SOFTIMAGE®

SOFTIMAGE®|XSI®

Version 4.0

Modeling & Deformations

Documentation Team
Judy Bayne, Grahame Fuller, Edna Kruger, Luc Langevin, and Jamal Rahal.

05 2004

Contents

Modeling & Deformations

Section I • Modeling & Deformation Basics

Section II • Polygonal Modeling

Section III • NURBS Modeling

Roadmap

Where to Find Information

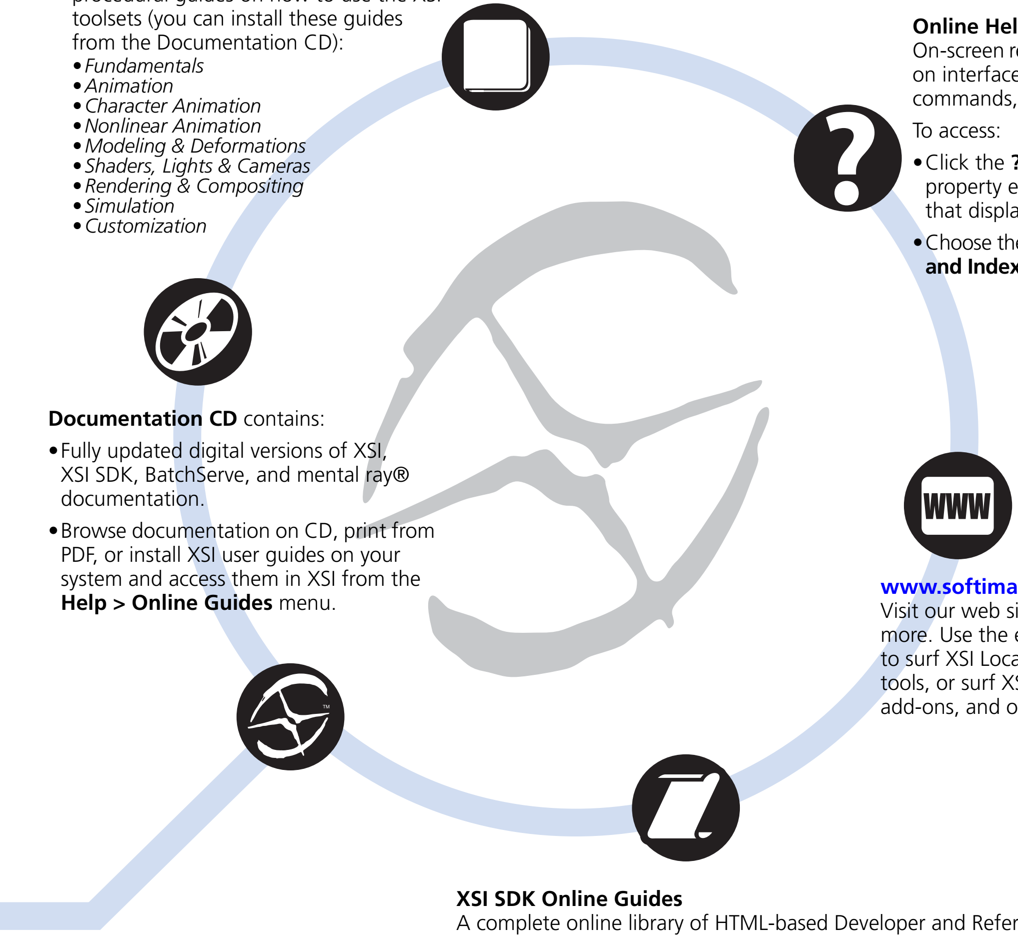
The SOFTIMAGE®|XSI® package includes a comprehensive set of learning and reference materials. Use this Roadmap to find the information you need to get up and running quickly and effectively.

Start with the ***Setup & Licensing*** guide to install, configure and license all components.

Familiarize yourself with the ***Limitations & Workarounds*** for this version, at **www.softimage.com/support**.

Explore ***The Principles of XSI***—a collection of video clips on DVD-ROM providing overviews of the core features, tools and workflows in XSI. New features for this release are also showcased here.

If you are new to SOFTIMAGE|XSI, work through the ***Tutorials*** to learn XSI in the context of basic productions. The tutorials provide a full-color set of lessons showing you how to perform typical tasks step-by-step. You can install the scenes from the ***Media CD***.



XSI Online Guides
A complete online library of HTML-based procedural guides on how to use the XSI toolsets (you can install these guides from the Documentation CD):

- *Fundamentals*
- *Animation*
- *Character Animation*
- *Nonlinear Animation*
- *Modeling & Deformations*
- *Shaders, Lights & Cameras*
- *Rendering & Compositing*
- *Simulation*
- *Customization*

Online Help
On-screen reference information on interface elements, commands, and parameters.

To access:

- Click the **?** button in any property editor or data view that displays this symbol.
- Choose the **Help > Contents and Index** menu.

Documentation CD contains:

- Fully updated digital versions of XSI, XSI SDK, BatchServe, and mental ray® documentation.
- Browse documentation on CD, print from PDF, or install XSI user guides on your system and access them in XSI from the **Help > Online Guides** menu.

www.softimage.com
Visit our web site for tutorials and more. Use the embedded Net View to surf XSI Local for samples and tools, or surf XSI Net for scripts, add-ons, and other resources.

