function to transform the null.

Restrictions

There are some restrictions on these skeleton structures:

- No vertices can be assigned to the null displayed as the chain root, or to a skeleton root of a local envelope.
- In the Envelope Weights dialogue box, only joints and effectors are listed, as well as models are displayed (except those which are the skeleton roots for local envelopes) for purposes of assigning or reweighting vertices.
- Automatic envelopes work only with chains, not other objects in the skeleton.

Colour Coding Joints and Vertices

When you define a skeleton, the skeleton elements (joints or objects) are automatically coloured using the default colours green, cyan, yellow, and magenta.

If, instead of loading a whole scene, you use **Get → Element** to load an actor (skeleton plus envelope), the actor's colours are lost. In this case, you can select the skeleton's colour coding manually:

1. Select a chain root.
2. Choose the **Skin → Envelope Assignment → Colour Skeleton** command in the Actor module.
3. Follow the mouse button instructions as described:
 - Left mouse button colours the entire chain and its descendants in the default colours.
 - Middle mouse button resets the colour to white.
 - Right mouse button exits the mode.

Changing the Default Colours

The **Skin → Envelope Assignment → Colour Joint** command allows you to change the default colour of skeleton joints or objects and the associated envelope vertices.

To change the default colours:

1. Select a joint.
2. Choose the **Skin → Envelope Assignment → Colour Joint** command in the Actor module.
3. Left or middle-click to toggle through the object colour palette (blue, red, green, cyan, yellow, and magenta).
4. Right-click to end colour selection.

Adding Skin (Envelopes) to a Skeleton

An *envelope* is any model that you assign to the skeleton or a part of it. As you would expect, once it is assigned, it moves with and is deformed by the skeleton (in most cases).

For details on assigning and editing envelopes, refer to the next chapter, *Envelopes*.

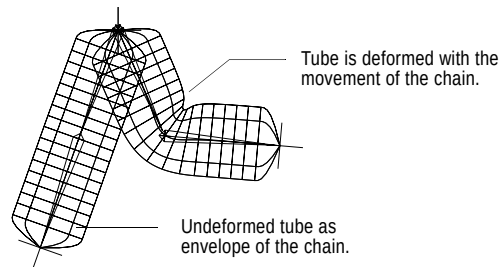
C H A P T E R E I G H T

Envelopes

About Envelopes

The previous chapter on skeletons described how skeletons are constructed, and how they help to animate and deform the “skin” or envelope which you associate with the skeleton.

An *envelope* is any model that you assign to the skeleton or a part of it; once assigned, it moves with and is (in most cases) deformed by the skeleton.



How Do Envelopes Move and Deform?

How envelopes respond to their skeletons depends on several things, such as:

- Is it an envelope which allows itself to be deformed or not? (flexible or rigid)
- Is the envelope assigned to the skeleton as a whole or to a part of the skeleton? (global or local assignment)
- Does the envelope model have geometry (as with most 3D objects) or not? (as with meta-clay objects)
- What kind of weighting formula did you use when assigning the envelope to the skeleton?
- Are there any bounding models to limit or fine-tune the envelope assignment to the skeleton?

What Can You Do with Envelopes?

The commands in the **Skin** menu in the Actor module allow you to:

- Create automatic envelopes, which saves modelling time.
- Create envelopes that are weighted relative to more than one joint. You can also modify this weighting at any time.
- View how vertices are assigned to joints by a colour coding scheme.
- Reassign parts of envelopes to different joints to improve their appearance and how they respond in accordance with the skeleton's movements. See *Reassigning Vertices Manually* on page 260 for more information.
- Define a bounding model to exclude certain vertices from the influence of skeleton models which would otherwise have an undesirable effect on their movement, or to force some vertices to be assigned to specific skeleton models. See *Creating a Bounding Area for Envelope Vertices* on page 261 for more information.
- In addition to assigning envelope vertices or using bounding models, you can choose a predefined curve profile and use it to deform the inner or outer curve of an envelope, as described in *Applying Deformation Curves to Envelopes* on page 265.

Assigning an Envelope

An envelope can initially be assigned to its respective skeleton (or to a part of a skeleton), then reassigned if you want to customize it.

There are five types of envelopes: Flexible, Automatic, Local, Global, and Rigid. You can select any of these in the **Skin** menu, except Flexible, which refers to Automatic, Local, and Global envelopes.

Before Assigning the Envelope

Creating Skeletons and Envelope Models

Create the skeleton hierarchy and the model you wish to use as a skin for all or part of it.

Optionally, you can group the skeleton before assigning the envelope and define some information related to the skin. This is not a necessary step, since the hierarchy is defined as a skeleton as soon as you assign an envelope to it.

Envelope Template Information

Some information about the skeleton's initial position, as well as any bounding models or deformation parameters you may have set, is saved when defining the skeleton.

You need not have created/assigned an envelope to save this information. Indeed, you can switch envelopes later and the previously defined information doesn't change.

This allows you to create a kind of "template" for enveloping a specific skeleton, which is applied no matter what envelope model you apply.

Positioning the Envelope

Careful positioning of the skeleton in its skin can improve the result. Typically, you should place the envelope model over the skeleton or the appropriate part of the skeleton:

- Position a joint where you want the envelope to bend.
- Place skeleton models at the centre of the region of the envelope you want to affect.

You can place envelopes away from the skeleton, but this might give unexpected results.

Assigning an Envelope Initially

Initially, envelopes are assigned to skeletons using an algorithm which is based on the proximity of any given vertex to one or more skeleton elements. The “relativity” is determined by your choices in the Envelopes Initial Assignment dialogue box, explained in the next section.

The dialogue box is the same for all envelope types, but the exact procedures for selecting and assigning vary according to the envelope type.

Envelopes Initial Assignment

Assignment Method:

☐ Full weight (old method)

☒ Weighted

Default nb. of assigned joints:

Assignment Depth:

☒ No maximum

☐ Maximal level

You can choose to assign vertices in one of two ways: 100% weighting to the nearest joint, or distributed weighting over a specified number of joints, based on relative proximity.

Full Weight Assignment

Each vertex of the object is initially mapped to the joint closest to it. Each vertex is assigned with a weight of 100%.

Weighted

Each vertex is assigned to the number of joints determined in **Default nb of joints**; the assigned vertex is weighted relative to the distance of each. By default, each vertex is assigned to the two nearest joints, with a percentage weight based on proximity. For example, a

vertex close to the mid-point of a joint may be assigned 100% to that joint, whereas a vertex closer to the point of articulation with the next joint probably has a weighting more like 60/40 or 80/20 and so on, depending on the relative proximity to each.

The weight value modifies how the joint movement affects the selected vertex. You can assign a vertex to n joints, but the total weight value for a vertex cannot exceed 100%.

Note

These choices are for initial assignment. In the case of flexible envelopes you may subsequently reassign a vertex to more or different joints as you wish, using the Envelope Weights dialogue box, even if initial weight assignment was Full Weight.

Default nb of Joints

Vertices are assigned according to their relative proximity to n joints. The default is 2, which means that vertices are assigned to a maximum of 2 joints, although many vertices will probably be assigned 100% to one joint, by reason of proximity.

Assignment Depth

Skeleton hierarchies have multiple levels. Each chain root, each joint, the effector, as well as any models used are all separate entities and levels in the skeleton. Thus, every component corresponds to a separate depth in the hierarchy.

By default, all levels of the skeleton hierarchy are included in the envelope assignment.

Initial Assignment and Bounding Models

Initial assignment also considers any bounding models you may have created before assignment. Bounding models help refine the zone influence of a joint, in relation to the surrounding envelope, as described starting on page 261.

The initial assignment algorithm should do all or most of the work. After that, if you see undesirable deformations of the envelopes around joints, you can “tweak” the assignment for these vertices. In some cases, you must choose the **Skin → Envelope Assignment → Reassign Automatically** command after creating bounding models. They do not execute automatically. For more information, see *Editing an Envelope* on page 259.

Assigning Flexible Envelopes

Any envelope model which allows itself to be deformed by the skeleton elements is a flexible envelope.

Note

There is no specific command in the **Skin** menu for flexible envelopes; rather, it is the default type of envelope. Local, Global, and Automatic Envelopes are *all* flexible envelopes. The exception is, of course, Rigid Envelopes.

Flexible envelopes can be made from any object that has geometry; that is, any object except a face or meta-clay element.

Characteristics

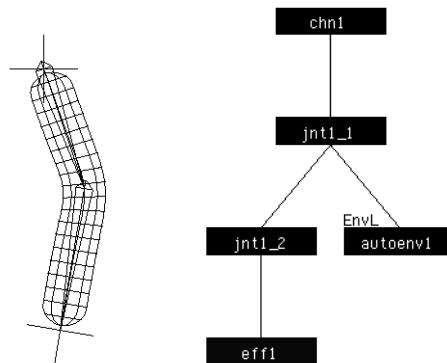
When assigned to skeleton chains, flexible envelopes map their vertices to the nearest skeleton joints. When assigned to skeleton objects, flexible envelopes map their vertices to the centre of the nearest skeleton object(s).

Flexible envelopes offer more local control than rigid envelopes because you can tag their vertices, reassign these vertices to different joints, and modify the weight of vertices relative to one or more joints.

If you scale a skeleton, a flexible envelope is also rescaled accordingly.

Assigning an Automatic Envelope

Automatic envelopes are, no surprise, created automatically and assigned to the chain in a one-step process, which makes them convenient for building envelopes quickly. They are assigned as local



Chain with automatic envelope.
Envelope is the child of the first joint of the chain.

envelopes to the skeleton or part of skeleton you select. An automatic envelope can only be assigned to a skeleton chain, not to an object in a skeleton. It is assigned as a local envelope, the child of the first joint of the chain to which it is assigned.

This method is particularly useful for an initial view of seeing how an actor moves.

1. If you want to use custom curve for the envelope shape, draw a curve (**Draw** → **Curve** in the Model module) in the xy plane.
2. Select the chain you have created.
3. Choose the **Skin** → **Automatic Envelope** command.
4. Follow the mouse line instructions:
 - Left mouse button allows you to pick the curve you drew and use it for the automatic envelope.
 - Middle mouse button selects a default circle for the envelope.

The curve or default circle is automatically extruded along the length of the chain, creating and assigning the envelope in one step.

5. The Automatic Envelope dialogue box is displayed.
6. Select the type of envelope (**Patch** or **NURBS**) if you are using the default circle. You can also select **Capping** for the top or bottom of the envelope, or set the **Steps per joint** which defines the subdivisions of the envelopes. Click **Ok**.
7. The Envelope Initial Assignment dialogue box appears, as shown on page 248.
8. Set the assignment and weighting parameters and click **Ok**.

Substituting the Final Envelope Model

Once you are satisfied with the result, you can create your own custom envelope models and assign them to the actor.

To do this, you would:

1. Cut all automatic envelopes from the chain using the **Cut** menu cell in the Actor (or Model) module.
2. Position the skeleton hierarchy inside your new detailed envelope model.
3. Assign the envelope, using local or global assignment, depending on whether you wish it to assign to several chains.

Assigning a Local Envelope

When you assign a model as a local envelope, it becomes the child of the first joint of the associated skeleton chain. As such, it is deformed by the chain and the children of the chain's effector, if any.

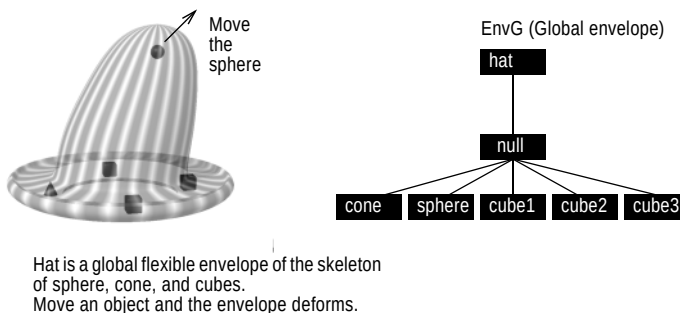
If the skeleton is made of objects, the envelope becomes the child of the skeleton root. No envelope vertices are assigned to the skeleton root.

You would normally use a local envelope to animate a part of a model independently. For example, if you wanted to create a dog with a wagging tail, you would probably use a local envelope for the tail. Local envelopes are, by default, flexible.

1. Select the skeleton chain root or object.
2. Choose the **Skin → Local Envelope** command.
3. Pick the object to assign as envelope. The Envelopes Initial Assignment dialog box is displayed, as shown on page 248.
4. Set the assignment and weighting parameters and click Ok.

Assigning a Global Envelope

When you assign a model as a global envelope, it becomes the root of a hierarchy which can include one or more chains or skeleton hierarchies, with or without other global envelopes.



You would normally use a global envelope to animate a model composed of several chains that work together, such as a hand. This type of envelope still deforms in accordance with the skeleton's movements.

1. Select the envelope model.
2. Choose the **Skin → Global Envelope** command.

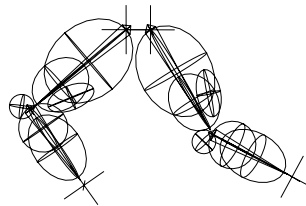
3. Pick the chain(s), object, or hierarchy of objects to envelope. The Envelopes Initial Assignment dialogue box is displayed, as shown on page 248.
4. Set the assignment and weighting parameters and click Ok.

Assigning a Rigid Envelope

Rigid envelopes simply follow the movement of the skeleton object, but are not deformed by this movement. Nevertheless, the term “rigid” is slightly misleading because you can use meta-clay elements, which are anything but rigid.

Using Meta-clay Elements

Rigid envelopes can be made from any kind of object, but are especially designed to make use of meta-clay elements as envelope models, with the resulting motion being more realistic. Some good examples of meta-clay elements as rigid envelopes are a hand, or even a full body character, using a fairly low number of meta-clay elements.



Chain with meta-clay system as a rigid envelope.

You could simply make the meta-clay element the child of a particular joint to use it in a skeleton, but then you would need to make numerous joints and parent a meta-clay element to each one to achieve realistic movement. Instead, by assigning meta-clay elements as rigid envelopes, you can achieve realistic movement using few joints.

Envelope Hierarchy

Rigid envelopes must be hierarchical models (not single objects). For rigid envelopes, be aware that the root model is never assigned to a joint; otherwise, you would get a transformation update cycle.

Envelope Transformations

Rigid envelopes map the *centre* of the envelope model to the nearest joint or joints. They use global envelope assignment, so they are the parent of the chain to which they are assigned.

Only translations and rotations of the skeleton chain are mapped for rigid envelopes. The current envelope scaling is preserved.

This means that the envelope follows the joint as it is scaled, keeping the same relative position, but the envelope model itself does not get scaled. For example, if you scale the chain, the scaling is not reflected in the rigid envelope.

This is useful when using meta-clay elements as rigid envelopes because you can scale meta-clay elements independently of their movement in relation to the joints.



Null objects are frequently used as the root model of a rigid envelope hierarchy, although any object can be the root.

If you use meta-clay systems as rigid envelopes, a root object is automatically created so you do not have to create one.

Here's the quick procedure for assigning rigid envelopes.

1. Select the envelope hierarchy.
2. Choose the **Skin → Rigid Envelope** command.
3. Pick the chain or skeleton object to envelope. The Envelopes Initial Assignment dialogue box is displayed as shown on page 248.
4. Set the assignment and weighting parameters and click Ok.

For plenty of details, see also **Skin → Rigid Envelopes** on page 1322 and **Skin → Envelope Assignment → Modify Weights** on page 1329 of the *Reference Guide*.

Choosing an Envelope Type

There are many ways to “skin” the proverbial cat in SOFTIMAGE 3D, but here are a few suggestions to think about when deciding what kind of envelope would suit your modelling purposes, and how best to integrate them into hierarchical structures.

Flexible or Rigid?

If you want to model something organic, you can use flexible envelopes for their deforming capabilities, but you could also use a meta-clay system and assign it as a rigid envelope to the skeleton.

Meta-clay is great for modelling liquids and soft shapes, however, there are some disadvantages to using meta-clay. For example, you can only use a single material and texture on the whole meta-clay system.

You can also create a model in meta-clay to take advantage of its organic qualities, convert the meta-clay model to a polygon mesh object, and finally use the Polygon Reduction effects to optimize the resulting model. For more information on Polygon Reduction and other optimizing effects, see *Optimizing Models* on page 317.

You probably won’t use rigid and flexible envelopes in the same hierarchy, but you might use rigid envelopes for “accessories” and flexible envelopes for the main model.

For example, if you were modelling a fish with a floppy fins, you could use a flexible envelope for the fish, but the fins might be meta-clay elements used as rigid envelopes. You could also simply make the rigid envelope hierarchy the child of a specific skeleton element located at the appropriate spot on the fish.

Local or Global?

The main body of a person or animal is usually assigned as a global envelope to the corresponding skeleton hierarchy. If you later add anything to the resulting hierarchy on the fly, you probably need to reassign the envelope.

By contrast, “peripheral” body parts such as the mane, the tail or the ears of a horse, or the beard of a goat, are often assigned as local envelopes to a skeleton chain. To attach the tail to a horse’s behind, you can make the tail the child of the first joint of the spine.

The advantage to this approach is that a local envelope hierarchy such as the tail can be attached to the global envelope hierarchy (main body) without affecting the assignment of the global envelope.

Likewise, you could assign a mane or ears as local envelopes and then attach them to the main body by selecting any bone of the head and making it the parent of the ears. This doesn't disturb anything in the assignment of the global envelope.

Viewing Envelopment Assignment

You can get a good idea of which parts of your envelope are assigned to which joint using visual cues, or display the exact weight assignment in percentages of a selected vertex.

Using Colour Cues

When you define a skeleton, the skeleton elements (joints or objects) are automatically coloured using the default colours green, cyan, yellow, and magenta.

1. Select the envelope.
2. To display the vertices' assignment, choose **Skin → Envelope Assignment → Show Vertices**.
3. If the skeleton has any kind of flexible envelope (automatic, local, global), the envelope vertices are colour coded to show their assignment.
 - **Coloured:** The vertex is assigned to the joint of the same colour with a weighting of more than 50%.
 - **White:** The vertex is not assigned to any joint with a weight of more than 50% or it is not assigned to any joint.

Based on the colour coding scheme, you can then decide whether to reassign selected vertices.

 If, instead of loading a whole scene, you use **Get → Element** to load an actor (skeleton plus envelope), the actor's colours are lost. In this case, the **Skin → Envelope Assignment → Colour Skeleton** command is especially useful.

If you wish to view the exact weighting of a vertex, tag the vertex and choose **Skin → Envelope Assignment → Modify Weight**.

Changing the Default Colours

You can also change the default colours of skeleton elements and the associate envelope vertices using the **Skin → Envelope Assignment → Colour Joint** command.

1. Select a joint.
2. Choose the **Skin → Envelope Assignment → Colour Joint** command.

3. Left or middle-click to toggle through the object's colour palette (blue, red, green, cyan, yellow, and magenta).
4. Right-click to end colour selection.

Viewing Weights of Vertices

If you want to view or edit the weighting of selected vertices, tag the vertices and choose **Skin → Envelope Assignment → Modify Weights**.

The Envelope Weights dialogue box appears in which you can select a vertex, and display or edit its joint assignment or weight values. For more information on this dialogue box, see *Modifying Envelope Weights* on page 265.

Editing an Envelope

After assigning the envelope, you may need to adjust the assignment or weighting of certain vertices to correct or exaggerate the way the envelope is deformed by the skeleton movement.

There are a few techniques for editing the look and behaviour of an envelope. Some techniques let you edit the envelope indirectly by adjusting its assignment to the skeleton; others let you apply the changes directly to the envelope model itself.

- To reassign vertices to different skeleton elements, you can:
 - Tag vertices and manually reassign them to a skeleton element (joint, object, or effector).
 - Create bounding models around the joints or around vertices on envelopes to prevent or enforce envelope assignment to certain skeleton elements.