XSI SDK Online Guides
A complete online library of HTML-based Developer and Reference guides on how to use the SOFTIMAGE|XSI SDK toolset. To access: open the script editor and click **?**.

Context-sensitive help on all scripting commands, objects, methods and properties is available through the online *Scripting Reference*. To access: open the script editor and choose **Help > Scripting Reference** or click **F1** while you are working in the editing pane.

Document Conventions

The following are ways that information is displayed in the SOFTIMAGE|XSI documentation.

Typography Conventions

Type style	Usage
Bold	Menu commands, dialog-box and property-editor options, and file and directory names.
Italics	Definitions and emphasis.
`Courier`	Text that you must type exactly as it appears. For example, if you are asked to type `mkdir style`, you would type these characters and the spacing between words exactly as they are appear in this book.
>	The arrow (>) indicates menu commands (and subcommands) in the order that you choose them: *Menu name* > *Command name*. For example, when you see **File > Open**, it means to open the **File** menu and then choose the **Open** command.

Visual Identifiers

These icons help identify certain types of information:

Notes are used for information that is an aside to the text. Notes are reminders or important information.

Tips are useful tidbits of information, workarounds, and shortcuts that you might find helpful in a particular situation.

The **3D icon** indicates information about differences in workflow or concepts between SOFTIMAGE|3D and SOFTIMAGE|XSI.

Warnings are used when you can lose or damage information, such as deleting data or not being able to easily undo an action. Warnings always appear *before* you are about to do such an action!

Keyboard and Mouse Conventions

SOFTIMAGE|XSI uses a three-button mouse for most operations. These are referred to as the *left*, *middle*, and *right* mouse buttons. In many cases, you will use the different buttons to perform different operations; always use the left mouse button unless otherwise stated.

- The two-button mouse is not supported in SOFTIMAGE|XSI.
- If you are using a left-handed mouse or other pointing device, substitute *primary button* for *left button*, and so on.

This table shows the terms relating to the mouse and keyboard.

When this term is used...	...it means this
Click	Quickly press and release the left mouse button. Always use the left mouse button unless otherwise stated.
Middle-click	Quickly press and release the middle mouse button of a three-button mouse.
Right-click	Quickly press and release the right mouse button.
Double-click	Quickly click the left mouse button twice.
Shift–click, Ctrl–click, Alt+click	Hold down the Shift, Ctrl, or Alt key as you click a mouse button.
Drag	Hold down the left mouse button as you move the mouse.
Alt+*key,* Ctrl+*key,* Shift+*key*	Hold down the first key as you press the second key. For example, "Press Alt+Enter" means to hold down the Alt key as you press the Enter key.

Comments?

If you have anything you'd like to say or ask about the SOFTIMAGE|XSI documentation, please do not hesitate to e-mail us at **education@softimage.com.**

Softimage Customer Service

Technical support in North America for SOFTIMAGE|XSI is provided directly from Softimage Customer Service. Immediate assistance for any technical issue is available through hotline, and through electronic and web support services.

Softimage resellers working together with Softimage Customer Service provide support worldwide.

Licensing Support

You can request your new license file through the license request form provided at **softimage.com/licensing** or by contacting your reseller directly.

SDK Support

Support for the SOFTIMAGE|XSI SDK is separate from product support for SOFTIMAGE|XSI. The SDK support program provides a private web-based discussion forum where you can talk to fellow developers and submit support questions through our web-based technical support form. The web also provides answers to frequently asked questions, source code examples and case studies. Late-breaking SDK information, workarounds, tips and techniques can also be found in the knowledge base.

Training Support

If you're interested in SOFTIMAGE|XSI training, you'll find a complete overview of courses, education centers, and training programs at **softimage.com/education**.

Hotline Support

If you've purchased a maintenance contract to receive support directly from your Softimage reseller, you'll find assistance for contacting your reseller at **softimage.com/Corporate/Sales/Buy/** In all other cases, contact Softimage Customer Service at the following numbers and during these hours:

World Wide	Hours
Tel: 1 (514) 845-2199 Fax: 1 (514) 845-8252	9 AM to 7 PM (EST) 2 PM to 12 AM (GMT) 1400 to 2400 (UTC)
North America	Hours
Tel: 1 800 387 2559 Fax: 1 (514) 845-8252	9 AM to 7 PM (EST) 2 PM to 12 AM (GMT) 1400 to 2400 (UTC)
UK and International	Hours
Tel: + 44 1 753 650 670	9 AM to 6 PM (GMT) 0900 to 1800 (UTC) 4 AM to 1 PM (EST)

Electronic Support

If you have an active maintenance contract and would like to reach us electronically, you will find a Support Request form at **http://www.softimage.com/Support/XSI/TechnicalSupport/ supportrequestform.asp**.

For general inquiries about Softimage support services, send an email to **support@softimage.com**.

Web Support

The Support and Download sections of softimage.com provide quick access to a wide range of resources from the SOFTIMAGE|XSI teams and user community. Downloads—including Presets, Scripts, and Quick Fix Engineering (QFEs)—provide the latest solutions for XSI. Online user guides, tutorials, and knowledge-base articles ensure you get the most out of working with XSI. It's like having a dedicated Softimage Customer Service engineer sitting at your desk!

Mailing Lists

If you have an e-mail account, you can join the worldwide network of SOFTIMAGE|XSI users exchanging ideas. To subscribe to the XSI discussion group, send an e-mail to **majordomo@softimage.com** with **subscribe XSI** as the body of your message. Use the message lists for information on related groups. Visit **softimage.com/Archives/Help.htm** for more information about our discussion groups and mail server.

The discussion groups are provided for technical exchanges between customers. Although customer service is not provided through these discussions, we do contribute.

Student/Teacher Program Support

If you've purchased SOFTIMAGE|XSI through the Student/Teacher program, installation support is offered by Softimage (see the support contact information above). For usage support, you are welcome to join the XSI mailing lists.

Softimage Customer Service Addresses

North America

Softimage Customer Service
3510 Saint-Laurent Boulevard
Montreal, Quebec
H2X 2V2 Canada

UK and International

Softimage Customer Service Europe
Pinewood Studios
Pinewood Road
Iver Heath, Buckinghamshire
SL0 0NH United Kingdom

Section I • Modeling & Deformation Basics

Chapter 1 Modeling in XSI®

Modeling in XSI

Modeling is the task of creating the objects that you will animate and render.

This chapter covers some fundamental concepts underlying modeling in XSI:

- The different types of geometric objects and their attributes like centers and normals.
- Some of the other useful tools for modeling and where you can find more information about them.

Geometric Objects

By definition in XSI, geometric objects have points. The set of these points and their positions determine the shape of an object and are often called the object's *geometry*. The number of points and how they are connected is called its *topology*. In addition, geometric objects have attributes like centers and normals.

Types of Geometry

SOFTIMAGE®|XSI® offers several types of geometry: polygon meshes, subdivision surfaces, curves, surfaces, and surface meshes.

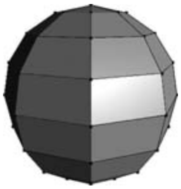
Polygon Meshes

Polygon meshes are quilts of polygons joined at their edges and vertices. Because their geometry is mathematically simple and quick to calculate, they are particularly useful when modeling for games and other real-time environments where speed is important.

Another nice feature of polygon meshes is that you can apply local materials and textures to selected polygons instead of the whole mesh or you can use vertex colors to reduce the amount of data needed to render them. You can also subdivide specific polygons in areas where you need more detail and remove them where you require less detail.

However, the main drawback of polygon meshes is that they are poor at representing organic shapes—you may require a very heavy geometry (that is, many points) to obtain smoothly curved objects.

Polygon meshes are described in more detail in *Chapter 8: Modeling Polygons & Polygon Meshes* on page 211.



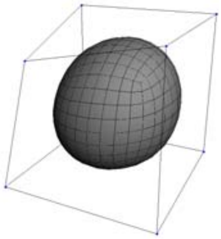
A polygon mesh sphere

Subdivision Surfaces

Subdivision surfaces consist of a low-resolution polygon mesh hull that controls a higher-resolution polygon mesh object. They provide many of the benefits of polygon meshes, plus the ability to approximate smooth surfaces without the need for heavy geometry in the control hull.

You can animate and texture the control hull for greater speed and interaction while you work, and then let the hull drive the high-resolution version for your final output.

Subdivision surfaces are described in more detail in *Chapter 9: Subdivision Surfaces* on page 325.



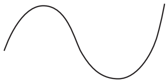
A subdivision surface created from a cube.

Curves

Curves are one-dimensional NURBS (non-uniform rational B-splines) of linear or higher degree. They have points but they are not renderable because they have no thickness.

Nevertheless, they have many uses, such as serving as the basis for constructing surfaces, paths for objects to move along, controlling deformations like deform by curve and deform by spine, and so on.

Curves are described in more detail in *Chapter 11: Curves* on page 359.



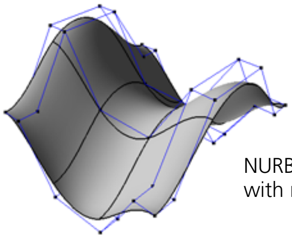
A simple cubic NURBS curve.

Surfaces

Surfaces are two-dimensional NURBS patches defined by intersecting curves in the U and V directions. In a cubic NURBS surface, the surface is mathematically interpolated between the control points, resulting in an smooth shape with relatively few control points.

The accuracy of NURBS makes them ideal for smooth, manufactured shapes like car and aeroplane bodies. One limitation of surfaces is that they are always four-sided; also, they do not support different textures on different areas.

Surfaces are described in more detail in *Chapter 12: Surfaces* on page 385.



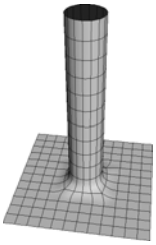
NURBS surfaces allow for smooth geometry with relatively few control points.

Surface Meshes

Surface meshes are quilts of NURBS subsurfaces acting as a single object. They overcome the limitation that surfaces must be four-sided; with surface meshes, you can create complex objects and characters with holes, legs, and so on.

You can control the continuity at the junctions between subsurfaces to create complex, yet seamless objects.