The previous methods are more visual, useful for changing or limiting which elements vertices are assigned to which parts of the skeleton.

- If you want to make finer adjustments, you can tinker with the weighting of vertices. You can tag vertices and display a dialogue box in which you specify the weighting of these vertices in relation to skeleton elements, according to different weighting formulae.
- Finally, you can deform the envelope using a curve profile and apply some explicit deformation parameters.

Reassigning Envelopes and Vertices Automatically

Some editing operations need an automatic reassignment to apply the changes. For example, if you assign a global envelope to a skeleton hierarchy and then insert a new element into the hierarchy (a chain, object, or bounding model), the new element is not recognized by the global envelope until you choose **Skin → Envelope Assignment → Reassign Automatically** or **Reassign Manually**.

The **Skin → Envelope Assignment → Reassign Automatically** command automatically resets the assignment and weighting to the initial (undeformed) assignment. It can be used for flexible and rigid envelopes.

1. Select an envelope.

2. Choose the **Skin → Envelope Assignment → Reassign Automatically** command.

A message warns you that automatic assignment will destroy the current assignment. This means that any manual reassignment or weight modifications will be lost because all vertices will be reassigned to the nearest element, regardless of their current weight values. The reassignment will nevertheless take into account any bounding models.

3. The Envelopes Initial Assignment dialogue box is displayed, as shown on page 248, in which you can again choose the weighting method (**Full Weight** or **Weighted**) by which to reassign the envelope.
4. Once you fill in the appropriate parameters in this dialogue box, all vertices are reassigned to the nearest skeleton element(s).

Reassigning Vertices Manually

The **Skin → Envelope Assignment → Reassign Manually** command allows you to reassign tagged vertices to another skeleton element or to remove the vertex's assignment from an element.

This operation is only applicable to flexible envelopes (all types except Rigid envelopes).

Tips

- The **Reassign Manually** command ignores any bounding models.
- Whenever vertices are assigned to parts of skeletons (whether automatically or manually), assignment is always based on the initial, undeformed position. Before reassigning points manually, it helps to reset the actor to its initial position to visualize the starting point before you start reassigning. Use the **Skeleton → Reset Actor** command to do this.
- If you subsequently use the **Skin → Envelope Assignment → Reassign Automatically** command, it overwrites any manual reassignment information.

Procedure

1. Select the envelope and show its vertices using **Skin → Envelope Assignment → Show Vertices**.
2. Tag the vertices you want to reassign using the **t Supra** key.

3. Choose the **Skin → Envelope Assignment → Reassign Manually** command.
4. Pick the new joint for the tagged vertices using the left mouse button. Middle-click to unassign the tagged vertices. The vertices are reassigned or unassigned.

Reassigning Rigid Envelopes

For rigid envelopes, you can either:

- Use the **Skin → Envelope Assignment → Reassign Automatically** command.

or

- Modify their weighting values using the **Skin → Envelope Assignment → Modify Weights** command.

Creating a Bounding Area for Envelope Vertices

You can select any skeleton element and create a bounding model around it using the **Skin → Envelope Assignment → Bounding Model** commands.

As the name suggests, a bounding model describes the area of influence around the joint/element or limits the assignment of envelope vertices. For example, using a bounding model, you can make sure that points on a human torso “stay put” and do not follow the movement of the arm as it is raised.

Note

You can create bounding models around skeleton elements even without an envelope, although the purpose is to control the assignment and weighting of envelope vertices to the selected joint.

You can adjust the effects after you have assigned the envelope. Bounding model information is saved with the skeleton, even if you cut the envelope.

You can choose to display the bounding model as a box, cylinder, or sphere. All shapes can be transformed, but only bounding boxes have points that can be edited. Bounding models are displayed only when the **Show → Controls** command is selected. Since they are the children of the joint, they have the same colour coding.

You can select from three different behaviour: Inclusive Limit, Inclusive, or Exclusive, as described in *Types* on page 262.

Restrictions

Bounding models are only applied on initial assignment and automatic reassignment. If you apply a bounding model after you have already assigned an envelope, you must choose **Skin** → **Envelope Assignment** → **Reassign Automatically** to actually implement the changes of assignment.

Note

Bounding models are not useful for rigid envelopes: see *Assigning a Rigid Envelope* on page 253.

Types

Two bounding model types have an “all or nothing” behaviour. The other one combines total assignment of some vertices and diminishing assignment of others.

Inclusive Limit

This type of bounding model defines the skeleton element’s zone of influence: any vertex falling outside its boundary cannot be assigned to the element.

Note

This type of bounding model relates more to the skeleton than to the envelope. It should be drawn and positioned in relation to the skeleton elements since you want to define the region you want the skeleton element to influence.

Inclusive

Any vertex falling within the bounding model is assigned 100% to the associated joint without regard to relative proximity to the skeleton element.

Furthermore, points falling outside any inclusive bounding model are assigned and weighted (according to their relative proximity) to the n closest joints; n is defined by **Default nb of joints assigned** in the Envelope Initial Assignment dialogue box. The default is 2.

Note

When building Inclusive models, think “envelope.” For example, position your envelope over the skeleton first. Tag the part of the envelope you want to include or exclude (tagging serves as a visual cue and is not necessary for creating the bounding model). Draw your bounding model in relation to it.

Exclusive

Any vertex falling within the bounding model is prevented from being assigned to the associated skeleton element.

The same tip applies for Inclusive and Exclusive bounding models: position your envelope over the skeleton first, tag the part of the envelope you want include or exclude, and draw your bounding model in relation to it.

Combining Bounding Models

There are occasions when you would want to combine bounding models for useful purposes.

Overlapping Different Bounding Models

For example, to achieve a progressively diminishing influence of the joint, you could use an Inclusive bounding model around joint_*a*, then a bigger Inclusive Limit box around it.

- Vertices inside the Inclusive model are assigned 100% to the selected joint.
- Vertices in the spillover area (outside the first bounding model but within the larger Inclusive Limit boundary) are probably assigned to joint_*a* at something less than 100% since they are farther away.
- All vertices outside the limit are prevented from being assigned to joint_*a* at all.

Overlapping Inclusive Bounding Models

Sometimes you may want to force some points to be assigned to certain skeleton elements (joints or objects), but not move deform in an unnatural “separated” manner.

To achieve this, create two or more inclusive bounding models which overlap near the articulation of two joints (or other skeleton elements). The envelope vertices which fall within the overlapping region of both Inclusive bounding models behave like this:

- They are assigned *only* to the respective skeleton elements, regardless of the value in the **Default nb of joints** parameter.
- They are still weighted according to their relative proximity to the two skeleton elements.

- Envelope vertices that fall only in one model are assigned with the usual 100% weighting to the associated skeleton element.

Creating Bounding Models

1. Create some skeleton elements (chains or objects).
2. Make them into a skeleton hierarchy by choosing the **Skeleton → Group as Skeleton** command, or assign an envelope to the hierarchy. (Assigning an envelope automatically identifies the hierarchy as a skeleton hierarchy).
3. To check vertices' assignment, you can either:
 - Select the skeleton hierarchy and choose the **Skin → Envelope Assignment → Show Vertices**. Verify the colour assignment of vertices to skeleton elements.
 - or
 - Tag vertices on the envelope and choose the **Skin → Envelope Assignment → Modify Weights** command as a visual cue. The Envelope Weights dialogue box appears in which you can verify exact weighting of vertices.
4. You can define any bounding model before assigning an envelope. Draw Inclusive Limit bounding models in relation to the skeleton; draw Inclusive and Exclusive bounding models in relation to specific parts of the envelope.
5. Select a single joint and choose **Skin → Envelope Assignment → Bounding Model → Inclusive, Inclusive Limit, or Exclusive**.
6. Select the shape of the bounding model using the mouse buttons as defined in the mouse line.

The bounding model is displayed in the same colour as the associated joint.

Editing Bounding Models

1. You can scale, rotate, and translate the bounding model.
2. For bounding boxes, you can also edit their geometry by first choosing **Show → Points** to show the points. You can then select and edit any of its points as you would any object.
3. If you assigned an envelope before building the bounding model, choose **Skin → Envelope Assignment → Reassign Automatically**

to overwrite the previous assignment with your bounding model data.

Note

Repeat this step any time you edit the bounding model.

Modifying Envelope Weights

After assigning the envelope, you may need to fine-tune the assignment or weighting of certain vertices to correct or exaggerate the way an envelope is deformed by the skeleton movement.

1. Tag the vertices you wish to reweight.
2. Choose **Skin → Envelope Assignment → Modify Weights** command. The Envelope Weights dialogue box appears.
3. You can:
 - Select one or more of the vertices in the **Tagged Vertices** scroll box and view their assignment and weighting in the **Joints/Weights** scroll box.
 - Edit the weighting of only the selected vertices or all the vertices in the scroll box.
 - Set the new weighting in absolute terms or additive terms (either add or subtract a percentage of the current weighting).

Restrictions and Tips

It is often easier to use bounding models or manual reassignment to reassign envelope vertices than to use the Envelope Weights dialogue box. This dialogue box is useful primarily for adjusting the weighting of a few vertices to skeleton parts.

The interaction of parameters in this dialogue box is somewhat complex and should be understood before using it. For more information, see *Modify Weights* on page 1329 of the *Reference Guide*.

Applying Deformation Curves to Envelopes

For greater control of envelope deformation, you can choose a predefined curve profile and use it to deform the inner or outer curve of an envelope. Alternately, you can draw your own curve profile and use it in the same way.

Either way, you can only set deformation parameters for flexible envelopes (Automatic, Local, or Global) and a skeleton composed of chains (not objects).

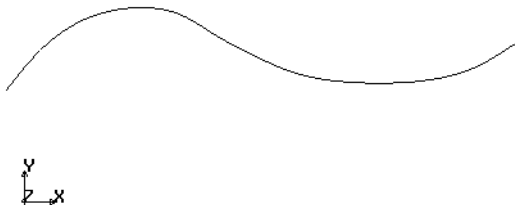
These deformation parameters are added to the effects of bounding models and vertices' reassignment.

Deformation information is saved with the skeleton, even if you cut the envelope.

Using a User-Defined Curve

1. Using the **Draw → Curve** commands in the Model module, draw a curve to use as a deformation profile.

The curve must be a xy function meaning that it is drawn in the xy plane and continues in one direction as below:



The line cannot be looped, or “reverse” on itself, like this:



2. Select a joint.
3. Choose the **Skin → Deformation Curves → Select Inner Curve or Outer Curve** command.
4. Left-click to assign the curve or middle-click to unassign it.

The envelope deforms to the shape of the curve, to the extent that the resolution of the envelope permits.

You can now modify the deformation profile of the curve using the **Skin → Deformation Parameters → Inner Side/Outer Side** commands.

Using a Predefined Curve

You can also use a predefined curve profile to deform the inner or outer side of the envelope. Inner and outer sides of the envelope

correspond to the way the joint bends. On an arm, the inner side is the bicep and the outer side is the tricep.

1. Select a chain root or joint to which you have already assigned a flexible type of envelope (Automatic, Local, or Global).
2. If you select a chain root, the whole envelope is affected along the inner or outer side of all joints when they bend. If you select a single joint, the envelope is affected by the deformation settings only around the selected joint when it bends.
3. Choose one of the **Skin → Deformation Parameters → Inner Side/Outer Side** commands. The Flexible Envelope Parameters dialogue box appears.
4. Select the **Finger** or **Bicep** deformation profile appropriate to the type of deformation you want to create.
5. Set the **Beginning** and **End** values to define the start and end points of the deformation. These are expressed as a percentage of the length of the selected joint(s).
6. Set the **Amplitude** (strength) of the deformation: 0% shows no deformation and 100% doubles the envelope thickness at the point where the deformation peaks.
7. To exaggerate the deformation more, adjust the **Scale** value.
8. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.
As you interact with the effector, you see the effects of your deformation settings on the envelope.

Changing the Joint Rounding

You can determine the percentage of a joint that is affected by rounding as the joint bends, up to a maximum of 50%, using the **Skin → Deformation Parameters → Joint Area** command.

1. Select the joint or chain.
2. Choose the **Skin → Deformation Parameters → Joint Area** command. The Joint Area Parameter dialogue box appears.
3. Use the **Joint Area** slider to set the percentage of joint rounding.
You should use values at or near the default value as follows:
 - For joints with **Weighted** envelopes, use 0 or low joint area values.
 - For joints with **Full Weight** envelopes, use 0.2.

4. Select **Auto Refresh** to automatically update the scene with the modifications you make, or deselect it and click the **Refresh** button only when you want to see the changes you have made.
5. Click **Ok** to accept the settings or **Cancel** to exit the dialogue box.

Undoing Envelope Operations

When you delete a model and then choose **History** ➔ **Undo**, some related information may be destroyed by the delete operation and this information is not restored when you undo the deletion.

When you cut or delete an envelope, the assignment data is lost; undoing the deletion recreates the hierarchical links, but the envelope is no longer assigned.

Nevertheless, some data related to the joints themselves, such as joint area parameters, bounding models, and so on, is retained. This simplifies the tasks of assigning (another) envelope to the skeleton.

C H A P T E R N I N E

Null Objects

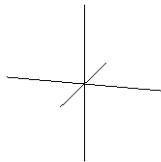
Introduction

A *null object* is an object which has no geometry. Because of this, it is not visible in a rendered scene and it cannot respond to any geometric modification, such as deformation.

A null is basically a centre represented by an icon. By default, it is displayed in the windows as a three intersecting lines. In the Schematic window, it is shown as a rectangle. The centre cannot be dissociated from the null, unlike other object types which allow you to move their centre inside or outside of them.

Other objects have control points added around their centres, points which define specific object types as polygon mesh, surface, or curve. No such points can be added to a null.

The null's cross icon can be easily mistaken for other objects of similar appearance. For example, if you use a null as a constraint on part of a skeleton, it can be difficult to distinguish it from the (similar-looking) chain roots and effectors, especially against the background of a grid. In this case, try using curved objects for the constraint, such as a linear circle (radius 1 or 2) or a short curve. While this takes up a little more disk space when saved, it makes selection easier and gives you all the abilities of a null.



Null object



Null object in Schematic window

Creating a Null Object

There are two basic ways of creating nulls: getting one directly or having one created when performing certain tasks.

To create a null object from scratch, choose the **Get → Primitive → Null** command. The null icon appears.

These are some commands in which a null is automatically created:

- When you choose the **Parent** menu cell without any object selected, then pick a child, a null is created as that child's parent.
- When you use the **Duplicate → Repetition** or **Duplication → Instance Repetition** command to create multiple duplicates or multiple instances of objects, the result is a hierarchy with a null object as the parent.
- When you create a cluster centre using the **Tag → Set Cluster** command, the cluster centre is a null object.
- When you choose **Get → Text** to create text, a null is created as a “base character” for each letter.

Using Nulls

Typically, nulls are used as assembly tools, especially useful in determining how objects will move in relation to each other or how material and texture is propagated through all or part of a hierarchy.

You could use any type of object to impose constraints or as nodes in a hierarchy. However, nulls are the best tools for the job because they have no overhead in modelling, rendering, or disk space.

Nulls are especially useful for occasions when you do not want any graphics associated with a function – see *Moon and Planet* on page 277.

Nulls in Hierarchies

In hierarchies, nulls are used as a centre of reference, mainly for determining movement and propagating materials and textures.

Centres and Transformations

Sometimes it is important to have independent centres on which to apply transformations. You can add as many levels of null objects as you wish for controlling transformations locally and separately. See the example of the plane on page 277.

Colour and Texture Propagation

You can use a null as a parent to several objects in a hierarchy and assign the null a colour which is then propagated to all its children. For more information, see *Creating a Hierarchy for Material Definition* on page 42 of the *Defining Materials and Textures User's Guide*.

Nulls as Constraints

You can use nulls to act as constraints, just as you would use other objects. For information on working with constraints, see *Constraints* on page 167 of the *Animating User's Guide*. See page 273 of this chapter with its suggestion for avoiding confusion between nulls used as constraints and other nulls used for different purposes in the same scene.

Cluster Centres

When you create a cluster, you can give it its very own centre, which is a null object. The centre is a separate object, and is the child of the

object which has the cluster. Cluster centres can be selected, parented, animated, and deleted. Clusters centres are also useful for constraints: for example, you can constrain an object to a cluster centre or vice-versa. For information, see *Constraining between Clusters and Objects* on page 182 of the *Animating User's Guide*.

Reference Objects for Precision Modelling

The **Polygon → Reference Frame** command in the Model module is used for precision modelling of polygon mesh objects. This command creates a null object with its coordinate system aligned with a selected polygon. Since the reference null lines up with the centre and local x, y, or z axes of a polygon on a polygon mesh object, you can draw in precise reference to that centre. For more information and an example, see *Precision Modelling on Polygon Mesh Objects* on page 72.

Similarly, when you select the Reference (REF) transformation mode, you can rotate or translate the selected object around the centre of another (called a reference object) according to the global axes' orientation of the reference object. Nulls can be used as a reference object, as can any other object.

Nulls in Skeletons and Envelopes

The chain root of an articulated chain, as well as the end effector, are null objects. Skeleton hierarchies can include objects as well as articulated chains and, once again, nulls can be included as elements in the skeleton hierarchy. For more information, see *Using Objects in Skeletons* on page 238.

Similarly, null objects are frequently used as the root model of a rigid envelope hierarchy, which is used as a skin for a skeleton hierarchy. Rigid envelopes are described in *Assigning a Rigid Envelope* on page 253.

Object Instances

When you choose **Info → Selection** on an object instance, the **Null Info** dialogue box appears, containing the information that it is an instance. Instance geometry is driven by that of the master object. However, you can scale, rotate, and translate instances independently. For more information, see *Creating Instances of Objects* on page 33.

Examples Using Nulls

Plane

If you apply all three rotation values to the centre of a plane, the effect between keyframes may be undesirable. Instead, create the following hierarchy: a null parent of a null which, in turn, is the parent of the plane. Then apply the y-rotation to the first null, the x-rotation to the child null, and apply the z-rotation to the plane itself.

The result is more realistic rotations and banking even between keyframes.

Moon and Planet

Say you wanted to model a moon rotating on its own axis, and also revolving around the earth at a faster rate than the earth is rotating and revolving.

To make the moon rotate around its own centre, animate the moon's rotation in y. You could make the earth the parent of the moon so that by animating the earth's rotation, the moon would revolve with it. However, this would not give the realistic appearance of different speeds.

Instead, you could place a null in the centre of the earth, make the null the parent of the moon, and animate the rotation of the null. In this way, you can independently animate the world's rotation at a different speed than the speed of revolution of the moon.

Furthermore, if you constrain the null's position to the centre of the earth, you can move the earth and everything will still move and rotate realistically.

Ferris Wheel

To model a ferris wheel in which the passenger seats revolve in sync with the wheel but always remain right side up, you could combine the effects of parenting and constraining with nulls:

1. Create two nulls: place one in the centre of the wheel and the second in the centre of the passenger seat.
2. Parent the ferris wheel to the null at its centre, then make that null the parent of the null in the seat.

3. Constrain the position of the passenger seat to the position of the null at its centre.

If you animate the rotation of the ferris wheel, the effect is that the passenger seat rotates with it (through the effect on the null at its centre), but keeps its upright orientation, just as it should if you don't want its passengers to fly out!

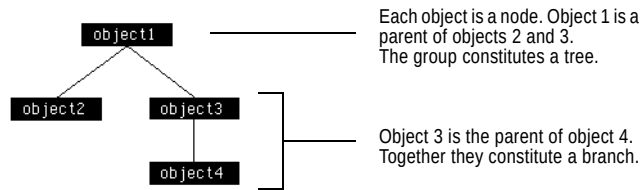
C H A P T E R T E N

Hierarchies

Introduction

Objects can be grouped together in a hierarchy for a number of reasons, such as to facilitate manipulation or to share material. This is normally referred to as *parenting*. In a hierarchy there is a parent, its children (the branches), the children's children (the nodes), and so on.