Surfaces meshes are described in more detail in *Chapter 13: Surface Meshes* on page 411.

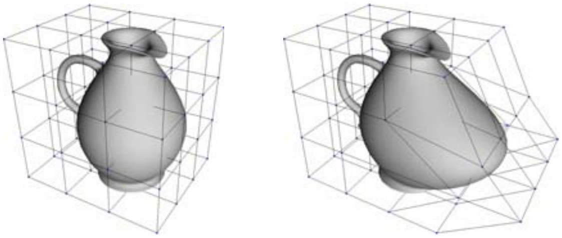


A smooth surface mesh composed of several NURBS patches.

Lattices

Lattices are a hybrid between geometric objects and control objects. Although they have points, they do not render and are used only to deform other geometric objects.

For more information, see *Lattices* on page 144.



Other Types of Geometry

There are a couple of other types of geometry in XSI:

- **Hair:** You can create hair and fur, style it, and apply dynamic simulations. See *Section III • Hair and Fur Simulations* in the *Simulation* guide.
- **Geometry Shaders:** You can use shaders to create geometry at render time. See *Chapter 18: Geometry Shaders* in the *Shaders, Lights & Cameras* guide.

- Faces and Meta-Clay™ systems are not supported. If you import these objects from SOFTIMAGE®|3D, they are converted to polygon meshes.
- Bézier, B-Spline, and Cardinal curves and patches are not supported. If you import these objects from SOFTIMAGE|3D, they are converted to NURBS curves and surfaces.

Density

Density refers to the number of points on an object. The more points there are, the more detail can be represented. However, extremely dense, "heavy" scenes can be slow to load and update.

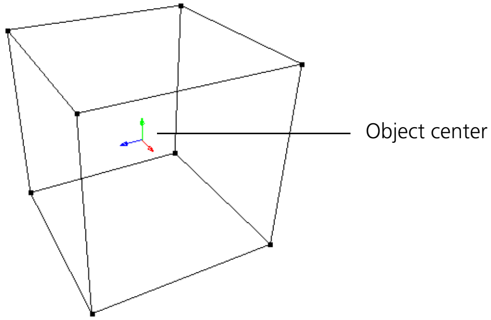
There are several approaches you can take to control the balance between density and detail. One is in the choice of geometry type: NURBS surfaces need fewer points than polygon meshes to represent smooth objects, while polygons are better for sharp details because you can add extra points only where they are needed. In addition, when getting primitives and creating other objects, you can control the number of points by specifying the number of subdivisions in each direction.

You can also create a lighter object that captures the general shape of a more complex object. This can be done in many ways, for example, by using the Shrinkwrap deformation, by cleaning or fitting surfaces, and so on.

When working with polygon meshes, you can take advantage of subdivision surfaces to work at a low level of detail, then render at a high one. For more information about subdivision surfaces, see *Chapter 9: Subdivision Surfaces* on page 325.

Centers

The 3D world has a center (called the *origin*) and each object in the 3D world has its own center. The center is the reference for defining the object's shape, location, orientation, and size. This means that, for all practical purposes, the transformations you perform on an object are really performed on the object's center.

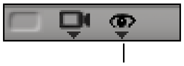


When you create an object, its (local) center is located at the origin by default. However, you can move this center to another location for performing different modeling tasks. This means you can transform objects using one of several possible coordinate systems, with very different results.

For more information about centers, see *Chapter 16: Transformations* in the *Fundamentals* guide.

Normals

On polygon meshes and surfaces, the control points form bounded areas. Normals are vectors perpendicular to these closed areas on the surface and 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.

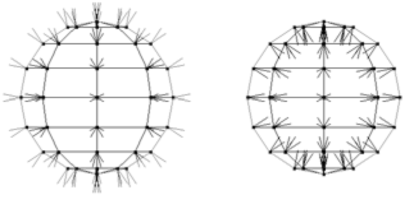


Normals are represented by thin blue lines. To display or hide them, click the eye icon (Show menu) of a 3D view and choose **Normals**.

When normals are oriented in the wrong direction, they cause modeling or rendering problems. You can invert them using **Modify** > **Surface** > **Invert Normals** or **Modify** > **Poly. Mesh** > **Invert Normals** on the Model toolbar.

If an object was generated from curves, you can also invert its normals by inverting one or more of its generator curves with **Modify** > **Curve** > **Inverse**.

Normals should point toward the camera.



Right

Wrong

You can import dotXSI files and SOFTIMAGE|3D scenes containing user-defined normals, as well as export user-defined normals to dotXSI files.

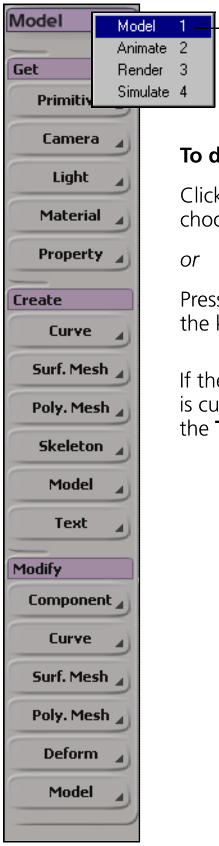
In XSI, user-defined normals are treated as user data that can be manipulated using scripted operators or compiled plug-ins. XSI does not automatically update user-defined normals when you deform objects.