The combination of links that can be created is almost infinite.



Hierarchies are clearly shown and easily manipulated in the Schematic window. For more information on it, see *Using the Schematic Window* on page 79 of the *Working with SOFTIMAGE 3D User's Guide*.

Materials can also be propagated from the parent in a hierarchy to its children. For more information, see *Creating a Hierarchy for Material Definition* on page 42 of the *Defining Materials and Textures User's Guide*.

Creating a Hierarchy

When creating a hierarchy, you should first analyze how you want the objects in the hierarchy to interact so that you can place them in the correct position.

The second step is to position the centre of each object so that it can be rotated or scaled as desired. This procedure is usually done while modelling.

Once all the objects are created and positioned, you can create the hierarchy. Here's how:

1. Select the object you wish to be the parent.
2. Choose the **Parent** menu cell in the Model or Actor modules.
3. Pick each of the children. Picking means to click without holding down the space bar as you do when selecting.



When you choose the **Parent** menu cell without any object selected, then pick a child, a null is created as that child's parent.

4. The most important step is to right-click to end the parenting mode. Otherwise, the next object you click will also be a child in the hierarchy, and so on, into a never-ending family tree!

There are no restrictions on whether you start creating your hierarchy by defining the parent or the children.

Selecting Elements in a Hierarchy

When objects are in a hierarchical relationship, you can use the different mouse buttons to select a single object in a hierarchy, a branch of the hierarchy, or the whole tree.

This is different than selecting multiple individual elements in **Multi** mode because it respects the hierarchy links that are created. For more information on using hierarchies.

Selecting with the Mouse Buttons

Selecting a Node

1. Press the space bar.
2. Left-click on any object in the hierarchy, whether it is a parent or a child.

Only the object is selected, the rest of the branch is not.

Selecting a Branch

1. Press the space bar.
2. Middle-click on the parent of a branch. The branch is selected.

This works only on the parent. If you middle-click on the child, it does not select the branch.

With this type of selection, you can only deselect the whole branch and not individual elements. For an alternative, see *Multi-Node Selection* below.

Selecting the Tree

1. Press the space bar.
2. Right-click on any object in the hierarchy.

The entire tree containing the object is selected.

Multi-Node Selection

When you choose the **Select ► Multi-Node** command and select a hierarchy, you can then deselect individual nodes, as desired.

Note

This is the same as if you node-selected individual elements of the branch or tree with the **Multi** menu cell selected.

The approach you choose depends on whether you want to apply an operation to a only few elements in a large hierarchy, or to most of them.

In both cases, you can apply transformations to selected nodes.

By contrast, if you middle-click to select a branch, you can only deselect the whole branch and not individual elements.

1. Choose the **Multi** menu cell.
2. Choose the **Select → Multi-Node** command.
3. Select a branch or tree.
4. Click any branch or node in the hierarchy to deselect it.

Cutting Links in a Hierarchy

It is often desirable to cut the hierarchical links between objects. For example, if you have been modelling an envelope for a skeleton using the Automatic envelope type for experimentation purposes, you will eventually detach the envelope from its skeleton in order to attach your real envelope model.

The **Cut** menu cell in the Model and Actor modules allows you to cut the link between a parent and its child or children in a hierarchy of objects.

If the child is also a parent, the links to its own children are not affected.

1. Select the object you want to detach from its parent.
2. Choose the **Cut** menu cell.

You can also use **Cut** in conjunction with the **Multi** menu cell to cut a branch of a hierarchy or the whole hierarchy's link:

1. Select the branch or hierarchy you want to cut.
2. Choose the **Multi** menu cell.
3. Choose the **Cut** menu cell.

Manipulating Hierarchies

By moving the parent (tree) of the hierarchy, all of the branches and the nodes move with it. If you move a single branch, its nodes move accordingly. If you move a single node, only that node moves.

In SOFTIMAGE 3D, the nature and scope of the transformation effect depends on what selection mode you use: that is, whether you select the node, branch, or tree before the transformation operation.

- Left mouse button selects only the node.
- Middle mouse button selects the branch.
- Right mouse button selects the tree (the entire hierarchy).

It also depends on two commands in the **Preferences** menu: **Hierarchy Compensation** and **Hierarchy Manipulation**.

Positioning of Objects

The **Preferences** ➔ **Hierarchy Compensation** command affects the relative positioning of objects when you link them in a hierarchy.

When hierarchy compensation is selected (the default mode) and you parent one object to another, the child remains at the location and the size at the moment when the link was created.

When hierarchy compensation is deselected, the child inherits the transformation parameters of the parent. As a result, the child adds its translation coordinates to those of the parent and its position and size change to reflect this.

Maintaining Positions of Objects

The **Preferences** ➔ **Hierarchy Manipulation** command maintains the position of children in relation to their parent as the parent is transformed. In this case, it does not matter whether you select the parent in node, branch, or tree selection mode.

For practice, make sure this command is selected, and then select the head of the hierarchy in *node* selection mode and translate it. You will see that the whole hierarchy moves as one.

If this command is not active, to have the same effect you would have to use either the branch or tree selection mode.

For purposes of the following example, make sure that both **Preference** → **Hierarchy Manipulation** and **Hierarchy Compensation** commands are active.

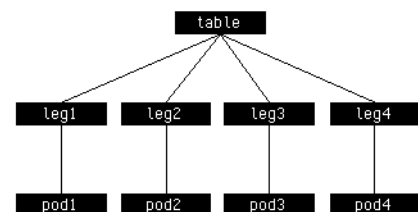
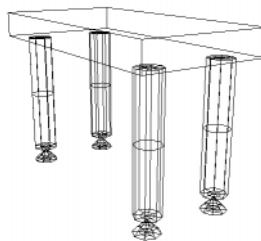
Example

To create a simple hierarchy such as a table, follow these steps:

1. Model the pieces needed: four legs and the tabletop (pods are optional).
2. Put all the pieces together in their correct positions.
3. Decide which piece will be the parent:

Remember that all other pieces move according to the parent, so you should choose a piece that makes manipulations easy. Choosing a leg as the parent would not be very practical, but the tabletop would be because its centre is in the centre of the table. In general, it is easier to manipulate objects when the object's centre is in the middle of the object itself.

4. To create the hierarchy, select the tabletop to be the parent.
5. Choose the **Parent** menu cell and follow the mouse button instructions in the status bar to define all the children by picking each object. The parent (tabletop) and the children remain highlighted as you add more children.
6. Don't forget to end the parenting mode by right-clicking or pressing the Esc key.
7. The pieces remain highlighted, and you can move the table as a unit.



Creating a Hierarchy for Articulation

In the animation of a hierarchy with multiple moving parts, it is important to know how you want each part to move.

The location of the centre of an object is very important because this determines how the object rotates and scales. When rotating an object, the centre acts as the axis or hinge; that is, if your model is a door you want to open and close, you would place its centre on the edge of the object. When scaling an object, the centre acts like the object's base. The object is scaled down to it or up from it; for example, a bouncing ball that squashes and stretches would have its centre on the bottom of the object.

In more complex hierarchies, the principles are the same. If you are building an arm, you would place the centre of each object at the joint where two objects meet.



Another way to quickly build this kind of hierarchical structure for animation is to create chains. For more information, see *Drawing Articulated Chains* on page 231.

Example

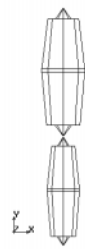
Make an arm and hand using cylinders:

1. Choose the **Get** → **Primitive** → **Cylinder** command to create the upper arm. Model the cylinder as desired.
2. Duplicate the cylinder to create the forearm. Translate it so that it lines up edge to edge with the upper arm. The forearm's centre is located at its midpoint by default.

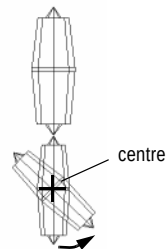
Rotate the forearm and you will see it pivot around this point. To have the forearm rotate about the point where it meets the upper arm (the elbow joint), you will reposition its centre.

3. Select the CTR manipulation mode in the status bar. The forearm's centre is displayed (red, green, and blue local axis). Rotation is now applied to the object's centre, not the object itself.

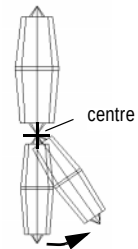
4. Translate the forearm centre to the elbow joint.



Model
Arm and forearm



Default centre
The forearm pivots around its midpoint where the centre is located by default.



Move the centre
Translate the centre to the top of the forearm where it meets the upper arm. The forearm pivots around the elbow.

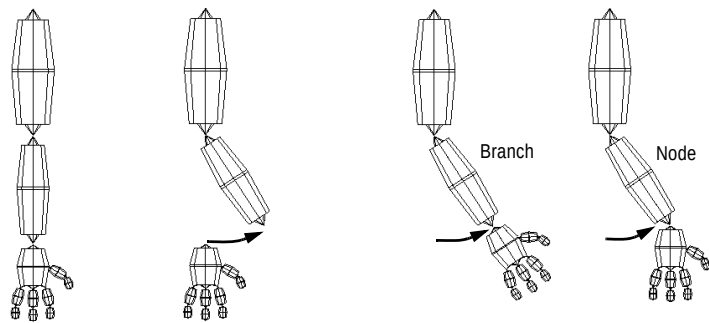
5. Return to object mode by selecting the OBJ manipulation mode in the status bar, and try rotating the forearm. It should now pivot around the centre at the joint.
6. Select the upper arm, and in centre (CTR) mode, translate the centre to the top where a shoulder joint should be. Return to object (OBJ) mode.
7. Select the forearm and duplicate it to create the hand.

The hand's centre is located at the top because it was created as a duplicate of the forearm.

8. Create fingers by duplicating existing cylinders. Scale and translate the hand and fingers as desired.

At this point, each object of your model will rotate about its own joint, but it will do so independently of the other objects. Rotate the

forearm and it will leave the fingers behind.

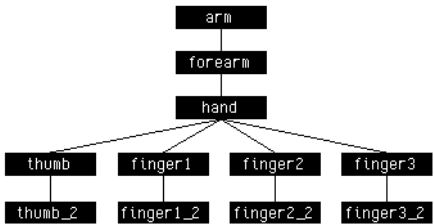


Model

Independent objects
Rotate the forearm and it leaves the hand and fingers behind.

Objects in a hierarchy
Select the forearm as a branch or node and the hand moves with it.

If you want the hand and fingers to move with the forearm, you can create a hierarchy with the forearm as the parent of the hand, and the hand as the parent of the fingers. The upper arm is at the top of the tree, as a parent of the forearm.



Note Before you create the hierarchy, make sure that the objects are positioned the way you would like.

In this hierarchy, you can select and rotate a branch and all of its elements rotate about a common point, namely, the centre of their parent. For example, if you select the forearm, all of its children (hand and fingers) rotate with it, pivoting around the elbow joint.

Remember, if the **Preferences → Hierarchy Manipulation** command is selected, every object in a branch follows the movement of its parent even if you select the parent as a node. Otherwise, use branch mode selection.

C H A P T E R E L E V E N

Deforming an Object's Shape

Introduction

The *Basic Modelling Tasks* and other chapters have described how to move points on objects using standard commands in the **Edit**, **Tag**, **Polygon**, and **Effect** menus.

This chapter discusses quick ways of editing the shape of an object using more global, automated methods. Typically, these methods act upon the object indirectly, using one object to affect the shape of another object. Such techniques are grouped under the title of *deformation*.

You can deform an object by the following means:

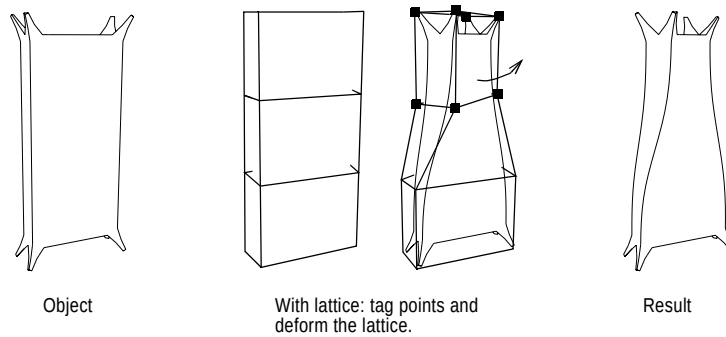
- Create a lattice structure around the object and transform the points of the lattice to deform the object (see page 294).
- Deform one object along the surface of another or on a curve (see page 298).
- Tag groups of points and save them as a cluster, and apply transformations to the cluster (see page 308).
- Save an object's deformation at different stages to create a library of different shapes that you can model with (see page 311).
- Assign the object as an envelope of a skeleton. As such, its points are weighted to the nearest skeleton elements and deform with its shape and movements (see page 312).
- Apply the cartoon-like Quick Stretch effect (see page 314).
- Use various custom effects (see page 306).
- Apply a quick and random displacement to points using the **Effect** → **Randomize** command, described on page 45.

Another deformation technique, called Waves, can be used for modelling purposed, but is essentially an animation effect. It is described in the *Deformations* chapter in the *Animating User's Guide*.

Typically, you can animate these deformations and combine their effects to create complex animations.

Deforming with Lattices

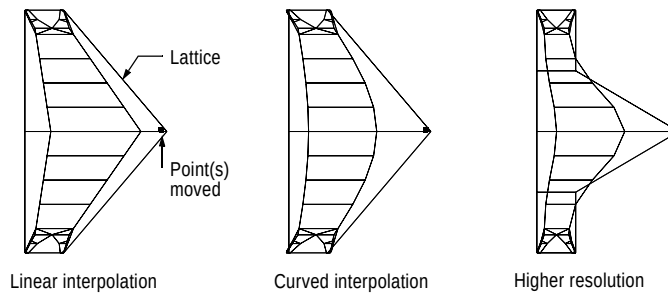
Lattices allow you to deform an object or a group of objects linked in a hierarchy. A lattice is a bounding box with a variable resolution. Every subdivision of the lattice is associated with a number of points on the object. By moving the lattice points, you can deform the object in the same way.



The effect of transformations on the object is determined by the lattice's resolution and interpolation.

- The lower the resolution, the more global the effect. For example, a global effect is quick to model by using a lattice of few points. A higher resolution creates more control points on the lattice, so each lattice point controls a small area, giving a more local effect.
- The interpolation type used in the lattice: linear or curve.

The interpolation type you choose can give quite a different result.



Using Lattices for Modelling

You can use lattices for modelling purposes: for example, if you wanted to model the head of a pitchfork, you draw the head with its tines in the Top window and extrude it, then use a lattice to easily apply the “bend.”

Most often, your objective is to animate the deformations, either by keyframing the deformations on the lattice, or by keyframing the movements of objects through the lattice (they are deformed as they move through the lattice). See *Creating a Lattice* on page 296 for a simple procedure. For more information on lattices from an animation point of view, see *Animating a Lattice* on page 101 of the *Animating User’s Guide*.

Lattices are good for deforming the object overall, but if you wanted to do some detailed localized work, you could use clusters instead (see page 308).

For example, you could use a lattice to model and animate a soft drink bottle doing a hula dance, but to model and animate a smile, you’d be better off using a cluster.

Types of Lattices

You can apply a lattice to a single object (Node lattice) or an entire hierarchy of objects (Branch lattice), in which case all objects in the hierarchy are deformed by the lattice.

Using Branch Lattices Creatively

A useful approach to using a branch lattice on a hierarchy is this:

A lattice forms a bounding box around the associated object. If you wanted to make a sphere deform and behave like a glob of liquid, this would be a somewhat lengthy process to do with a Node lattice on the sphere. You would have to create several deformations to the Node lattice of the sphere and animate them.

A simpler, quicker approach is to place the sphere inside a larger object and create a hierarchy. If you apply a Branch lattice to the hierarchy, the lattice surrounds the entire hierarchy. Now you can use lattice deformation on the larger object to deform the sphere. This technique is used essentially for animating a deformation.

Creating a Lattice

1. Select an object or a branch of a hierarchy.
2. Choose the **Lattice** → **Node** → **Create** command in the Model module if an object is selected, or the **Lattice** → **Branch** → **Create** command in the Model module if a hierarchy is selected. The Create Lattice dialogue box is displayed.
3. For each axis (x, y, and z), you can set the **Subdivisions** (resolution). Low resolution allows a more global, “crude” deformation effect. High resolution allows more localized deformations of the latticed object.

You cannot change the resolution of a lattice after it is created, so make sure the resolution is correct when you create it.

4. Select the **Interpolation** type to be either Linear or Curve. Curve interpolation lets you achieve a smoother deformation.
5. When you create the lattice, it is highlighted in yellow and surrounds the object you selected. In the Schematic window, NL and BL indicate node and branch lattices.

Deforming with Lattices

After you have created a lattice, you can move the points on it as you would for any other object.

1. Create a node or branch lattice.
2. Tag points on the lattice using **t** Supra key.
3. In TAG mode, transform the tagged lattice points using the **ScaleXYZ**, **RotXYZ**, or **TransXYZ** menu cells. You can also choose **Edit** → **Move Point** to move individual points on lattices.

The object deforms accordingly to the deformation of the lattice points.

For information and procedures for animating a lattice's deformation, see *Animating a Lattice* on page 101 of the *Animating User's Guide*.

Freezing a Lattice

When you finished modelling the object or branch with the lattice, you can save its current shape using the **Lattice → Node/Branch → Freeze** commands in the Model module, then discard the lattice. This makes a lattice an effective modelling tool.

1. Select an object or branch of a hierarchy with a lattice.
2. Choose the **Lattice → Node → Freeze** command if an object is selected or the **Lattice → Branch → Freeze** command if a hierarchy is selected.

The deformed object or branch of a hierarchy is saved in its current shape and the lattice is immediately discarded.

Creating Local Lattices with Custom Effects

To create a lattice to be used for local deformations, you can use the **Lattice → MakeLattice** command in the Model module. You can position the lattice anywhere on the object being deformed, and apply the effect only to the area within the lattice bounding box. The deformation it produces, as well as the position of the lattice on the object being deformed, can be animated.

The position of the lattice is determined by its scale, rotation, and translation values, and like the SOFTIMAGE 3D lattice, the deformation produced is defined by editing its points.

To apply the lattice to the object, you must use the **LocalLattice** or **BranchLattice** effect.

For more information on using this effect, see its command description in the *Reference Guide*.

Deforming by Curves and Surfaces

You can associate objects in such a way that one is deformed according to the shape of the other one. The object you wish to deform can be anything except a meta-clay system because meta-clay elements have no geometry, which is the basis for deformation.

The deforming tool can be any type of curve, or a patch or NURBS surface (not polygon mesh objects).

You can deform a single object, a whole hierarchy of objects, or selected objects in a hierarchy. For example, model a pie and make it deform on the face of the character – splat! – and then run down his face.

You might also have a hierarchy consisting of a candlestick object and drops of wax formed to its surface. Most of the hierarchy would be static, but some drops might drip down the candle, deforming to the surface as they run down.

Layering of the effects of deformations can give some interesting results, particularly when the results are animated. For examples of animated deformations, see *Animating Curve and Surface Deformation* on page 105 of the *Animating User's Guide*.

The Deformation World

Deformation defines a new coordinate system which is not regular and linear like the global, or “normal,” world coordinate system. As a result, deformation depends on two factors:

- The shape of the deforming object.
- The scaling, rotation, and translation of the deformed object in the “deformation world.”

Setting Up for Deformation

You can take steps to prevent unexpected offsets or wrong orientation during deformation, and make it easier to scale and rotate the deformed object.

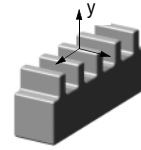
Ensuring Correct Orientation

Before deforming, check that the object you want to deform is correctly oriented so that it moves as you want it to. For example, do you want the object to move along “face first,” lying on its back, or

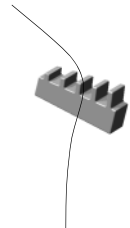
“feet first”? To deform by a surface, the surface object’s xz plane represents the uv directions for the object. For a curve, it is considered to be the y-axis and lines up the object’s axes accordingly.