Useful Tools for Modeling

All the important modeling commands can be found on the Model toolbar. In addition, there are many other tools and features that you will find very useful while modeling.

Model Toolbar

You'll find the Model toolbar at the far left of the screen. These commands are also available from the main menu.



Get commands
Create generic elements, including primitive objects, cameras, and lights (also available on Animate, Render, and Simulate toolbars).

Create commands
Draw new objects or generate them from existing ones.

Modify commands
Change an object's topology or deform its geometry.

To display the Model toolbar

Click the toolbar title and choose **Model**.

or

Press **1** at the top of the keyboard.

If the Palette, Paint, or Hair panel is currently displayed, first click the **Toolbar** icon or press Ctrl+1.

Context Menus

Many modeling commands are available from context menus. The context menu appears when you Alt+right-click in the 3D views (Ctrl+Alt+right-click on Linux).

- If you click a selected object, the menu items apply to all selected objects. On Windows, you can also press the context-menu key (next to the right Ctrl key on some keyboards).
- If you click an unselected object, the menu items apply only to that object.

- When components are selected, you can Alt+right-click anywhere on the object that “owns” the selected components. The items on the context menu apply to the selected components.
- If you click over an empty area of a 3D view, the menu items apply to the view itself.

Views

XSI includes several different views that can help you with your modeling tasks.

Explorer

The explorer displays the contents of your scene in a hierarchical structure. When modeling, it is particularly useful for seeing an object’s construction history, or *geometry operator stack*.

- For more information about the explorer, see *The Explorer* in Chapter 4 of the *Fundamentals* guide.
- For more information about the operator stack, see *Operator Stack* on page 51.

Object Viewer

The object viewer is a 3D view that displays only the selected scene elements. It is useful for isolating a specific object when modeling. For more information, see *The Object View* in Chapter 5 of the *Fundamentals* guide.

Schematic

The schematic view presents the scene as connected nodes in a graph so you can analyze the way a scene is constructed. It can display graphical links that show the modeling relationships between objects. For more information, see *The Schematic View* in Chapter 4 of the *Fundamentals* guide.

XSI Explorer

The XSI explorer is a view that combines an explorer tree for selecting objects and a viewing pane that can show other views, such as an object view or a schematic view. For more information, see *The XSI Explorer* in Chapter 4 of the *Fundamentals* guide.

Spreadsheet

The spreadsheet lets you execute queries and displays scene information about elements and their parameters in a grid. For more information, see *The Spreadsheet* in Chapter 4 of the *Fundamentals* guide.

Net View

Net View is an integrated Web browser within XSI. You can use it to access sample scenes and gift tools installed with XSI as well as those available on **http://www.softimage.com/xsinet**. You can also create your own intranet for sharing resources across your workgroup.

- For information about using Net View, see *Net View* in Chapter 4 of the *Fundamentals* guide.
- For information about building your own intranet for use in XSI, see *Chapter 13: Intranets for Net View* of the *Fundamentals* guide.

Duplicating and Cloning

You can quickly build objects with similar parts by duplicating and cloning objects. A duplicate is simply a copy of another object, while a clone is a “linked” copy—you can modify all instances by modifying the original object. For more information, see *Duplicating and Cloning Objects* in Chapter 9 of the *Fundamentals* guide.

Symmetry

To make a symmetrical copy of an object or hierarchy, use **Edit** > **Duplicate Symmetry**. For more information about duplicating, see *Duplicating and Cloning Objects* in Chapter 9 of the *Fundamentals* guide.

There are also other symmetry features in XSI:

- You can create symmetrical copies while drawing curves and chains (see *Drawing Curves Symmetrically* on page 368).
- You can maintain symmetry while manipulating points (see *Manipulating Components Symmetrically* on page 117).
- You can paint weight maps symmetrically (see *Weight Maps and Symmetry* in Chapter 11 of the *Fundamentals* guide).
- You can mirror envelope weights (see *Mirroring Envelope Weights* in Chapter 3 of the *Character Animation* guide).
- You can also apply a symmetry constraint for transformations (see *Symmetry Constraints* in Chapter 8 of the *Animation* guide).

Transformations

As you model objects, you will be moving them around; that is, translating, rotating, and scaling them and their components. For more information about transformations in general, see *Chapter 16: Transformations* of the *Fundamentals* guide.

Clipping Planes

When you are working on complex models, you sometimes want to focus on a specific part. For example, you may want to hide the backside of an object while you are moving points on the front.

To do this, you can set a camera’s clipping planes as described in *Setting Clipping Planes* in Chapter 14 of the *Shaders, Lights & Cameras* guide.

Rotoscopy

You can use rotoscopy to display background plates in the 3D views. This lets you use images as guides when modeling. For more information about rotoscopy, see *Chapter 6: Rotoscopy* of the *Fundamentals* guide.