Orient the object's centre
Object's y-axis points to direction of deformation.



Compare with different orientation
Object's y-axis is oriented along direction of comb, so the result is different.



Ensuring Correct Positioning

When you apply a deformation, you probably want your object to be placed on the curve or surface, or some predictable spot in space.

To ensure that the deformed object has a reasonable positioning:

1. Before deforming, move the soon-to-be-deformed object to the centre of the world (origin).
2. Choose **Effect → Freeze → Transformations**.

This freezes the object’s transformations and places the object’s centre at the origin.

After you apply deformation, the object may look quite different from its normal shape. To recover the object’s normal proportions:

- If you are using curve deformation, reset the object’s transformations, by setting all scaling factors to 1, and rotation and translation values to 0.
- If you are using surface deformation, you may need to scale down the object to recover its normal proportions.

The position of the edge flag on the deforming curve also affects the position of the deformed model relative to the curve. To show this effect when you deform an object along a curve of surface, select the curve and choose **Effect → Curve Controls**. If you shift the starting point of the curve, or if you inverse the direction in u, you see the effects on the deformed object.

This is also true when deforming objects on surfaces.

Deforming an Object along a Curve

You can associate an object to a curve using the **Deformation → By Curve** command in the Model module. The object is deformed according to the shape of the selected curve and controlled by deformation transformation.

The points of the object deformed along the curve tend to converge on the curve side and expand on the side farthest from the curve, like an accordion.

Translating it in x or z offsets it from the curve, but the same squeezing and expansion of points occurs.

Applying a Deformation

1. Select the object you want to deform.
2. Choose the **Deformation → By Curve → Node → Create** command for a single object or the **Deformation → By Curve → Branch → Create** command for objects in a hierarchy.
3. Select the deforming curve.

If the resolution of the deformed object is insufficient, it may not deform smoothly along the curve. In this case:

4. Deactivate the deformation using the **Deformation → By Curve → Node/Branch → Deactivate** command.
5. Increase the object's resolution.
6. Reapply the deformation.
7. Translating the object in y makes it move along the curve, and deform with its twists and turns. To offset the deformed object away from the curve, translate it in x or z.

Example

In this example, you deform text along a curve.

1. Create text using the **Get → Text** command.
2. Get a B-Spline circle primitive with a radius of 10.
3. Select the text hierarchy, and orient it: rotate by 90 in x, 180 in y, and 90 in z.

4. Translate it so that an equal portion is above and below the origin (xz plane). Use the f Supra key to frame selected objects in the window.
5. Choose **Effect → Freeze → Transformations** to move the centre of the hierarchy to origin.
6. Select the text and choose **Deformation → By Curve → Branch → Create**. Pick the circle as the curve by which to deform the text.

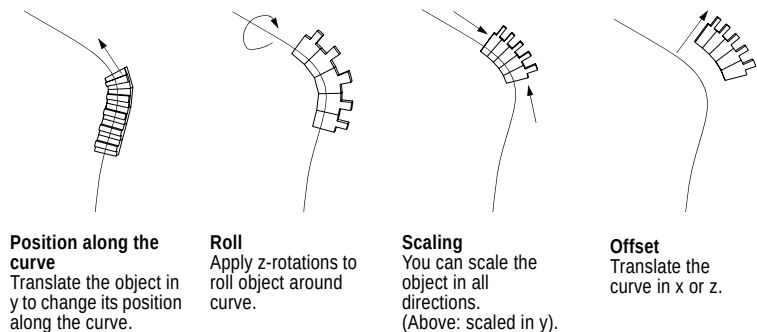
Et voila! The text deforms to the curve.

7. Experiment with where you want the text by translating it in z.

Transforming the Deformed Object

The curve is considered to be the y-axis and the two other axes are perpendicular to the curve.

To move the object along the curve, translate the object in y.



Scaling uses the same laws: for example, scaling in y scales the points along the curve. The points converge or expand depending on the position from the curve, as described above. Scaling is done from the centre of the object, which is why it is important to set it properly before the deformation.

Rotation (called “Roll” in the **SaveKey → Object → Node/Branch Curve Deformation** command in the Motion module) is limited to the z-axis. The points rotate around the curve, deforming in accordance with the same principles described above.



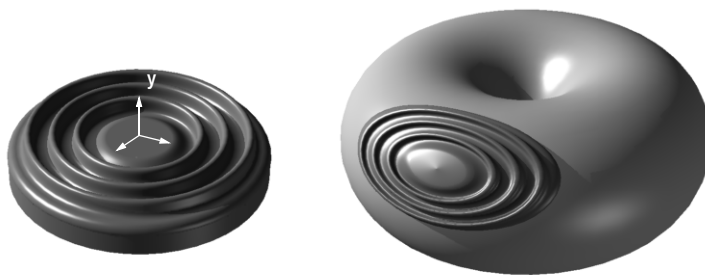
Remember you can also change the position of the deformed object relative to the deforming curve, by selecting the curve and inverting its direction or shifting its starting point in u (**Effect → Curve Controls**).

You can create, deactivate, reset, freeze, or remove a deformation relationship using the subcommands in the **Deformation** menu. See the tasks for these subcommands starting on page 305.

Deforming an Object along a Surface

You can associate objects to a surface using the **Deformation → By Surface → Node/Branch** commands in the Model module.

The deformation process using a surface is similar to deformation along a curve. The main difference is that a patch or NURBS surface acts as a deforming plane.



Object before deformation
Y-axis will be perpendicular to surface. X and z axes will map to u and v of the surface.

Object after deformation on surface
Object spreads and twists to conform to the surface (a NURBS torus in example above).

You can deform a single object or a hierarchy of objects, or selected objects within a hierarchy.

For example, you might have a hierarchy consisting of a bumpy candle and drops of wax dripping along the surface. Most of the hierarchy would be static – the candle surface itself and the hardened wax drops – but some drops are dripping down the candle, deforming to the candle's bumpy surface.

Setting Up for Deformation

For the same reasons described in *Setting Up for Deformation* on page 298, you should do the following to prevent any unexpected offsets of the deformed object:

- Move the soon-to-be-deformed object to the global origin.
- Freeze it using the **Effect → Freeze → Transformations** command.



If your model “explodes” after applying the deformation, try scaling it down to the desired size before manipulating it.

As in curve deformation, the position of the edge flags on the deforming surface affects the position of the deformed model relative to the surface. To show this effect when you deform an object along a surface, select the surface and choose **Effect → Curve Controls**. By shifting the starting point of the surface or if you inverse the direction in u or v, you see the effects on the deformed object.

Deforming an Object

The procedures for deforming an object along a surface are the same as those for deforming along a curve, but use the **Deformation → By Surface → Node/Branch** commands instead.

You can create, deactivate, reset, freeze or remove a deformation relationship using the subcommands in the **Deformation** menu cell following the same procedures as for curve deformation, but substituting “surface” for “curve.”

For more information, see *Deforming along a Surface* on page 106 of the *Animating User’s Guide*, which discusses how to animate the deformation transformations.

Transforming the Deformed Object

The deforming surface is considered the xz plane of the deformation world. Transformations deform the object in the same way as curve deformation (described previously), except that the convergence or expansion of points is mapped in two directions (u and v) instead of one (u), as in the case of curves.

To move the object perpendicularly to the deforming surface (xz plane), translate the object on y.

Translation on x and z move and deform the object according to the shape of the deforming surface.

Note

Translating too far beyond the deforming surface boundaries can cause the model to deform excessively.

Scaling originates from the object’s centre and can be performed on all three axes. Rotation on the three axes is also possible.



You can also change the position of the deformed object relative to the deforming surface using **Effect → Curve Controls**, as described in *Setting Up for Deformation* on page 298.

Example for Surface Deformation

1. Create the deforming object: get a patch or NURBS grid using **Get → Primitive → Grid**.
2. To make the surface bumpy, choose the **Effect → Randomize** command and apply the default values for x, y, and z.
3. Tag some of the points and in TAG mode, deform the surface some more (choose **Edit → Move Point** and translate the selected points).
4. Create the object you want to deform. Choose **Get → Primitive → Grid** using the same settings as before.
5. Give this object a material to distinguish it from the deforming object. Choose the **Material** menu cell in the Matter module and make a purple colour.
6. Make sure the soon-to-be-deformed object is selected.
7. In the Model module, choose **Deformation → by Surface → Node → Create** and pick the deforming surface (the first grid you created).

The purple grid is deformed to the surface of the bumpy grid.

8. Hide the deforming grid by choosing **Display → Hide → Toggle & Desel. Hidden**.
9. To see the effect, translate the purple grid over the deforming surface.
10. When you have achieved the deformation result you want, choose the **Deformation → by Surface → Node → Freeze** to break the link to deforming surface. The current shape of the purple grid is preserved and you can now transform it according to the “normal” world transformations.

For additional steps to animate this example, see page 106 of the *Animating User's Guide*.

Deactivating Deformations

You can temporarily suspend deformation of the object, such as when you need direct control over the object's transformation or when you want to manipulate the object's centre or points.

1. Select the object you have deformed.
2. Choose the **Deformation → By Curve/By Surface → Node/Branch → Deactivate** command.
3. To reactivate deformation, choose the **Deformation → By Curve/By Surface → Node/Branch → Create** command.

Resetting Deformations

You can reset all deformation transformations to the original values set when the deformation was created.

1. Select the object you have deformed.
2. Choose the **Deformation → By Curve/By Surface → Node/Branch → Reset** command.

Freezing Deformations

When you have achieved the deformation result you want, you can save its current deformed shape and break the link to the associated deformation curve or surface by choosing the **Deformation → By Curve/By Surface → Node/Branch → Freeze** command.

1. Select the object you have deformed.
2. Choose the **Deformation → By Curve/By Surface → Node/Branch → Freeze** command.

The object is saved in its deformed state. At this point, you can transform the object according to the “normal” world, as opposed to the deformation world.

Removing Deformations

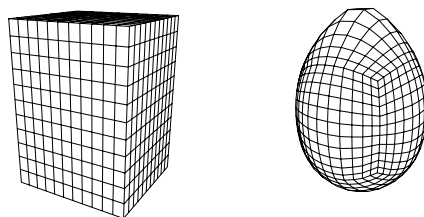
You can remove the object from the associated deformation curve or surface and return it to its original shape.

Select the object you have deformed and choose the **Deformation → By Curve/By Surface → Node/Branch → Remove** command.

Custom Effects for Deforming

Shrinkwrapping an Object to a Surface

The **Deformation** → **ShrinkWrap** and **HrcShrinkWrap** commands in the Model module allow you to “shrinkwrap” a model (curve, polygon mesh, patch, or NURBS surface) or hierarchy to the surface of a polygon mesh object. You can control the direction of the shrinking, as well as the amount of shrinkage. You can apply the effect to the entire model, or only to tagged points.



ShrinkWrap is typically used to:

- Create a metamorphosis between two dissimilar objects by shrinkwrapping another object (such as a large patch or NURBS sphere or a subdivided cube) to the two objects and then applying Shape interpolation on the results.
- Project a curve path onto a surface, such as an uneven terrain or a model of a character. This lets you easily create paths that are guaranteed to lie on the surface.
- Create seamless models. A large patch or NURBS surface can be shrunk to a model composed of several objects, such as a head modelled from several scaled and deformed spheres. This creates a simpler model consisting of a single patch or NURBS surface without seams.

For more information on using this effect, see its command description in the *Reference Guide*.

Using another Object's Geometry

The **Duplicate** → **CopyGeometry** command in the Model module allows you to substitute the geometry of one object with that of another. This allows you to quickly perform animations and deformations on very simple objects. When satisfied with those results, you can substitute the geometry of a more complex object.

The **Effects → Magnet** command in the Motion module allows you to deform the points of one model by using another model as a magnet. You can control which points of the object are moved, how much they are moved, and in which direction. All elements, including the shape and location of the magnet and the magnitude and extent of the force, can be animated. By freezing the magnetic effect, you can use Magnet as a modelling tool.

These are typical effects that can be created using Magnet:

- An object moving under a rug forcing the rug to deform according to its shape.
- A footprint can be easily impressed into a grid or other surface. The edges are automatically softened by the magnetic effect on nearby points.
- A muscle that is animated with an inverse kinematic chain may be made to bulge realistically by adding magnets to the chain's envelope.

The **Effects → Push** command in the Model module allows you to move points on a model in the direction of the model's surface normal. You can specify how far to move the points and the software calculates the rest using the vertex normals in the case of polygon meshes, and computing the surface normals in the case of patches, NURBS surfaces, and curves.

For more information on using these effects, see their command descriptions in the *Reference Guide*.

Deforming a Surface by Textures

The **Deformation → BumpMap** command in the Model module allows you to deform the surface of a model according to a picture file. BumpMap works on the same principle as bump mapping in 2D textures except that the control points of an object are actually offset according to the intensity of the image channels (similar to displacement mapping of 2D textures using the mental ray renderer). You have independent control over each of the red, green, blue, and alpha channels. Filtering parameters ensure that the result does not show aliasing artifacts.

For more information on using this effect, see its command description in the *Reference Guide*. Also, see the *Defining Materials and Textures Guide* for information on 2D texture mapping.

Deforming with Clusters of Points

What's a Cluster?

Clusters are groups of tagged points that you have saved with a group name. You can create and name different clusters on the same object; later, you can select a specific cluster from the object's Cluster List and model with it. This avoids the tedium of tagging the same group of points before modelling them.

In comparison with lattice deformation, which is good for overall object deformation, clusters allows you to do more finely grained modelling work. For example, you could use lattice deformation on a soft drink bottle to make it do a hula dance, but use cluster deformation to model the bottle's mouth into a smile.

Cluster deformations can be animated. For examples and procedures for animating clusters, see *Shape and Cluster Animation* on page 91 of the *Animating User's Guide*.

Cluster Centres

When you create a cluster, you can give it its own centre, which is a null object by default. It is a separate object, and is created as the child of the object which has the cluster. In the Schematic window, compare this with an object's cluster that has no centre.

Since cluster centres are objects, they can be selected, parented, animated, cut, deleted, and constrained to other objects as you wish. Clusters centres are also useful for constraints: for example, you can constrain an object to a cluster centre or vice-versa. For information, see *Constraining between Clusters and Objects* on page 182 of the *Animating User's Guide*.

Creating a Cluster

1. Select an object and tag a group of points, using the **t Supra** key. Points can belong to more than one cluster group.
2. Choose the **Tag → Set Cluster** command in the Model module. The Set Cluster dialogue box is displayed.
3. Enter the cluster's name in the **Name** text box.
4. If you want the cluster to have its own centre, for purposes of deformation or animation, select the **Create Center** option.

5. Click **New** to create the cluster.

The name of the new cluster is added to the Cluster List.

6. Click **Ok** to accept the settings or **Cancel** to exit without creating the cluster.

Adding and Deleting Points from a Cluster

1. Select the object with the cluster.
2. To choose the cluster you wish to update, choose the **Tag → Cluster List** command in the Model module.
3. In the list, select the cluster. Its tagged points are highlighted on the object.
4. Click **Ok** to exit.
5. In any window, tag or untag points as necessary to update the points included in the cluster.
6. Choose **Tag → Set Cluster**. The Set Cluster dialogue box appears.
7. Click **Replace**. The cluster is updated, but the cluster name remains the same.

Deforming with a Cluster

Transforming a cluster with a centre has a different effect than transforming a cluster with no centre. When you select a cluster centre, the points in the cluster are scaled, rotated, and translated in relation to the cluster centre rather than object's centre.

Note

Cluster centres are distinct objects from the rest of the object. If an object has a cluster centre child and you select the parent object (in node selection), transformations are applied to points outside the cluster and the cluster itself is not affected.

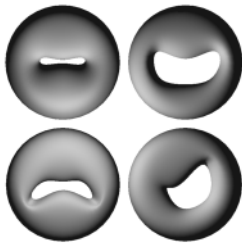
1. If there is a cluster centre, select it, scale, rotate, or translate it in OBJ mode. The points in the cluster are transformed in relation to the cluster centre.
2. If the cluster has no centre, select the object which contains the cluster. In TAG mode, scale, rotate, or translate it.

In this case, the cluster points are transformed in relation to the object's centre.

If tagged points belong to more than one cluster, the transformations are averaged. For example, if a point belongs to three clusters with cluster centres, and you move any of the clusters or centres, the resulting position of the point is an average of the transformations it receives from the three clusters.

Shape Deformation

You can tag and move points on an object to deform it, then save an image of the model, called a *shape*, using the **Shape → Select Key Shape** or **SaveKey → Object → Shape** command in the Motion module.



A library of shapes

You can create a Shape List for the same object, and animate these changes. For information and procedures, see *Shape and Cluster Animation* on page 91 of the *Animating User's Guide*.

From a modelling perspective, shape deformation is useful for the library of shapes that you can create for the same object, any one of which can be selected and used if necessary. For example, if you saved a library of shapes which interpolated from an ordinary car to a fire-breathing hot rod, you could later select any of the in-between shapes and use it for modelling.

Shape and Cluster Deformation

Shape deformation saves the entire geometrical structure (topology) of an object, compared with cluster deformation which only saves information about the clusters on the object. As a result, you can achieve similar animation results using cluster deformation, while storing less information. On the other hand, shape deformation provides you with the library of different versions of the whole model.

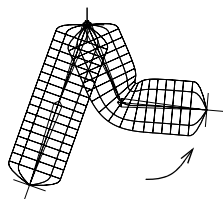
You can animate such things as a face using both shape and cluster animation. Larger facial parts such as the head, eyebrows, and mouth could be shape animated, while using cluster animation to alter a smaller part of the object, such as changing the position of the cheekbones in response to frowning or smiling.

Deforming Envelopes

An *envelope* is any model or hierarchy that you assign to cover a skeleton or a part of a skeleton. Once assigned, it moves with the skeleton, and (in most cases), is deformed by the skeleton.

Ways of Tweaking Envelope Deformation

You can use any type of model for an envelope. The manner and extent to which the envelope is deformed by the skeleton depends on the type of envelope assignment and by some tweaking techniques available.



Flexible envelope (tube) is deformed by the chain.

- You can designate the model as a flexible envelope, whose vertices deform with the skeleton movements. A rigid envelope does not.
- You can create bounding models, to prevent or enforce the assignment of object vertices to specified parts of the skeleton.
- Assignment of envelope vertices to skeleton elements is handled automatically (when you initially assign the envelope), but you can reassign vertices or change their relative weighting towards parts of the skeleton, to prevent deformation oddities, particularly around points of articulation (bulging kneecaps). Alternately, you might want to create odd and exaggerated deformations using the same technique!
- You can write expressions to relate the transformation of one part of the model to those of another. For example, create an expression to relate the scaling of certain parts of the envelope to the rotation of a joint, to make a bicep bulge or stretch when the elbow is bent.

For more information and procedures for assigning envelopes to skeletons, see *Envelopes* on page 243.

Using Objects in Skeletons

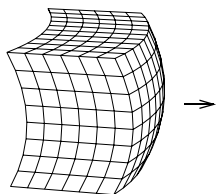
Using objects in a skeleton framework offers greater control of local deformation. It lets biceps ripple under the skin, or lets a cartoon character's knee "throb" up and down in an exaggerated fashion independently of the forward motion.

When you use objects as skeleton elements and assign an envelope, you can take advantage of the way envelope assignment and weighting allows deformation over an area with subtler, diminishing effects.

A skeleton of an arm could be composed of a chain and a sphere, scaled to the shape of a bicep muscle, positioned, and then constrained to `joint_a` of the chain. To animate the deformation, a simple expression relates the scaling of the sphere (bicep) with the rotation of the forearm joint: as the arm flexes, the sphere is scaled to make a swelling bicep under the skin.

You could also have a face with a null object at the cheek level. Some envelope vertices are automatically assigned to the null. To give the character a nervous twitch without affecting the whole face, you could apply an expression using a random time-related function to transform the null.