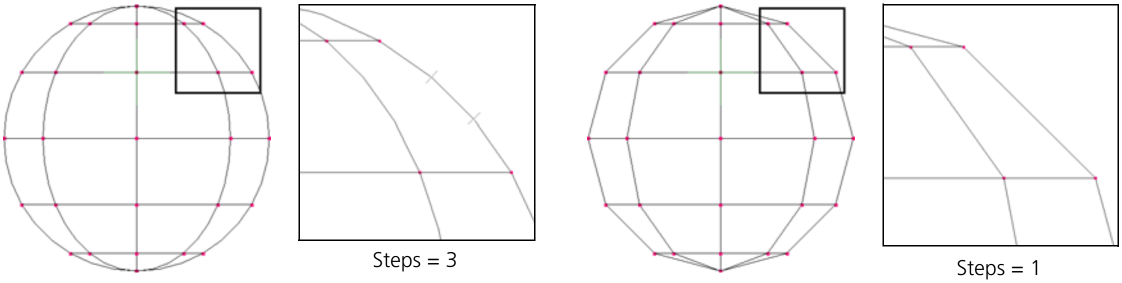
Geometry Approximation Parameters

The geometry approximation parameters control how the geometries of objects are approximated—the number of steps drawn per curve segment and so on.

By default, a new scene defines the geometric-approximation parameters for the Scene Root. This property is branch-propagated to all objects in the scene.

The Hardware Display parameters define the settings used for the OpenGL display in 3D views, while the Surface, Surface Trim, Polygon Mesh, and Displacement parameters define the settings used for rendering.

For more information about geometry approximation, see *Viewing and Rendering Polygon Meshes* on page 196 and *Geometry Approximation on NURBS* on page 356.



Geometric approximation: Setting the number of steps between knots on a surface

Visibility, Display, and Selectability

Each object has visibility, display, and selectability parameters that can be set independently of the camera settings for the 3D views.

- The visibility parameters determine whether the object appears in the 3D views and when rendering.
- The display parameters control how the object appears as you select and modify it.
- The selectability parameter controls whether you can select or pick the object in a 3D view. You can always select objects in the Explorer no matter what the value of its selectability parameter.

These parameters are propagated through hierarchies; an object can either inherit its parent’s parameters or use its own local values. The global values for a scene are stored in the Scene_Root model. By changing these parameters, you can work faster by simplifying the tasks of viewing, selecting, and navigating around your scene.

For more information, see *Setting Object Visibility* in Chapter 5, *Setting Object Display* in Chapter 5, and *Defining Selectability* in Chapter 8 of the *Fundamentals* guide.

Hierarchies

To organize the objects of a scene into a hierarchy, you parent one element to another—this lets you propagate properties down the hierarchy from the root element.

You can select an entire hierarchy at once in tree mode by right-clicking on any member of the hierarchy or select an individual branch by middle-clicking on the top node of the branch you want. For example, when you select an object in branch mode and translate it, its children are translated with it.

You can create hierarchies using the **Parent** button or by dragging and dropping nodes in the explorer. For more information about hierarchies in general, see *Chapter 13: Hierarchies* of the *Fundamentals* guide.

Layers

Layers are a useful tool for organizing your scene. You can divide your scene into layers and control the visibility, display, and selectability of all objects in a layer at once. For more information, see *Chapter 14: Layers* of the *Fundamentals* guide.

Info Selection

Select an object and choose **Edit** > **Info Selection** or press Shift+Enter to get useful information about an object: name, type of geometry, number of components, number of triangles when tesselated for rendering, and so on.

Info Geometry

Selecting an object and choosing **Edit** > **Info Geometry** opens a page in Netview showing details about the object's points and polygons (faces). If you select another object while the page is open, click Update to refresh the page with information about the newly selected object.

Chapter 2 Starting From Scratch: Primitives and Text

Starting From Scratch

When modeling, you need to start with a geometric object. Two of the basic ways to do this are:

- Getting a primitive object, setting its size, subdivisions, and other parameters.
- Creating text and converting it to curves or polygon meshes.

In addition to what is described in this chapter, there are some other ways of getting a geometric object that are described elsewhere:

- You can draw polygons as described in *Adding and Editing Polygons (Drawing)* on page 214.
- You can draw curves as described in *Drawing Curves* on page 366.
- You can also import 3D objects from SOFTIMAGE|3D and other software as described in *Chapter 25: Considerations for Importing SOFTIMAGE|3D Scenes* and *Chapter 24: Importing and Exporting* in the *Fundamentals* guide.

Primitives

Primitives are basic shapes and other objects that you can add to a scene. These primitive shapes can form the basis for more complex objects. For example, you can start with a sphere and move points to create a head. You can then attach eyeballs and ears to the head and put the whole head on a model of a character.

There are several different primitive shapes for each geometry type. Each primitive shape has parameters that are particular to it—for example, a sphere has a radius that you can specify, a cube has a length, a cylinder has both height and radius, and so on.

There are also several parameters that are common to all or to several primitive shapes: **Subdivisions**, **Start** and **End Angles**, and **Close End**.

Getting Primitives

You add a primitive object to the scene by choosing an option from the **Get** > **Primitive** menu on any of the toolbars at the left of the main window.