For more information and procedures for deforming envelopes with objects in skeletons, see *Using Objects in Skeletons* on page 238.



Cube flexing with motion.

Deforming with Quick Stretch

Using Quick Stretch, objects are deformed interactively by their own movements, velocity, or acceleration. To use it, create an object with a fairly high number of subdivisions so that you will notice the deformation.

How does Quick Stretch relate to modelling? You can save object shapes as they are deformed by Quick Stretch and then model these saved shapes:

- Once you have a shape you like, you could simply duplicate it and model with the resulting object.
- For greater flexibility and choice among shapes, animate the Quick Stretch effect using keyframing, as described in detail in *Quick Stretch* on page 109 of the *Animating User's Guide*.
- You can then go frame by frame and choose from any of the interpolated shapes of the object, duplicate them, and model them from that point.

Types of Quick Stretch Effects

The basic types of Quick Stretch deformation are Flex, Stretch, and Yield.

Flexing

A flexible object when moving rapidly in one direction appears to bend or flex in the direction of the motion, due to the resistance of the air (or water).

Stretching

Stretching lets you create the classic cartoon “squash and stretch” effect, where the object elongates in the direction of the motion, and becomes thinner in the other directions.

Yielding

Depending on the mass assigned, a moving object might appear to bulge due to the internal displacement of its mass.

Origin of Deformation

The origin of the deformation depends on the type of force affecting the object. For example, if a building faces such a strong wind that it

bends a little, the deformation should originate from the ground up. In all cases, the forces that make the object move are applied somewhere on the object, and this affects how a flexible object deforms.

Creating a Quick Stretch Effect

To use Quick Stretch, create an object (with a good number of subdivisions), then follow these few steps:

1. Choose the **QStretch** ➔ **Setup** command in the Actor module.
2. Set the parameters (at minimum, select Flex, Stretch, or Yield) and click Ok.
3. Move the object around, and watch it squash, stretch, flex, or yield like all your favourite cartoon characters.

The origin of the deformation is the centre of the object. Two control icons are displayed at the centre and cannot be disassociated from it.

Note

This kind of interactive deformation is not possible if you have explicitly plotted velocities and acceleration.

C H A P T E R T W E L V E

Optimizing Models

Overview of Optimizing Models

Heavy, complex models can take a long time to render. SOFTIMAGE 3D provides several tools for lightening models and reducing rendering times.

Optimizing models is useful in a variety of situations. For example, games and real-time simulators require fast rendering. In another situation, you might just want to speed up rendering somewhat without losing any noticeable detail. You can lighten distant objects by a large amount, and leave more detail on objects that are near the camera.

Because the goals of optimization vary from case to case, the tools described in this chapter require a great deal of your input. You should experiment with different settings until you achieve the desired result.

This chapter describes how to optimize models using the following effects that are found in the Model module:

- The Polygon Reduction effect can reduce the number of polygons in a model using several methods. A variety of options let you control the amount of reduction versus the amount of detail in the reduced model. For more information, see *Reducing the Number of Polygons* on page 321.
- The Web effect can convert a detailed, intricate model to a simple surface that represents only the external appearance of the original. For more information, see *Webbing Models* on page 354.
- The Create NURBS Net effect does not lighten models, but it generates QTM's (quadratic texture maps). These are a type of model required by the graphics hardware of certain games platforms, and this effect is usually applied after the model has been lightened. For more information, see *Generating Quadratic Texture Maps* on page 356.

Although both the Polygon Reduction and Web effects work only on polygon meshes, you can convert other types of models to polygon meshes in order to reduce them. The next section provides some tips for accomplishing this.

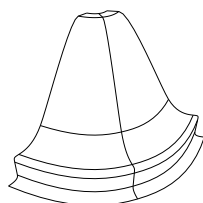
In addition to optimizing your models, you might want to apply colour reduction. This allows you to decrease the number of colours used by textures, resulting in faster rendering times and lower

memory requirements on games hardware. For more information about colour reduction, refer to *Reducing the Number of Palette Colours* on page 145 of the *Defining Materials and Textures User's Guide*.

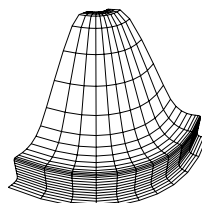
Converting Objects to Polygon Meshes

The Polygon reduction and Web effects work only on polygon mesh models. However, you can convert heavy patches, NURBS surfaces, and meta-clay elements to polygon meshes in order to lighten them.

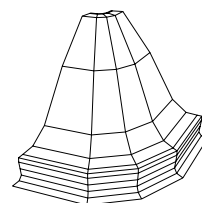
To convert a patch or NURBS surface, use the **Effect → Convert** command. Before you apply this command, you can control the number of polygons in the result by selecting the surface and choosing the **Info → Selection** command. Set the **Step** options to smaller values in order to generate fewer polygons.



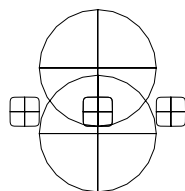
NURBS surface



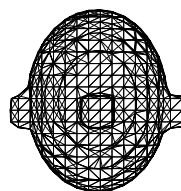
Converted to polygon mesh object using different step values of the NURBS surface.



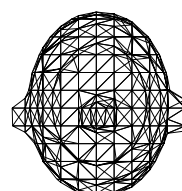
To convert a meta-clay element, use the **Meta-Clay → Convert to Polygon** command. Again, you can control the number of polygons in the result by first selecting the meta-clay system and choosing the **Info → Selection** command. Set the **Wireframe Slice Size** option to a larger value in order to generate fewer polygons.



Meta-clay system



Converted to polygon mesh object using different wireframe slice values.



Reducing the Number of Polygons

As the term *polygon reduction* suggests, the objective is to reduce the number of polygons in a model to a minimum, while still maintaining a useful degree of resemblance to the original. SOFTIMAGE 3D provides several methods for achieving this goal.

By reducing the number of polygons, you can greatly decrease the overhead associated with rendering. Fewer polygons mean faster rendering.

For this reason, polygon reduction can be used by anyone who must balance high performance with high fidelity to the original model. It is particularly useful for developing games and real-time simulators, where you must meet specific performance requirements.

Considerations for Using Polygon Reduction

With polygon reduction, there is always a compromise between model complexity and rendering speed: you have to decide on the relative importance of realistic graphics versus performance.

There may be several possible approaches to simplifying any given model, and the polygon reduction techniques are not simply a matter of uniformly reducing the polygon count. To decide what approach to take, ask yourself exactly what you can afford to simplify about the model.

For these reasons, assessing your model is an important first step in achieving satisfactory results. Once you have determined what you can afford to simplify, reduce, and eliminate, you can choose the most useful approach.

Comparing Reduction Types

There are three basic types of polygon reduction effects:

- The Rule Based effect
- The Optimization Based effect
- The Filter effect

Generally speaking, geometric models are best handled by the Rule Based effect. More organic models with smooth, flowing curves are best handled with the Optimization Based effect. You can also combine effects to achieve the results you want.

Rule Based

The Rule Based effect allows you to reduce the number of polygons in a model by eliminating edges according to specific rules. It can be applied using the **Effect → PolygonRuleBased** command or the **Rule Based** option of the **Effect → PolygonReduction** command in the Model module.

Because they use specific rules to eliminate edges, rule-based reductions are faster than optimization-based ones. In addition, they are particularly useful where rendering time is important because they can reduce a model to an exact polygon count. However, there is usually some cost to model fidelity.

Optimization Based

The Optimization Based effect allows you to reduce the number of polygons while preserving fidelity to the model. It can be applied using the **Effect → PolygonOptimization** command or the **Optimization** option of the **Effect → PolygonReduction** command in the Model module.

Optimization-based reductions generate vertices whose final placements may be different than those of the original model. This means that they have higher fidelity to the original model than most rule-based reductions.

However, optimization-based reductions are considerably slower than rule-based reductions since the choices involved in the comparison of a starting polygon mesh and point cloud are infinite. Because they can take so long, a set of options lets you query and stop an optimization during processing.

To achieve a balance between fidelity and reduction, optimization-based reductions use the reduction target as a rough guideline only. For example, you could specify a polygon count of 300, but the resulting model may have more or less polygons than that.

Filter

The Filter effect cleans up models using a variety of methods. This is particularly useful before or after a polygon reduction process. It can be applied using the **Effect → PolygonFilter** command or the **Filter** option of the **Effect → PolygonReduction** command in the Model module.

The Filter effect does not apply any reduction algorithms, but some of the options may result in a reduced model. For more information about using the Filter effect, refer to the description of **Effect → PolygonReduction** on page 478 of the *Reference Guide*.

Combining Reductions

You can apply reductions additively or non-additively. In non-additive reductions, the reduction is applied to the previous version of the model; this allows you to experiment with different settings on a trial-and-error basis. In contrast, additive reductions are cumulative; the reduced model is used as the starting point for the next reduction.

Additive reductions let you achieve your goal by combining several reductions with different settings. For example, you might need to combine several rule-based reductions with different criteria to get the reduction you want.

You can also combine different reduction effects with the **Effect → PolygonReduction** command, which includes the parameters for the Rule Based, Optimization Based, and Filter effects in one dialogue box. For example, you can use the Filter effect to clean up the model after a rule-based reduction, or you might want to apply a rule-based reduction in order to speed up an optimization.

Overview of the Procedure

Applying the Polygon Reduction effect involves two procedures:

- Apply the effect the first time to create a cone icon representing the effect.
- Modify the effect by editing its parameters.

As you work with the effect, you will probably want to experiment with various settings. See *Using the Effect* on page 327 for more information on applying, undoing, and reapplying the effect. For information about preserving specific features or restricting the effect to defined areas, see *Controlling the Scope of the Effect* on page 329.

If you want some hands-on experience with using polygon reduction, see *Tutorial Examples* on page 346.

Before You Start

Before you begin to reduce your model, there are some issues you should consider.

Apply Polygon Reduction before Deformation Animation

You should apply polygon reduction before you define any deformation or shape animation for your model. This is because polygon reduction may remove and create vertices, invalidating any attached animation.

If you have already applied materials and textures to the model, you should make certain to select the **Preserve Materials** option when you apply the effect.

Study the Model

To find out how many polygons are in the model, use the **Info → Selection** command.

By studying the model before you apply any reduction, you can get a better idea of what you can reduce and how you can reduce it.

Consider these types of questions:

- How much reduction is required?
- Is it more important to reach an exact polygon count or simply to make the model lighter overall?
- Do you want to reduce flatter areas and keep detailed features, or do you want to eliminate noisy, highly detailed features such as grass or hair?
- What distinguishes the areas you want to reduce from the detail you want to keep? The lengths of edges? The areas of polygons? The angles between faces?

Tag Points

If there are features within the model that must remain exactly as they are, tag those points and apply a reduction on untagged points only. Alternatively, you can tag the areas to be reduced and apply the effect on tagged points only. For more information, see *Controlling the Scope of the Effect* on page 329.

Even if the detail that needs to be preserved has many polygons that could be reduced, you should probably still tag them and reduce

them separately. This separates the reduction of detailed areas from the overall reduction, giving you greater control over the effect.

For example, suppose you wanted to reduce a head, with large polygons making up the back of the head and many small polygons making up the highly detailed facial features. In this case, you should probably tag the vertices of the face and reduce them separately. This is because certain reduction criteria tend to reduce smaller things first: this would remove facial detail before reducing the back of the head.

Tag the points before you open the effect's dialogue box. If you tag points before you apply the effect the first time, the original set of tagged points are restored if you decide to return to the original model.

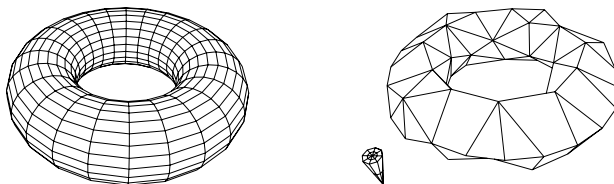
Applying the Effect the First Time

When you apply the effect to a model for the first time, it creates a cone icon that represents the effect. The cone icon is a polygon mesh object that stores information about the effect and its settings. You can select the icon in the same way you would select any other model, but it does not appear in a Shade view or rendered image.

To apply the effect for the first time and create the cone icon:

1. Choose a polygon reduction command from the Model module.
 - If you will be using only one reduction effect, choose **Effect → PolygonFilter**, **Effect → PolygonOptimize**, or **Effect → PolygonRuleBased**, as appropriate. For a quick description of each effect, see *Comparing Reduction Types* on page 321.
 - If you are not certain which reduction effect to use, or if you want to combine different effects, choose **Effect → PolygonReduction**, which includes all parameters for the three effects in one dialogue box.
2. Accept the default parameters or enter some new ones and click **Ok**. The dialogue box disappears and the status line prompts you to click on a model.
3. Pick a polygon mesh model.
 - The effect is applied according to whatever parameters you used.
 - A cone icon appears and is automatically selected.
 - A message appears stating that animation, local materials and textures may now be invalid. This is why you should apply

polygon reduction before defining any shape animation. If you have applied materials and textures, make sure that **Preserve Materials** is selected.



The cone icon is not included when you save your scene. You should make sure that you are satisfied with the effect before you save your scene and exit SOFTIMAGE 3D.

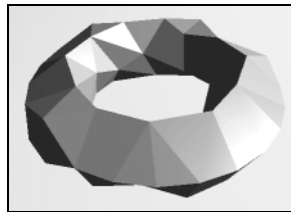
Modifying the Effect

You can modify the effect by editing its parameters. This lets you experiment with different settings while applying, undoing, and reapplying the effect. It also lets you cumulatively apply several reductions to the same model, using different settings at each stage.

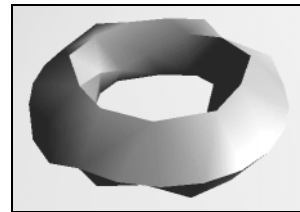
To modify an effect:

1. Select the cone icon representing the effect.
2. Choose **Effect → Custom → Edit Parameters**. The effect's dialogue box is displayed.
3. If you are using the Polygon Reduction effect, select the **Rule Based**, **Optimization Based**, or **Filter** option. For a quick description of each effect, see *Comparing Reduction Types* on page 321.
4. Set the parameters:
 - For information about rule-based reductions, see *Setting Rule Based Options* on page 332.
 - For information about optimization-based reductions, see *Using the Optimization Based Effect* on page 340.
 - For information about the Filter effect, refer to the description of **Effect → PolygonReduction** on page 478 of the *Reference Guide*.
 - See also *Tips and Tricks* on page 343.
5. Apply the effect. The results of the effect appear in the SOFTIMAGE 3D display behind the dialogue box.

- For more information about applying, undoing, and reapplying the effect, see the next section, *Using the Effect*.
- 6. Repeat steps 3 to 5 to experiment with other settings, or to apply different settings cumulatively to the reduced model. For example, you might want to perform a rule-based reduction before an optimization-based reduction in order to reduce the number of calculations. Alternatively, you might want to iterate rule-based reductions using different criteria, or finish off with the Filter effect to clean up your model.
- 7. When you are satisfied with the results, close the dialogue box by clicking on Ok or Cancel.
- 8. The resulting model may appear more faceted than the original. For a smoother version of the model, do the following:
 - Select the model and choose **Info → Selection**.
 - In the Polygon Info dialogue box, select the **Automatic Discontinuity** option. The model now has a smoother, less faceted appearance.



Faceted



Interpolated

Using the Effect

In addition to applying the effect, you can also undo the effect, reapply new settings to the previous version of the model, restore the original model, or highlight edges that would be affected by the current settings.

To perform one of these tasks, select the appropriate settings in the **Processing Options** area and click Apply. If you click Ok, the dialogue box closes but the effect is not applied; however, if you have made any changes to the parameters, they are saved and displayed the next time you open the dialogue box. If you click Cancel, the dialogue box closes and any changes you have made to the parameters are lost.



If you have difficulty applying an effect, or have unexpected results, it is often because you selected the wrong processing option. Always check that the correct option is selected before clicking Apply.

Applying the Effect

To apply the effect with the current settings, make sure that **Standard** is selected in the **Processing Options** area and click Apply.

The effect is applied to the model in its current state. This results in an additive effect. If you have already applied some reduction to the model, the settings are applied to the reduced model. This lets you reduce a model cumulatively, using different settings at each stage.

Undoing the Last Effect

To undo the last **Reapply** or **Standard** operation, make sure that **Undo Last** is selected in the **Processing Options** area and click Apply. Note that you cannot undo an **Original** or **Undo Last** operation.

Reapplying the Effect

Reapplying an effect applies the current settings to the previous version of the model. This results in a non-additive effect. It is equivalent to **Undo Last** followed by **Standard**. To reapply an effect, make sure that **Reapply** is selected in the **Processing Options** area and click Apply.



If you change any of the other processing options (such as **Tagged Only**, **Untagged Only**, **All** or **Follow Hierarchy**), **Reapply** may not behave as expected. For this reason, it is recommended that you use **Undo Last** followed by **Standard** if you want to change the processing options.

Restoring the Original Model

To restore the original model, make sure that **Original** is selected in the **Processing Options** area and click Apply. If you tagged points before you applied the effect for the first time, the original set of tagged points is restored.

Highlighting Affected Edges

Instead of applying the effect, you can highlight the edges that would be affected by the reduction. This allows you to see which edges would be removed by particular settings. If desired, you could then

change settings, or close the dialogue box and tag points that you want to exclude from the reduction.

To highlight edges that will be affected by the current settings:

1. Close the effect's dialogue box.
2. Make the effect's cone icon the parent of the model.

Note

If you move the icon when it is the parent, it may cause uncontrollable and cumulative displacement of the model. Avoid moving the icon if it is the parent, or first cut the hierarchy link between the two.

3. Make sure that the **Show → Edge Flags** command is selected.
4. Select the effect's cone icon and choose **Effect → Custom → Edit Parameters**.
5. In the effect's dialogue box, set the options as desired.

Note

There are several settings that do not work with **Highlight**. In these cases, the **Highlight** option is ignored and the effect is applied immediately.

- For the Rule Based effect, **Highlight** works with all reduction criteria except **None**.
 - For the Optimization Based effect, **Highlight** is ignored.
 - For the Filter effect, **Highlight** works only with:
 - Remove Collinear Edges
 - Quads in the Output area
6. In the Processing Options area, select both **Highlight** and **Standard**, and then click Apply. Edges that would be affected are highlighted in magenta.
 7. If you wish to apply the effect, you must deselect **Highlight** and click Apply.

Controlling the Scope of the Effect

A variety of settings in the **Processing Options** section of the effect's dialogue box let you control the scope of the effect. You can:

- Restrict the effect to tagged or untagged points.
- Include or exclude the hierarchy below the selected model.
- Preserve edges which are boundaries of the model.

- Preserve edges which are boundaries between materials and textures.

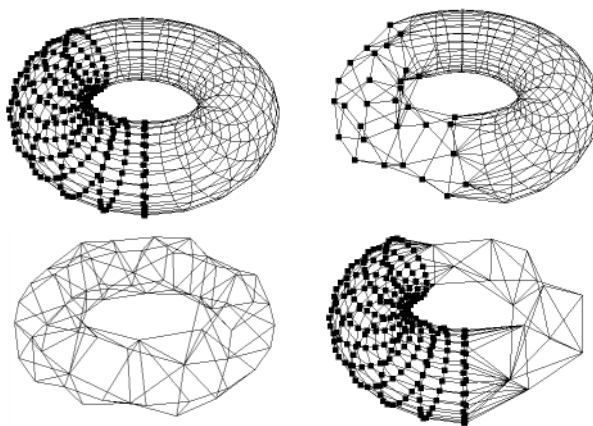
Tagging Points

The **Tagged Only**, **UnTagged Only**, and **All** options control which points are considered by the effect. You must tag points before displaying the effect's dialogue box. To tag or retag points while using an effect, you must exit the dialogue box using Cancel or Ok, change the tagging on the model, then display the dialogue box again.