1. Choose **Get** > **Primitive**.
2. Choose an item from the menu:
   - **Null** creates an object that has little more than a location in space. Nulls have many uses, such as for setting constraints and organizing objects in hierarchies.
   - **Curve** displays a submenu from which you can choose an available NURBS curve shape. For more information about modeling with curves, see *Chapter 11: Curves* on page 359.
   - **Polygon Mesh** displays a submenu from which you can choose an available polygon mesh shape. For more information about modeling with polygon meshes, see *Chapter 8: Modeling Polygons & Polygon Meshes* on page 211.
   - **Surface** displays a submenu from which you can choose an available NURBS surface shape. For more information about modeling with surfaces, see *Chapter 12: Surfaces* on page 385.
   - **Implicit** displays a submenu from which you can choose an available shape.

     Implicits are basic shapes defined by a mathematical formula. They are not renderable but can be used, for example, as constraints, bounding volumes, control objects for a character rig, and so on.

     You can scale, rotate, and translate implicit objects, but you cannot deform them because they have no points. Note that implicit objects are not exported to the IGES format.

   - **Control Object** displays a submenu from which you can choose an object that can be used to deform or control other objects.

     Control objects include waves and forces like gravity and wind. They do not render; instead, they exist to define the properties of deformations and other effects.

     For more information about wave deformations, see *Waves* on page 171. For more information about physical forces, see the *Simulation* guide.

   - **Lattice** creates a three-dimensional scaffolding that can be used to deform other objects. For more information about lattice deformations, see *Lattices* on page 144.

   - **Model** displays a submenu from which you can choose one of several predefined, complex models. For more information about the geometries, skeletons, and rigs available on this submenu, see the *Character Animation* guide.

   If applicable, a property editor opens and displays parameters for the chosen primitive.

3. Set the parameters as desired. The geometric primitives (curves, polygon meshes, and surfaces) have certain typical controls:

   - The shape-specific page contains the basic characteristics of the shape. Each shape has different characteristics; for example, a sphere has one radius and a torus has two.

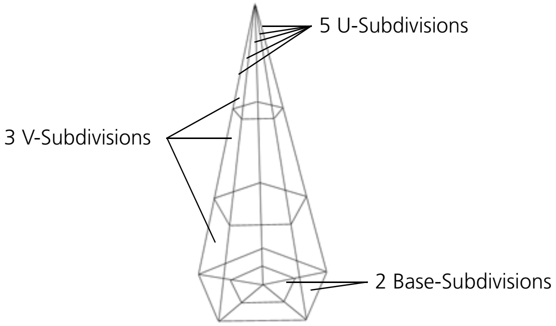
   - The Geometry page controls how the implicit shape is subdivided when converted into a surface. More subdivisions yield more points, resulting in greater detail but heavier geometry.

   For more information about a particular parameter, select the corresponding tab, then refer to online help by clicking the ? button. Note that some primitives, like the rhombicosidodecahedron and the soccer ball, have no parameters that you can modify other than the object’s name.

Once you have added a geometric primitive to your scene, you can modify it like any other.

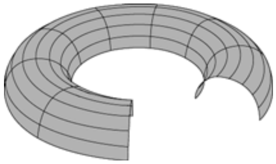
U, V, and Base Subdivisions

When you get a primitive object, you can specify the number of subdivisions. More subdivisions result in more points, thereby giving a heavier geometry with more detail. Two-dimensional primitives like grids let you specify the number of subdivisions in the U and V directions, while three-dimensional primitives let you specify the number for the base as well.



Start and End Angles

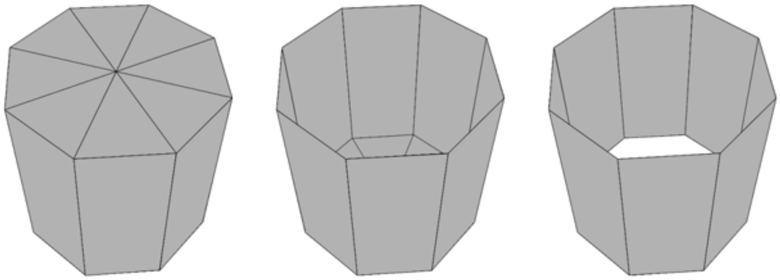
Many primitive objects have **Extent** settings that let you set the **Start** and **End** angles. You can use these options to open up the basic primitive shapes, for example, to create a hemisphere or take a slice out of a disk.



Torus open in both the U and V directions

Open and Closed Ends

Several primitives have one or more **Close** options that can be used to cap their ends. By default these options are on, but you can turn them off to open the ends.



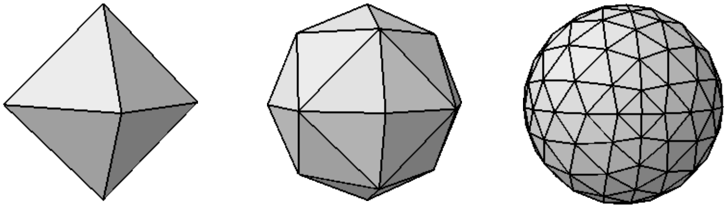
Cylinder with ends closed Cylinder with top end open Cylinder with both ends open

Geodesics

Icosahedra, octahedra, and tetrahedra have geodesics parameters that control how they are subdivided:

- **Geodesic Frequency** controls the number of subdivisions in each of the original edges.