If you select **Tagged Only**, an edge is considered for reduction only if both vertices are tagged. Any new vertices created by the effect automatically are tagged. If you iterate the effect, these new points are included in the reduction.

If you select **UnTagged Only**, an edge is considered for reduction only if neither vertex is tagged. New vertices created by the effect are untagged. If you iterate the effect, these new points are included in the reduction.

If you select **All**, the entire model is reduced regardless of tagging.



Clockwise from top left: model with points tagged; reduction applied to tagged points only; reduction on untagged points; reduction on all points.

There are a few options where tagging information is not applied as expected:

- The **Merge** option (Filter effect) merges only tagged points regardless of which tagging option you select.

- If you select the criterion **None** (Rule Based effect):
 - The reduction is applied only to tagged areas if either **Tagged Only** or **All** is selected.
 - The reduction is applied to untagged areas if **Untagged Only** is selected.
 - No reduction is applied if no points are tagged.

Applying to a Hierarchy

The **Follow Hierarchy** option applies the reduction settings to the entire branch of the hierarchy below the selected model. If this option is not selected, only the root node of the selected branch is affected.

Preserving Boundaries

The **Preserve Boundaries** option preserves edges with only one adjoining face, as found on the edges of grids or in models with holes. If **Preserve Boundaries** is selected, boundary edges of the model are not processed during a reduction.

There are some dependencies which determine whether a boundary edge is considered for reduction and whether the **Preserve Boundaries** option has any effect.

- For the Rule Based effect, a boundary edge is not considered for reduction if any of the following options is selected:
 - **Preserve Boundaries**
 - **Angle**
 - **Keep Sharp Angles**
 - **Remove**
- For the Filter effect, **Preserve Boundaries** is relevant only with the **Remove Collinear Edges** option.
- For the Optimization Based effect, **Preserve Boundaries** is not relevant.

Preserving Materials

The **Preserve Materials** option preserves edges which are boundaries between different materials or textures. If this option is selected, these edges are not processed.

Setting Rule Based Options

In a Rule Based reduction, you specify a criterion such as length or area, as well as a target such as an exact number of polygons or a percentage. The effect then eliminates edges based on the criterion, until the target is reached.

Setting the options for the Rule Based effect involves the following tasks:

1. Select a criteria (**Radius, Length, Area, Height, Angle** or **None**) used to determine an edge's position in the elimination queue.
2. Specify the desired amount of reduction by selecting a target (**Face Count, % Face Count, Value, or % Value**).
3. Select **Collapse** or **Remove**. These options determine what happens when an edge is eliminated.
4. If desired, use the remaining options to bias the elimination queue according to secondary criteria.

These tasks are described in more detail in the sections that follow. See also *Rule Based Reduction Tutorial* on page 350.

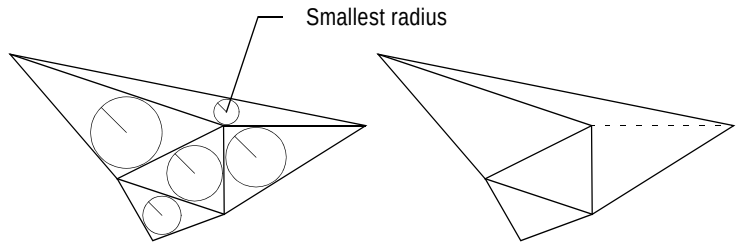
Selecting Rule Based Criteria

Different criteria tend to eliminate different edges. As a result, the reduced model might look quite different, depending on the criterion used. After a while, you will develop an instinct for what types of criteria work best for various types of models and reduction needs. For an illustration of the same model reduced according to different criteria, see *Comparing Rule Based Criteria* on page 337.

Radius

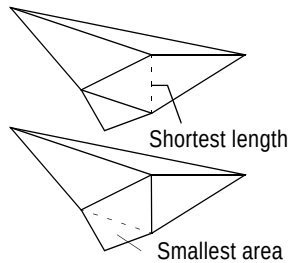
When the **Radius** criterion is selected, the reduction considers each edge in turn. In each adjoining face, it draws a circle. The length of the edge is multiplied by the smaller radius, and the resulting value determines the edge's position in the elimination queue. Edges are

eliminated from the smallest value to the largest until the target is reached.



Reduction by Radius

Preserve Boundaries and Remove on.



Compare with Length and Area

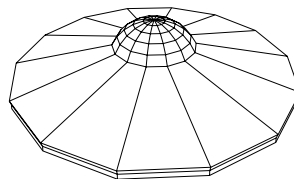
Reduction by Length eliminates shorter edges first.

Reduction by Area removes edges with smallest adjacent polygons.

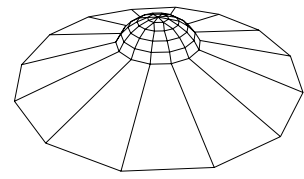
Length

When the **Length** criterion is selected, edges are eliminated based on their length, from shortest to longest until the target is reached.

The following is an example of a reduction by **Length**. The model was reduced from 196 to 156 polygons with the % Face Count target and a value of 20.



Original model



Reduction by Length

While the polygon count of the original is not high, there are a number of unnecessary polygons, particularly along the thick wing edges. This detail does not show up from a distance and is not crucial

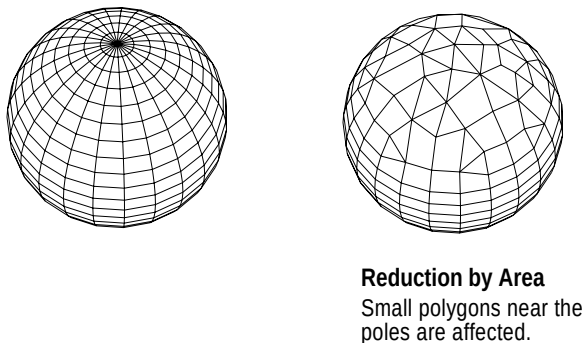
to the model. Furthermore, some of the small polygons at the poles of the sphere can be removed without losing quality or detail.

In this example, a reduction by **Area** would not work well, since the polygons along the wing edges are narrow but still have a fairly large area overall. Instead, a reduction by **Length** removes both the short edges along the wings and those at the poles of the spheres.

Area

When the **Area** criterion is selected, the reduction considers each edge in turn and calculates the area of the two adjoining faces. The smaller result determines the edge's position in the elimination queue. Edges are eliminated from the smallest value to the largest until the target is reached.

Here is simple example of the use and effects of a reduction by **Area**. Start with a polygon sphere with 25 subdivisions in both U and V. Apply a reduction using the **Area** criterion with the **% Face Count** target set to 40. The result looks like this:



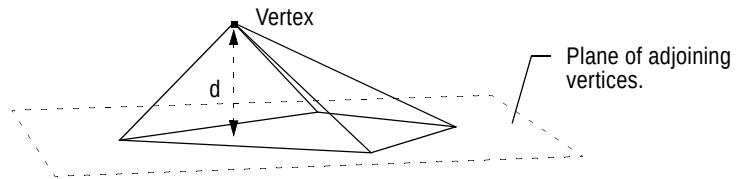
The figure shows clearly how the smaller polygons near the poles are now fewer and larger. Their edges have been eliminated and the sphere is reduced from 625 to 374 polygons.

Height

Unlike other reduction criteria, **Height** eliminates vertices rather than edges, but the end result is the same – fewer polygons. To use this option, you must also select **Remove**.

When the **Height** criterion is selected, the reduction considers each vertex in turn and calculates the average plane of its neighbouring vertices (those vertices which are joined to it by an edge). The height

of the vertex above the plane determines its position in the elimination queue. Vertices are eliminated from the smallest value to the largest until the target is reached.



Height

The shortest distance d between vertex and plane.

Tip

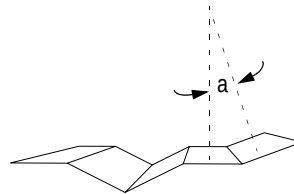
The **Height** criterion can be useful for reducing the number of polygons on a surface which is already relatively smooth, such as a cheek.

You should probably use the **Height** criterion on tagged points only, otherwise you might eliminate vertices in areas you do not want to affect.

Angle

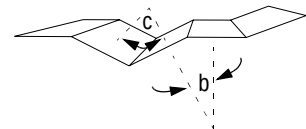
When the **Angle** criterion is selected, the reduction considers each edge in turn and calculates the dihedral angle of its adjoining planes. The size of the dihedral angle determines the edge's position in the elimination queue. Edges are eliminated from the smallest value to the largest until the target is reached.

The dihedral angle of an edge is the angle between the normals of its adjacent faces. Edges on flatter areas of a model have small dihedral angles, while edges on angular areas have large dihedral angles.



Dihedral angles

Angle a is small (less than 10 degrees).

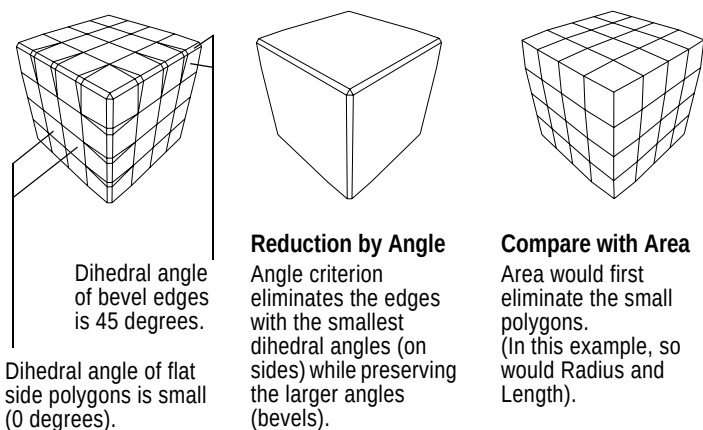


Angle c is largest (corresponding to sharpest edge).

You can also balance a reduction by **Angle** with the **Orient** and **Square** options:

- **Orient** biases the reduction in favour of eliminating those faces which are not oriented directly on the x, y, or z-axes.
- **Square** biases the reduction in favour of eliminating faces which are not square.

In the following example, you would likely want to eliminate unnecessary polygons on the sides of the cube without losing the bevelling detail. If you applied either the **Area** or **Length** criteria, the smaller polygons of the bevel would be removed first. Instead, you can retain the bevel detail and remove flatter polygons with the **Angle** criterion. The results were achieved by selecting **Angle** and a **Value** target of 5 degrees.



None

The **None** criterion lets you eliminate edges directly:

- If you select either the **Tagged Only** or **All** processing option, edges are eliminated if both vertices are tagged.
- If you select **Untagged Only**, edges are eliminated if neither vertex is tagged.
- If no points are tagged, no reduction is performed.

You must tag points before displaying the dialogue box.

Comparing Rule Based Criteria

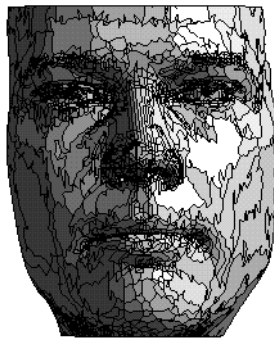
The following illustration shows the results of rule-based reductions using five different criteria. The original is a high-resolution image reduced by 80% from its original 15423 polygons to 3083 polygons in each case.

While the number of polygons is the same in each case, the number of triangles varies greatly as shown in the example of the reduction by **Angle**.

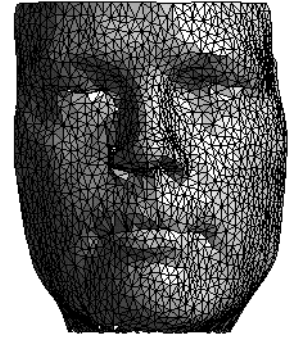
The reduction by **Height** works very well, leaving large polygons on the flat cheeks, and numerous polygons to show the detail of the nose and lips.



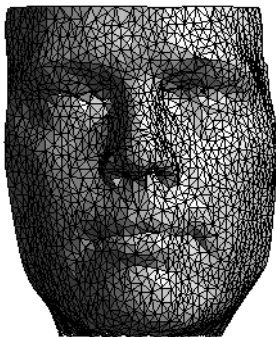
Original (unreduced)
number of polygons: 15423
number of triangles: 15447



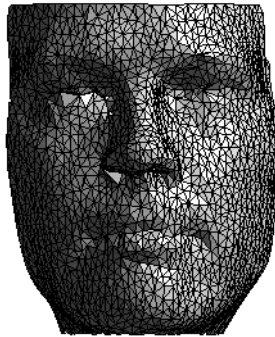
Angle criterion
3084 polygons
12233 triangles



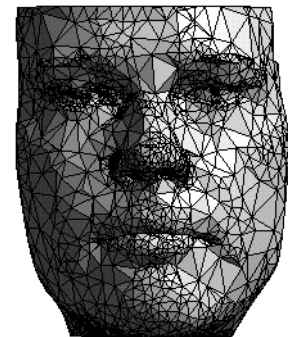
Radius criterion
3083 polygons
3083 triangles



Length criterion
3083 polygons
3083 triangles



Area criterion
3083 polygons
3084 triangles



Height criterion
3083 polygons
3224 triangles

Setting Rule Based Targets

The targets provide different ways of specifying the desired amount of reduction:

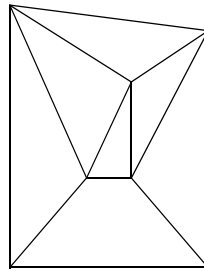
- **Face Count** reduces the number of polygons to the specified value. For example, if you enter 400 in the corresponding text box, the reduced model has 400 polygons.
- **% Face Count** reduces the number of polygons by the specified percentage. For example, if you enter 80 in the corresponding text box and there are 1000 polygons in the model, the reduced model will have 200 polygons.
- **Value** eliminates edges whose criterion is less than the specified value. For example, if you select the **Angle** criterion and enter 30 in the corresponding **Value** text box, edges with dihedral angles of less than 30 degrees are eliminated.
- **% Value** calculates the range of values for the selected criterion and eliminates edges, from smallest value to largest, until the specified percentage is reached in the range of values. For example, if the range of angles in a model is from 10 to 110 degrees, and you specify a reduction by **Angle** with a **% Value** of 50, all edges with dihedral angles of less than 60 degrees would be eliminated.

Removing or Collapsing Edges

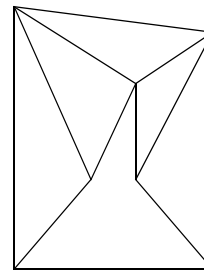
The **Collapse** and **Remove** options control what happens when an edge is eliminated:

- If the **Collapse** option is selected, the two end points of the eliminated edge are merged into a single new vertex. This new vertex inherits the edge connections of the deleted end points.
 - In most cases, the new vertex is created at the midpoint of the eliminated edge.
 - If one of the two end points is on a boundary, the new vertex is created at that position.
 - If the edge is a boundary and the **Preserve Boundaries** processing option is not selected, the new vertex is created at the midpoint. If **Preserve Boundaries** is selected, the edge is not considered for elimination.
 - If the end points are tagged and the **Tagged Only** processing option is selected, the new vertex is also tagged.

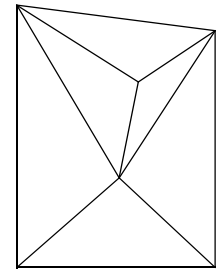
- If the **Remove** option is selected, edges are simply removed. Vertices are neither moved nor created. This option is useful if vertices must stay in the same positions.
 - If all edges referencing a given vertex are removed, then the vertex is also removed.
 - Boundary edges are never removed.



Original polygon mesh



Remove



Collapse

Biasing Rule Based Reductions

You can bias a rule-based reduction by specifying secondary criteria. These secondary criteria affect an edge's position in the elimination queue. These options are useful for balancing reductions. When the bias options are selected, different edges may be eliminated than if the reduction was based on the main criterion alone.

- **Keep Sharp Angles** biases the reduction in favour of keeping edges on sharp angles, regardless of the selected reduction criterion. If this option is selected, then edges with dihedral angles larger than the value specified in degrees are moved to the end of the elimination queue. This is useful for keeping sharp detail while reducing other areas. For a definition of dihedral angles, see the description of the **Angle** criterion on page 335.



If you use **Keep Sharp**, study the model and compare the flat areas to the sharp areas. Enter an angle value somewhere between the dihedral angles of the flat and sharp areas.

- **Keep Big Areas** biases the reduction in favour of keeping edges on polygons with large areas. Use the slider to control the percentage weight or influence of the bias on the reduction. For example, this option is useful in cases where long edges are adjacent to faces with relatively small areas. A reduction by **Length** might not eliminate these edges because of their relative length. However, if you select

Keep Big Areas, the system considers the area of the faces adjoining an edge before it considers just its length.

- The **Orient** and **Square** options are applicable only with the **Angle** criterion. See *Angle* on page 335.

Using the Optimization Based Effect

The Optimization Based effect allows you to reduce the number of polygons in a model while achieving a compromise between fidelity and the reduction target.

This effect is a two-step process. The first step generates an invisible point cloud corresponding to the geometry of the model. The second step uses this point cloud to gradually reduce the polygon count until it achieves the best compromise between fidelity to the model and the specified reduction target.

You can perform these steps together or separately. For example, you can create a point cloud, then reduce the polygon mesh with the Rule Based effect, and finally perform an optimization-based reduction using the original point cloud and the reduced polygon mesh. This approach is recommended because it reduces the time required for the reduction while maintaining optimal fidelity.

There is a tutorial for the Optimization Based effect on page 352.



The **Highlight** and **Preserve Boundaries** processing options are not applicable for optimizations.

The recommended procedure for optimization-based reduction is:

1. Display the effect's dialogue box:
 - If this is the first time you are applying the effect to the model, choose **Effect** → **PolygonReduction**.
 - If you have already applied the effect, select the effect's icon and choose **Effect** → **Custom** → **Edit Parameters**.
2. Select the **Optimization Based** option.
3. The first task is to generate a point cloud. Select **Set Point Cloud** and deselect **Reduce**. The point cloud consists of the vertices of the model, as well as extra points added by the effect.
 - The default options should work well with most models, but you can change them if desired.

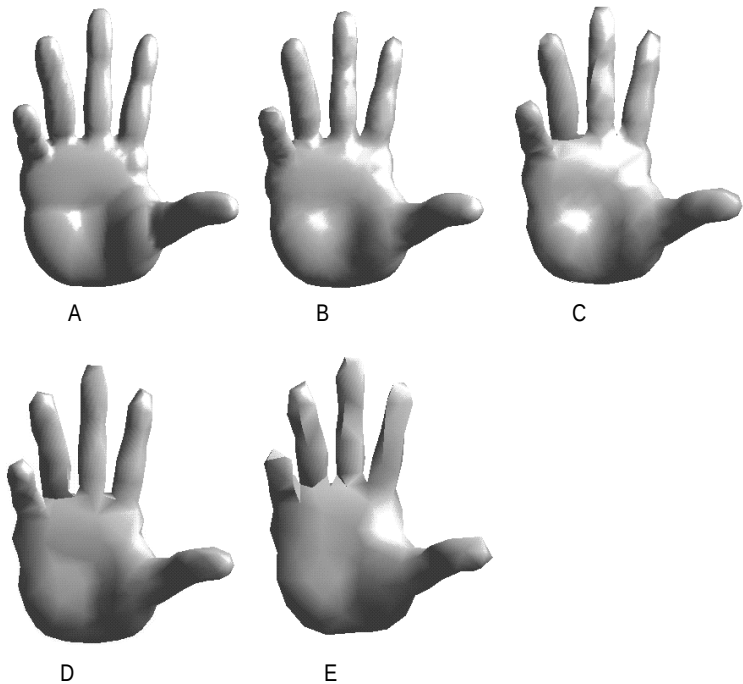
- **Face Sampling** distributes the specified number of points randomly over the model's surface. For performance reasons, this value should be no more than 2000.
 - **Sharp Edge Sampling** specifies the number of points randomly added to the sharp edges. For performance reasons, this value should be no more than 3.
 - **Edges Sharper Than** specifies the dihedral angle above which an edge is considered sharp. For a definition of dihedral angles, see the description of the **Angle** criterion on page 335.
4. Select **Start** in the **Optimization Based** area and **Standard** in the **Processing Options** area, then click **Apply**.
 - If you are applying the effect for the first time, click **Ok** instead of **Apply**, pick the model, then display the dialogue box again by choosing **Effect** → **Custom** → **Edit Parameters**.
 5. If desired, you can significantly reduce the time required for the reduction step of the optimization by applying a rule-based reduction first.
 - Select the **Rule Based** option, and refer to *Setting Rule Based Options* on page 332 for more information.
 6. The next step is to reduce the model based on the point cloud. Make sure that the **Optimization Based** option is selected, then select **Reduce** and deselect **Set Point Cloud**.
 7. Set a reduction target by specifying either the desired number of polygons in the **Face Count** option, or the desired reduction percentage in the **% Face Count** option. Remember that these values are rough guidelines only.
 8. Select **Start** in the **Optimization Based** area and **Standard** in the **Processing Options** area, then click **Apply**.
 9. Go get a coffee. Optimization-based reductions can take a long time, even for relatively simple models. This is because of the amount of computation involved in comparing the starting polygon mesh and point cloud.
 10. To see how the reduction is progressing, make sure that **Standard** is still selected, select **Query** in the **Optimization Based** area and click **Apply**. The current state of the model is shown in the SOFTIMAGE 3D display behind the dialogue box.

- Optimization does not necessarily follow a uniform pace. Changes may be more apparent at the beginning, then there may be little apparent change in the model until the optimization finishes.
- To terminate the reduction before it has finished, make sure that **Standard** is still selected, select **Stop** and click Apply. The reduction is stopped and the SOFTIMAGE 3D display behind the dialogue box is updated.

Note If a reduction has not terminated on its own, you must stop it before you can undo it, return to the original model, or apply a different reduction.

11. When the reduction has finished, there is no visual cue. You must select **Query** and click Apply to update the SOFTIMAGE 3D display, then close the dialogue box by clicking Ok or Cancel.
- To check whether the optimization has finished, close the dialogue box and click on the word **Message**, **Error**, **Warning**, **Status** or **Mode** on the status line. Scroll to the bottom of the See Contents dialogue box. If you see the message “Mesh analysis: FINAL” near the end of the list, the optimization has finished.

The following illustration shows the combined use of optimization-based and rule-based reductions.



A = original model, 9696 polygons
B = after rule-based reduction, 1938 polygons
C = after 2.5 minutes optimization, 892 polygons
D = after 5 minutes optimization, 606 polygons
E = final result, after 10 minutes optimization, 414 polygons

Tips and Tricks

This section contains some information that can help you use polygon reduction more effectively.

Hints

Polygon reduction works best when you first take the time to figure out what results you want to achieve. This section suggests questions you should ask yourself about your model, proposes reduction techniques, and lists some common problems and solutions.

- How important is fidelity to the original model? If there are certain edges that need to stay exactly as they are, try using **Preserve Boundaries** (page 331). If there are features within the model that must remain exactly the same, tag those points and perform a rule-

based reduction on untagged only. If the detail that needs to be preserved has many polygons that could be reduced, you should probably still tag them and reduce them separately.

- Is it important to retain the position of vertices and edges when using a rule-based reduction? Try deselecting the **Collapse** option (page 338), then follow up with the **Remove Collinear Edges** option in the Filter effect. For more information about the Filter effect, refer to the description of Effect → **PolygonReduction** on page 478 of the *Reference Guide*.
- Are there any characteristics of the polygons that are very obvious and would be easy to deal with? For example, if there are large flat areas that are made up of many polygons, tag the area and use a rule-based reduction to reduce it to only a few polygons.
- How important are curves in the model? If the curves are the most important feature, consider using rule-based reduction with the **Angle** criterion, or any other criterion with the **Keep Sharp** option selected (see *Biasing Rule Based Reductions* on page 339).
- Does the model have noisy, highly detailed features that you want to remove, such as hair, grass, or small pointed features? Use a rule-based reduction with the **Area** or **Length** criterion (see *Selecting Rule Based Criteria* on page 332), which removes the smallest polygons. Running this effect with the **Tagged Only** option selected can reduce a very heavy lawn to a flat surface.
- Is a rule-based reduction affecting areas that you don't want it to? Try running it on tagged or untagged points to isolate the area that you don't want to reduce. The **Highlight** option (page 328) serves as a great guide for isolating those areas.
- Even if you intend to run a optimization-based reduction, study the model to see what can easily be removed with rule-based reductions first. The more reductions you can do easily, the faster the final optimization will be.
- Very heavy models, such as those generated by converted meta-clay models, patches, and NURBS surfaces, can often be reduced by 40% to 50% using rule-based criteria without affecting model fidelity.
- When you convert a model to polygons, the result sometimes has many long thin triangles that give a bad tessellation. The Polygon Reduction effect can help remove those triangles and improve the

tessellation. Try applying one or two rule-based reductions by **Area** with the **% Count** target set to 30.

- To collapse edges that are not otherwise removed by one of the automated reduction methods, tag points and merge them manually using the **Merge Tagged Vertices** option in the Filter effect. Alternatively, tag points and perform a rule-based reduction using the **None** criterion.

Avoiding Problems

Polygon reduction effects are somewhat more complex than other custom effects in both their use and interaction. To minimize the most common problems and mistakes in using them, follow these hints.

Important Tips

- Remember that the cone icon is not included when you save your scene. You should make sure that you are satisfied with the effect before you save your scene and exit SOFTIMAGE 3D.
- Polygon reduction overrides shape animation. For this reason, you should reduce the model before animating it. If you have applied materials or textures, make sure to select **Preserve Materials**.
- If the effect's icon is the parent of the model (as is necessary when using the **Highlight** option) and you move the icon, it may cause uncontrollable cumulative displacement of the model. Avoid moving the icon if it is the parent of the model, or else cut the link between the two.
- By default, the reduced model's **Faceted** option is set. To give the model a smoother appearance, do the following:
 - Select the model and choose **Info → Selection**.
 - In the Polygon Info dialogue box, select the **Automatic Discontinuity** option.
- If you close the dialogue box and the area behind it doesn't display properly, press the **r** Supra key to refresh.

Don't Forget

- Select the correct processing option each time: for example, you may forget to change **Original** to **Standard** before clicking Apply.

- Make sure you have the desired effect selected if you are using the Polygon Reduction dialogue box. Otherwise, you may inadvertently apply the wrong effect.
- Enter an associated value when you select a target or a filter type, otherwise they might not have an effect. For example, the % Value target has no effect if you just click Apply: you need to enter a useful value because the default is zero.
- Click on the model when you run the effect for the first time to connect the effect to the model. The instruction in the status line can be easily overlooked.
- Make sure that the effect's icon is selected if you want to edit the effect's parameters.
- Make sure that the model is selected if you want to tag points or get information on the model. For example, to check the polygon count of the model, make sure the model is selected before choosing **Info → Selection**.

Tutorial Examples

The exercises provided in this section are intended to give you a guided tour of the polygon reduction effects, hands-on practice using each of the effects, and the chance to experiment and view the results.

All exercises make use of example models provided for you with SOFTIMAGE 3D. Once you have installed SOFTIMAGE 3D version 3.5 from the CD-ROM, you can find the models in the **PolyRed** database.

Processing Options Tutorial

This section demonstrates the use of the global **Processing Options** common to the Rule Based, Optimization Based, and Filter effects.

Run the Effect to Create the Icon

As with many custom effects available with SOFTIMAGE 3D, the icon associated with the effect is created the first time you run the effect. For polygon reduction, this icon looks like a cone.

Unlike other custom effects, however, it is unlikely that you will achieve the desired result the first time. Successful reduction of a model normally requires experimentation and iteration.

Begin by running the effect on a test model to create the icon and view the initial results.

1. In the Model module, choose **Get → Element**.
2. Load the model **truck-cab-sphere.1-0** from the MODELS directory in the PolyRed database.

This model has a truck cab for a parent, with two spheres as children.

3. Choose **Effects → PolygonReduction**.
4. In the Polygon Reduction dialogue box:
 - Select the **Rule Based** option.
 - Select the **Length** criterion.
 - Select the **% Face Count** target and set its slider to 88 (to reduce the model by 88%).
 - Select the **Collapse** option.
5. In the Processing Options area, click Ok. Your settings are saved and the dialogue box closes.
6. In the SOFTIMAGE 3D display, click the model of the truck cab.

The model is reduced by 88% of its faces. Note that only the truck cab is modified, and not the entire model hierarchy.

Modification 1

In some cases, you may wish to tag certain portions of the model for reduction, leaving the untagged portions unaffected. In this modification, you will tag a subset of the model and apply the reduction to only that tagged subset.

1. Display the effects's dialogue box by choosing **Effects → Custom → Edit Parameters**.
2. Choose **Original** or **Undo Last**, and click Apply.

This step must be performed before you tag any points, otherwise you will lose tagging information.

3. Close the dialogue box by clicking Ok or Cancel.
4. Tag the front section of the cab and half of each sphere.
5. Display the effects's dialogue box again.
6. Select **Tagged Only** and **Standard**, then click Apply.

The reduction settings are applied only to the tagged portion of the cab, leaving the rest of the cab and the spheres unaffected.

Modification 2

In other cases, it may be desirable to tag a portion of the model that you wish to leave “as is” and leave the rest available for reduction. In this modification, you will apply the reduction to only the untagged portion of the model.

1. Select **Undo Last** and click **Apply**.

If you change processing options such as **Tagged Only**, **UnTagged Only**, or **Follow Hierarchy**, it is better to use **Undo Last** followed by **Standard** instead of **Reapply**.

2. To apply the modification to the untagged portion of the model, select **UnTagged Only** and **Standard**, then click **Apply**.

The reduction settings are applied only to the untagged portion of the cab, leaving the tagged portion unaffected.

Modification 3

In this modification, you will apply the reduction to the entire model hierarchy rather than just the parent node.

1. Select **Undo Last** and click **Apply**.
2. To apply the reduction to all points, regardless of the tagging information, select **All**.
3. Select **Follow Hierarchy** and **Standard**, then click **Apply**.

This time, the reduction is applied to both the cab and the spheres. Try this with **Tagged Only** or **UnTagged Only** and see what happens.

Modification 4

In this modification, you will reduce the model while preserving its boundaries.

1. Select **Undo Last** and click **Apply**.
2. Deselect **Follow Hierarchy**.
3. Select **Preserve Boundaries** and **Standard**, then click **Apply**.

In this case, the number of polygons is drastically reduced, but none of the boundary edges have changed. For information on the use and limitations of this option, see *Preserving Boundaries* on page 331.

Filter Tutorial

The Filter effect is useful for cleaning up the model and otherwise modifying it, before or after using the actual reduction tools. This tutorial demonstrates how to use the options of the Filter effect to do the following:

- Generate a triangulated polygon mesh version of your model.
- Generate a quadrilateral polygon version of your model.
- Remove extraneous collinear edges.

Modification 1

For many reasons, including hardware requirements, you may need to generate a triangulated version of the model.

1. Assuming that you are continuing from the previous section, select **Undo Last** and click **Apply**.
2. Select the **Filter** option.
3. In the **Output** area, select **T-mesh**.
4. In the **Processing Options** area again, select **Standard** and click **Apply**.

The model is now triangulated.

Modification 2

In this modification, you will transform the model into quadrilateral polygons. This option requires that the model be triangulated first, which you did in the last modification.

1. In the **Output** area, select **Quad** and leave all other values at the default.
2. In the **Processing Options** area, click **Apply**.

The model is now composed of quadrilateral polygons. Some of the quadrilaterals may be degenerate: they look like a triangle, but there is an extra edge drawn from a point to itself, making four edges in total.

Modification 3

In this modification, you will remove extraneous collinear edges of a model. The Filter effect considers edges to be collinear if they share a vertex that has no other edges; this situation occurs when two faces

share multiple edges. This can occur as a result of certain types of rule-based reductions.

At this point, the example model doesn't have any such collinear edges so you have to create some. First, follow the steps below to run a typical rule-based reduction on the model using **Angle** as the reduction criterion. This creates the kind of collinear edges which can then be removed.

1. Select the **Rule Based** option and set the following options:
 - Select the **Angle** reduction criterion.
 - Select the **Value** reduction target.
 - In the corresponding text box, enter a value of 25. This eliminates all edges with a dihedral angle of less than 25 degrees.
 - Select **Remove**.
2. In the **Processing Options** area, click **Apply**.

There are now a number of places where two faces share a vertex that has only two adjoining edges. These pairs of edges are candidates for removal using **Remove Collinear Edges**. Note that the default angle under which edge pairs are collapsed to a single edge is 30 degrees.

3. Select the **Filter** option.
4. Select the **Remove Collinear Edges** option.
5. In the **Processing Options** area, click **Apply**.

Collinear edges are removed.

Rule Based Reduction Tutorial

To apply a rule-based reduction, you specify a reduction target and a criterion. You can also specify whether to remove or collapse edges, as well as specify how to bias the reduction according to secondary criteria.

1. Start a new model:
 - Delete the current model.
 - Choose **Get → Element**.
 - Load the model **truck-cab.1-0** from the **MODELS** directory in the **PolyRed** database.
2. Choose **Effects → PolygonReduction**.
3. Select the **Rule Based** option and set the following options:

- Select the **Length** reduction criterion.
 - Select the **Face Count** reduction target and set the value to 300 (to reduce the model to 300 faces).
 - Make sure that the **Collapse** option is selected.
4. In the **Processing Options** area, click **Ok**.
 5. Pick the model of the truck cab.

The truck cab is reduced to 300 faces and the effect's icon is created. Note that you could have obtained a similar reduction using any of the other target options. For example, since the original model has 2656 polygons, you could have reduced it to about 300 faces using a **% Face Count** of 11.

Changing the Criterion

In this part of the tutorial, you will experiment using the same target as before, but different criteria.

1. Choose **Effects** → **Custom** → **Edit Parameters** to display the effect's dialogue box.
2. Select a different criterion and enter 300 in the appropriate **Face Count** box.
3. In the **Processing Options** area, select **Reapply** and click **Apply**.
The new settings are applied to the previous version of the model.

Notice that although the reduced model has the same number of faces, different edges have been removed. Repeat steps 2 and 3 using different criteria. Remember that you must select **Remove** for the **Height** criterion to have any effect.

Collapse versus Remove

This part of the tutorial demonstrates the difference between **Collapse** and **Remove**.

1. Select the **Area** criterion.
2. Make sure that **Collapse** is selected.
3. Make sure that **Reapply** is still selected and click **Apply**.
 - Notice how vertices have been rearranged. If **Preserve Boundaries** was not selected, boundary edges have also moved.
4. Select **Remove** and click **Apply**.

Notice that vertices have not been moved. However, there are now a great deal of irregular, many-sided polygons with collinear edges.

Try repeating this exercise using other criteria.

Biasing the Reduction

Now you will bias the reduction using secondary criteria to achieve a more balanced effect.

1. Select the **Length** criterion and make sure that **Collapse** is selected and **Keep Sharp Angles** is deselected.
2. Take a good look at an area of the model that has short, angular edges, such as between the cab door and windshield.
3. Make sure that **Reapply** is still selected and click Apply.
 - Look at the reduced model and see how edges were eliminated.
4. Select **Keep Sharp Angles** and click Apply.
 - Notice how different edges are eliminated. There are more edges in the angular areas of the model, even if they are relatively short.

Repeat this exercise using the slider to specify a different angle. Try it using **Keep Big Areas** instead of **Keep Sharp Angles**, or combining the two. If you set the criterion to **Angle**, you can also see the effect of the **Orient** and **Square** options.

Optimization Based Reduction Tutorial

In this tutorial, you will use an optimization-based reduction to achieve a balance between fidelity to the original model and the number of polygons.

1. If you are continuing from the previous tutorial, delete the model.
2. Load **cow.hide7.1-0** from the **MODELS** directory in the **PolyRed** database.
3. Choose **Effects** → **PolygonReduction**.
4. Select the **Optimization Based** option, deselect **Reduce** and make sure that **Start** and **Set Point Cloud** are selected.
5. Make sure that **Standard** is selected and click Ok.
6. Click on the cow's hide.

The effect's dialogue box disappears and an invisible point cloud is generated.

7. Choose **Effects → Custom → Edit Parameters** to display the effect's dialogue box again.
8. The next step is to lighten the model somewhat with a rule-based reduction. This shortens the amount of time required for the optimization.
 - Select the **Rule Based** option.
 - Select the **Angle** criterion.
 - Select the **% Face Count** target and enter a value of 50.
 - Make sure that **Standard** is still selected and click **Apply**.
9. Now, begin the optimization using the original point cloud and the reduced model.
 - Select the **Optimization Based** option.
 - Select **Reduce** and deselect **Set Point Cloud**.
 - Set **Face Count** to 350.
 - Deselect **Uniform**. This allows a greater concentration of vertices in areas of high detail.
 - Make sure that **Start** and **Standard** are selected, and click **Apply**.
10. After a while, check on the effect's progress. In the **Optimization Based** area, select **Query** and click **Apply**. The SOFTIMAGE 3D display behind the dialogue box is updated.
11. After a few minutes, the optimization process should have finished. To update the display, select **Query** and click **Apply** again.
12. Close the dialogue box by clicking **Ok** or **Cancel**.
13. Select the model and choose **Info → Selection**. Notice how the **Face Count** you specified was used as a rough guideline only.

Webbing Models

The **Effect → Web** command in the Model module allows you to convert a detailed, intricate model to a surface that represents only the external appearance of the original, ignoring any internal structure. The output surface can be a polygon mesh, or a cubic or quadratic NURBS surface.

The Web effect is designed to retain overall form, while greatly reducing the number of control points required to represent the surface. Because of this, it does not produce good results when a model has small but critical details.

Because it smooths out details, the Web effect is useful for developing games and real-time simulators, where you must meet stringent performance requirements when rendering.

The Web effect is also a useful first step before using the **Effect → CreateNURBSNet** command to convert a model to QTM (quadratic texture maps), particularly if you also want to simplify the model.

It is also useful for converting heavy objects, such as those from a three-dimensional geometric scanner, into reduced models with quadrilateral polygons.

Note

The Web effect does not preserve deformation or shape animation. Therefore, it should be applied before you define any such animation for the model. In addition, textures must be bound to the model's UV coordinates with the **Txt_Oper → Info UV Coord.** command in the Matter module.

How It Works

The Web effect works by drawing a web-like “sleeve” around the model, using a non-NURBS curve as a guide. This sleeve is then collapsed until it fits the model; the process is similar to shrinkwrapping an article in plastic.

Depending on the settings, there may be an additional pass. This second pass uses raytracing to project the results of the first pass onto the model. The projection may be parallel or cylindrical.

Overview of the Procedure

Applying the Web effect involves two steps:

- Apply the effect the first time by choosing **Effect** ➔ **Web**. This creates a web icon representing the effect.
- Modify the effect by editing its parameters. Like polygon reduction, the Web effect needs a great deal of your input. You should experiment with different settings until you achieve the desired result.

For a detailed description of the procedure and options, see page 508 of the *Reference Guide*.

Generating Quadratic Texture Maps

QTMs (quadratic texture maps) are quadratic NURBS surfaces with associated textures. They are required by the graphics hardware of certain games platforms.

The **Effect → CreateNURBSNet** command in the Model module generates QTMs. This effect can be run either on polygon meshes with triangular or quadrilateral faces, or on quadratic NURBS surfaces. It is optimized for models generated by the **Effect → Web** command. Use polygon meshes if you want to keep a higher level of detail. If you want to eliminate detail in order to speed up rendering, use NURBS surfaces.

When the effect converts a polygon mesh, it creates a separate QTM for each face. Similarly, when the effect converts a quadratic NURBS surface, it creates a separate QTM for each area of the surface bounded by control points. In both cases, the QTMs are grouped in a hierarchy under a null object. The original model is not destroyed.

For a detailed description of the procedure and options for generating QTMs, see page 462 of the *Reference Guide*. After you have run the effect, you can optimize the textures further by selecting them in the Schematic view and choosing the **Mat_Oper → Optimize Selected** command in the Matter module.



Textures must be bound to the model's UV coordinates with the **Txt_Oper → Info UV Coord.** command in the Matter module before using **Effect → CreateNURBSNet**.

Index

Numerics

- 2D chains 229
- 3D chains 229
- 3D objects
 - creating by extrusion 131
 - creating by revolution 135
 - creating by skinning 138
 - creating from curves 140
 - primitives 5
- 3D space, origin 4

A

- Actor 227
 - resetting initial position 237
- Adding
 - lines and arcs 115
 - points 124, 125
 - polygons 53, 63
 - rows 177, 179
 - vertices 65
- Aligning objects 47
- Animating Boolean operations 40
- Arcs
 - adding to curves 115
 - creating 90
- Articulated chains 229
 - creating 231
 - effectors 230
 - envelopes 245
 - hierarchies 288
 - inverse kinematics 227
 - joints 230
 - editing 233
 - preferred angles 232, 234
 - roots 229
 - tips for drawing 232
 - toggling 2D, 3D 232
- Assigning envelopes 247
 - initial assignment 248
- Automatic envelopes 250

Autotracing 106

B

- Bevel custom effect 135
- Bevelling edges 68, 135
- Bezier curves
 - autotracing 106
 - editing 120
 - information 120
- Blending, see *Merging*
- Boolean operations 38
 - animating 40
 - custom effects 40
 - tips 39
- Border custom effect 116
- Bounding models 261
 - combining 263
 - creating 264
 - editing 264
 - types 262
- Breakup custom effect 78
- B-Spline curves 90
- BumpMap custom effect 307

C

- Camera, displaying 22
- Capping surfaces 183
- Cardinal curves 90
- Centre (CTR) manipulation mode 26
- Centres
 - cluster 308
 - copying 37
 - definition 4
 - displaying 21
 - in polygon modelling 77
- Chains, see *Articulated chains*
- Clearing the workspace 18
- Clipping planes 60, 125
- Clusters
 - centres 308
 - nulls 275
 - creating 308
 - deformation 308

- on curves 126
- tagging points 43
- transforming 309
- updating points in 309
- Colours
 - envelope vertices 257
 - joints 241
- Constraints with null objects 275
- Continuity
 - Bezier curves 120
 - NURBS curves 95
 - editing 123
 - NURBS surfaces 179
- contour standalone 107
- Control points 92
 - displaying 21
 - NURBS curves 94
 - see also *Points*
- Converting
 - curves
 - to faces 117, 150
 - to other curve types 117
 - faces
 - to other types 152
 - to polygon mesh 57, 152
 - meta-clay to polygon mesh 56
 - polygon mesh to surface 166
 - polygons to triangles 70
 - surfaces to polygon mesh 56, 166
- Coordinate systems, polygons 74
- Coplanar custom effect 65
- CopyCenter custom effect 37
- CopyGeometry custom effect 306
- Copying, see *Duplicating*
- CopyTransform custom effect 31
- CreateNURBSNet custom effect 356
- Creating
 - 3D objects from curves 130
 - articulated chains 231
 - bounding models 264
 - clusters 308
 - faces 150
 - hierarchies 282
 - lattices 296
 - meta-clay objects 209
 - modelling relation 144
 - polygon mesh objects 52
 - skeletons 235
 - surface objects 160
 - text 152, 153
- CTR manipulation mode 26
- Curves
 - adding lines and arcs 114, 115
 - adding points 124
 - Bezier 89
 - editing 120
 - borders 116, 141
 - B-Spline 90
 - Cardinal 90
 - closing 108
 - clusters 126
 - control points 92
 - converting
 - to faces 117, 150
 - to other curve types 117
 - copying 104
 - cross-sections, see *skinning*
 - cutting 114
 - definition 5
 - deformation 300
 - deleting
 - control points 126
 - projected curves 201
 - direction 127
 - drawing 101
 - duplicating 104
 - editing 119
 - resolution 128
 - extracting
 - contour standalone 107
 - from imported images 106
 - from surfaces 104, 173
 - projected curves 199
 - relational modelling 174
 - extruding 131
 - polygons along 81

- fractals 129
 - inverting direction 128
 - joining 110
 - Linear 89
 - matching points 127
 - merging 110
 - moving points 119
 - opening 108, 127
 - projecting 199
 - parallel projection 191
 - perspective projection 192
 - preview 190
 - projection methods 191
 - restrictions 187
 - results 193
 - revolving 135
 - sampling 189
 - see also *NURBS curves*
 - skinning 138, 139
 - starting point 127
 - steps 93, 101
 - Custom effects
 - Bevel 135
 - Border 116
 - Breakup 78
 - BumpMap 307
 - Coplanar 65
 - CopyCenter 37
 - CopyGeometry 306
 - CopyTransform 31
 - CreateNURBSNet 356
 - Eater 65
 - EffectsText 154
 - Equalize 67
 - extrusion 134
 - Flake 129
 - Fractalize 46
 - GetEffectsFont 154
 - Glue 59
 - GuidedExtrude 134
 - HrcBorder 116
 - HrcShrinkwrap 306
 - Magnet 307
 - MakeLattice 297
 - Measure 20
 - MergeGeometry 59, 113
 - ModelNote 20
 - Mosaic 65
 - MultiSlice 40
 - Patch2Mesh 167
 - PatchFormula 161
 - Pipe 116
 - PolyBevel 69
 - Polygon Reduction 321
 - PolyOrder 71
 - PolySkin 59
 - Push 307
 - ResetTransf 30
 - SaveEffectsFontCustom 154
 - Shoe 59
 - Shrinkwrap 306
 - Slice 40
 - Splines2Faces 117
 - SplineTag 44
 - Stitch 59, 113
 - SurfaceTag 44
 - Sweep 135
 - TBranch 135
 - VolumeDuplicate 37
 - Wakeup 73
 - Web 354
 - Weld 59
 - Cutting
 - curves 114
 - surfaces 175
- ## D
- Deformation
 - by curve 298, 299, 300
 - by surface 298, 299, 302
 - by texture maps 307
 - clusters 308
 - deactivating 305
 - envelopes 265
 - freezing 305
 - lattices 294

- methods 293
- Quick Stretch 314
- removing 305
- resetting 305
- saving shapes 311
- shrinkwrapping 306
- Degree, NURBS curves 95, 157
- Deleting 17
 - generator curves
 - holes in faces 151
 - points from curves 126
 - polygons from objects 65
 - rows on surfaces 180
 - selected items 17
 - trim or projected curves 201
 - unnecessary vertices 71
- Deselecting elements 15
- Detaching holes in faces 151
- Displaying
 - camera information 22
 - centres 21
 - meta-clay
 - influence zone 211
 - surfaces 211
 - objects 23
 - attributes 21
 - information 19
 - preferred settings 22
- Drafting command 114
- Drawing curves 101
- Duplicating
 - curves 104
 - objects 32
 - mirror images 36
 - using custom effects 37
 - polygons 33
 - transformations on objects 31

E

- Eater custom effect 65
- Edges
 - adding/removing 67
 - beveling 68
 - optimizing 71
 - rounding 68
- Editing
 - curves 119
 - envelope weights 265
 - envelopes 259
 - assignment 260
 - joints in chains 233
 - meta-clay systems 212
 - NURBS curves 123
 - polygon mesh objects 61
 - proportional editing 45
 - skeleton bounding models 264
 - surface resolution 182
- Effectors, see *Articulated chains*
- EffectsText custom effect 154
- Encapsulated PostScript files 106
- Envelopes 246
 - assignment 247
 - colour coding 257
 - depth 249
 - editing 259
 - initial 248
 - automatic 250
 - bounding models 261
 - choosing 255
 - colour coding 241
 - changing 257
 - deformation 245
 - deformation curves 265, 266
 - editing 259
 - flexible 250
 - global 252
 - local 252
 - meta-clay 253
 - null objects 276
 - positioning on skeletons 247
 - reassigning
 - automatically 259
 - manually 260
 - rigid 261
 - resetting initial position 237
 - rigid 253

- reassigning 261
- transformations 254
- undoing operations 269
- vertices 250
- weights
 - editing 265
 - initial 248
- Eps2soft custom effect 106
- Equalize custom effect 67
- Extracting
 - contours 107
 - curves
 - contour standalone 107
 - from imported images 106
 - from surfaces 104, 173
 - projected curves 199
- Extruding
 - axis of extrusion 132, 144
 - curves 131
 - custom effects 134
 - faces 152
 - polygons along curves 81

F

- Faces
 - converting 152
 - creating 150
 - detaching/deleting holes 151
 - extruding or revolving 152
 - holes 150
- Flake custom effect 129
- Flexing, Quick Stretch 314
- Fractalize custom effect 46
- Fractals
 - curves 129
 - polygon mesh 46
- Freezing
 - deformation 305
 - lattices 297
 - modelling relation 146
 - transformations 30

G

- Geometry of objects 5
- GetEffectsFont custom effect 154
- Ghosting objects 25
- Glue custom effect 59
- GuidedExtrude custom effect 134

H

- Hierarchies
 - articulated chains 288
 - creating 282, 287
 - manipulating 29, 286
 - null objects in 275
 - selection modes 15, 283
 - transformations 286
- Holes in faces 150
- HrcBorder custom effect 116
- HrcShrinkwrap custom effect 306

I

- Importing
 - .eps files 106
 - meta-clay files 210
- Information
 - Bezier curves 120
 - meta-elements 212
 - objects 19
- Instances
 - converting to objects 35
 - swapping with masters 35
 - versus masters 34
- Inverse kinematics, articulated chains 228

J

- Joining
 - curves 110
 - points on objects 59
 - see also *Merging*
 - surfaces 168
- Joints 230
 - colours 241
 - moving 233

K

Kinematic chains, see *Articulated chains*

Knots, NURBS curves 94

L

Lattices 294

creating 296

freezing 297

Linear curves 89

M

Magnet custom effect 307

MakeLattice custom effect 297

Manipulation modes 26

Centre (CTR) mode 26

Object (OBJ) mode 26

Polygon (POL) 74

Tag (TAG) mode 26

Texture (TXT) mode 26

Measure custom effect 20

MergeGeometry custom effect 59, 113

Merging

curves 110

objects 58

surfaces 168, 169

Meta-Clay 207

blending 215

converting to polygon mesh 56

defining groups 216

displaying surfaces 211

elements

creating 209

definition 207

displaying surfaces 22

editing 212

importing 210

influence zone

definition 207

displaying 211

editing 212

materials and textures 214

rendering slice 213

resolution 210, 213

shapes, editing 212

systems

animating 218

as envelopes 253

creating 209

saving shapes 221

wireframe slice 213

Modelling 4

deformation methods 293

polygons 74, 79

precision modelling techniques 72

relational, see *Relational modelling*

selected polygons 74

ModelNote custom effect 20

Mosaic custom effect 65

Moving points 44

MultiSlice custom effect 40

N

Normals 7

displaying 21

Null objects 273, 277

cluster centres 275

constraints 275

in hierarchies 275

in skeletons and envelopes 276

instances 276

REF transformation mode 276

reference frames 72, 276

texture propagation 275

NURBS curves

adding points 123

blending 111

centripetal 97

chord length 97

closing 108

continuity 95

control points 94

cubic 96

curve fitting parameters 100

cutting 114

degree 95

- editing
 - continuity 123
 - weight 121
- extracting 173
- knots 94, 95
- linear 95
- merging 111
- non-uniform 98
- parameterization
 - effects 99
 - options 97
- projecting onto surfaces 185
- quadratic 96
- reparameterizing 103
- setting initial parameters 102
- setup 102
- skinning 139
- subdividing 123
- trimming surfaces 194, 197
- types 95
- uniform 98
- weight 96

NURBS surfaces

- adding rows 177, 179
- centripetal 97
- chord length 97
- continuity 179
- cutting 175, see also *trimming*
- definition 157
- deleting rows 180
- effects of parameterization 158
- modelling 185
- non-uniform 98
- opening and closing 162
- parameterization 97
- projecting curves onto 187
- texture mapping 158
- trimming 194
 - options 195
 - restrictions 197
- uniform 98
- weight 96

O

- Object (OBJ) manipulation mode 26
- Objects 5
 - aligning 47
 - creating
 - from skinned curves 138
 - mirror images 36
 - using Boolean operations 38
 - deformation
 - by curve or surface 298
 - by surfaces 302
 - lattices 294
 - displaying and hiding 23
 - displaying bounding box 24
 - duplicating 32
 - edges 67
 - geometry 5
 - ghosting 25
 - in hierarchies
 - selecting 283
 - transforming 286
 - in skeletons 238
 - moving points on 44
 - normals 7
 - null 273
 - picking 14
 - selecting
 - single or multiple 14
 - within hierarchies 15
 - tagging points 42
 - types 5
 - viewing information 19
- Optimizing polygon mesh objects 71, 319

P

- Parameterization
 - effects 158
 - non-NURBS curves 97
- NURBS
 - centripetal 97
 - chord length 97
 - effects 99

- non-uniform 98
 - types 97
 - uniform 98
- Patch 157
 - custom effects 161
 - deformation 302
- Patch2Mesh custom effect 167
- PatchFormula custom effect 161
- Picking objects 14
- Pipe custom effect 116
- Planes, clipping 60, 125
- Points 4
 - adding 124, 125
 - clusters 43
 - deleting 126
 - from polygon mesh 65
 - joining 59
 - moving 44
 - polygon mesh, see *Vertices*
 - randomizing 45
 - scaling 26, 28
 - see also *Control points*
 - selecting 42
 - tagged, editing 43
 - tagging, untagging 42, 44
- POL manipulation mode 74
- PolyBevel custom effect 69
- Polygon (POL) manipulation mode 74
- Polygon mesh
 - adding and deleting vertices 65
 - adding polygons to 53
 - beveling 68, 69
 - converting to surfaces 166
 - converting to triangles 70
 - creating
 - from faces 57
 - from meta-clay system 56
 - from multiple objects 58
 - from surface object 56
 - user-defined 53
 - using predefined objects 52
 - editing 61
 - merging 58
 - optimizing 71, 319
 - precision modelling 72
 - rounding edges 68
 - see also *Polygons*
 - subdividing 70
 - triangulating 70
 - types 52
- PolygonFilter effect 322
 - tutorials 349
- PolygonOptimize effect 322, 340
 - tutorial 352
- PolygonReduction effect 321
 - tutorials 346
- PolygonRuleBased effect 322, 332
 - tutorials 350
- Polygons
 - adding 53, 63
 - centres 77
 - coordinate system 74
 - deleting 65
 - duplicating 33
 - edges, adding/removing 67
 - extrusion along curves 81
 - curve placement 82, 83
 - illustrations 85
 - modelling 74, 79
 - optimizing number 319
 - POL mode 74
 - selecting 61
 - subdividing 70
 - transformations 62
 - vertices
 - adding and deleting 65
 - reordering 71
- PolyOrder custom effect 71
- PolySkin custom effect 59
- Precision modelling techniques 72
- Preferences
 - saving 22
 - setting 22
 - transforming hierarchies 286
- Primitive objects 5
- Projecting curves onto surfaces 185, 199

- parallel projection 191
- perspective projection 192
- Proportional editing 45
- Push custom effect 307

Q

- Quadratic texture maps (QTMs) 356
- Quick Stretch
 - creating 315
 - deformation modes
 - flexing 314
 - stretching 314
 - yielding 314

R

- Raycasting polygon selection 62
- Reassigning envelopes
 - automatically 259
 - manually 260
- Reducing number of polygons 321, 354
- Reference (REF) transformation mode 276
- Reference objects, nulls 72
- Relational modelling 143
 - creating 3D objects 130
 - creating a modelling relation 144
 - curves projected on surfaces 200
 - deactivating 146
 - deleting generator
 - editing generated object 145
 - editing the generator 145
 - effects on generated objects 130
 - extracted curves 105, 174
 - removing links 146
 - revolving a curve 136
 - skinning curves 141
 - trimming surfaces 200
 - viewing 144
- Removing
 - deformation 305
 - see also *Deleting*
- Resetting skeletons 237
- ResetTransf custom effect 30
- Resolution

- curves 128
- meta-clay 213
- surfaces 182
- Revolving curves
 - modelling relation 135
 - process 135
- Revolving faces 152
- Rotation 28
- Rounding edges 68

S

- Sampling projected curves 189
- SaveEffectsFontCustom effect 154
- Scaling
 - points 26
 - see also *Transformations*
 - tagged points 28
- Selecting
 - branch selection 16
 - clearing selections 15
 - multi-node selection 16
 - multiple elements 14
 - node selection 15
 - objects in hierarchies 15, 283
 - picking 14
 - points 42
 - polygons on objects 61
 - single elements 14
 - toggling selections 15
 - tree selection 16
- Shoe custom effect 59
- Shrinkwrap custom effect 306
- Skeletons
 - anatomy 227
 - articulated chains 231
 - bounding models, see *Bounding models*
 - chain roots 229
 - colours 241
 - composition 227
 - defining 235
 - envelopes 245
 - grouping objects as 235
 - hierarchies

- creating 235
- examples 239
- levels 237
- objects in 238
- inverse kinematics 228
- labels in Schematic 236
- null objects 276
- objects in 238
- resetting initial position 237
- restrictions 240
- Skin, see *Envelopes*
- Skinning
 - between surfaces 172
 - curves 138
 - matching points 127
 - order of selection 138
 - relational modelling 141
 - vertices on polygon mesh 59
- Slice custom effect 40
- Splines2Faces custom effect 117
- SplineTag custom effect 44
- Step, curves 93, 101
- Stitch custom effect 59, 113
- Stretching, see *Quick Stretch*
- Subdividing
 - NURBS curves 123
 - NURBS surfaces 177, 179
 - polygons 70
- Surfaces 157
 - capping 183
 - converting to polygon mesh 166
 - creating 160
 - creating from four border curves 141
 - cutting 175
 - deformation 298
 - deleting rows 180
 - inverting curve direction 182
 - inverting U and V directions 164
 - merging 168
 - modifying 181
 - moving starting point 164
 - normals 7
 - opening and closing 162, 163

- resolution 182
 - editing 182
 - Shade view 183
- see also *NURBS surfaces*
- skinning between 172
- types 157, 160
- zipping 170
- SurfaceTag custom effect 44
- Sweep custom effect 135

T

- Tag (TAG) manipulation mode 26
- Tagging points 42
- TBranch custom effect 135
- Text, creating 152, 153
- Texture (TXT) manipulation mode 26
- Texture mapping 158
- Tracing pictures 106
- Transformation modes 26
- Transformations
 - absolute values 27
 - additive values 27
 - clusters 309
 - copying 31
 - freezing 30
 - on hierarchies 286
 - polygons 62
 - resetting 29
 - rotation 28
 - scaling 26
 - translation 28
- Translation 28
- Trimming NURBS surfaces 187
 - deleting trim curves 201
 - error messages 202
 - intersecting trim regions 196
 - options 195
 - relational modelling 200
 - restrictions 197
 - tips 198
- TXT manipulation mode 26

U

Undoing operations on envelopes 269

V

Vertices

- adding and deleting 65

- envelopes

 - colouring 241, 257

 - initial assignment 248

 - mapping 250

 - reassigning automatically 259

 - reassigning manually 260

- removing unnecessary 71

- reordering 71

- skinning between 59

VolumeDuplicate custom effect 37

W

Wakeup custom effect 73

Web effect 354

Weight, see *NURBS curves*

Weld custom effect 59

Y

Yielding, see *Quick Stretch*