Octahedron with different Geodesic Frequency values

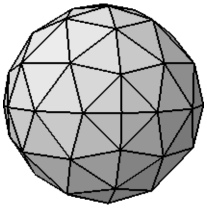


Geodesic Frequency = 1

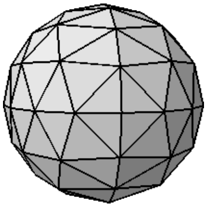
Geodesic Frequency = 2

Geodesic Frequency = 6

- **Subdivision Method** controls how the edges are subdivided: **Linear** uses equilinear subdivision and **Angle** uses equiangular subdivision.



Linear

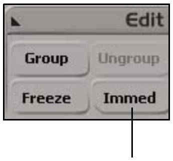


Angle

Creating Text

You can create text in XSI, as well as import it from RTF (rich text format) files. Text is not a type of geometric object in XSI; instead, text information is immediately converted to curves. After that, the curves can be optionally converted to planar or extruded polygon meshes.

Overview of Creating Text



Immediate mode off

Here is a quick overview of creating text. More detail is provided in the sections that follow.

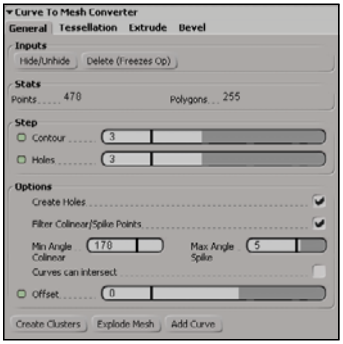
1. Make sure that **Immed** (immediate mode) is off on the Edit panel.

   In immediate mode, each operation is frozen as soon as it is applied, so you wouldn't be able to type any text after creating the text object.

2. Choose one of the following commands from the Model toolbar:

   - **Create** > **Text** > **Curves** creates a Text primitive and converts it to a curve object.
   - **Create** > **Text** > **Planar Mesh** creates a Text primitive, converts it to a curve object, and then finally converts the curve to a polygon mesh with the Extrusion Length set to 0. The curve object is automatically hidden.
   - **Create** > **Text** > **Solid Mesh** creates a Text primitive, converts it to a curve object, and then finally converts the curve to a polygon mesh with the Extrusion Length set to 0.5 by default. Once again, the curve object is automatically hidden.

In each case, a property editor with the following pages is displayed:



Stats

Points 470

Polygons 255

Step

Contour 3

Holes 3

Options

Create Holes ☑

Filter Colinear/Spike Points ☑

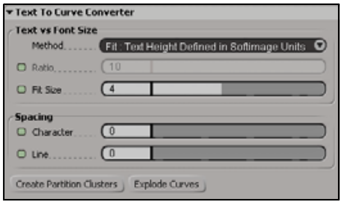
Min Angle Colinear 170

Max Angle Spike 5

Curves can intersect □

Offset 0

Convert curves to polygon meshes (optional).



Text vs Font Size

Method Fit : Text Height Defined in Softimage Units

Ratio 10

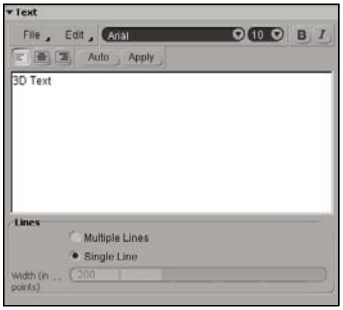
Fit Size 4

Spacing

Character 0

Line 0

Convert text to curves.



Arial 10

3D Text

Lines

○ Multiple Lines

● Single Line

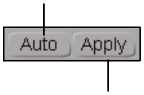
Width (in points) 200

Enter text and font properties.

Typing Text

You can type text into the editing pane of the Text property page in the same way as you would type text in any simple word processor. Use the items on the Edit menu to undo, cut, copy, paste, or clear.

Automatically update 3D views as you type



Click to update 3D views manually

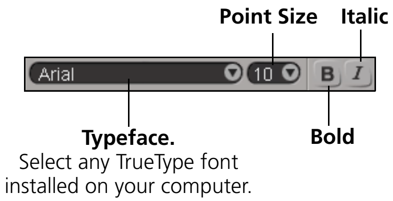
As you type, the text in the scene may update automatically depending on whether **Auto** is on or off:

- If Auto is on, the 3D views update automatically as you type. Note that there is a small delay before the scene updates, to make sure that you have finished typing.
- If Auto is off, the 3D views do not update automatically; instead, click **Apply** to see the result of your changes.

  By leaving Auto off, you avoid delays that may occur as XSI tries to update heavy geometry as you work, particularly when there are extrusions and bevels.

Setting Font Attributes

Font attributes include the typeface, size, weight (bold or regular), and angle (italic or roman). As with most word processors, you can change the attributes for selected characters or for new characters. New characters use the formatting of the previous character unless you change the font attributes explicitly.



Select any TrueType font installed on your computer.

Make sure that the font you select is installed on all machines on which you intend to work with or render the scene. Otherwise, the typeface is automatically replaced by a default if you have not frozen the operator stack.

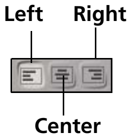
Installing Fonts on Linux

SOFTIMAGE|XSI uses the fonts installed in your $MWHOME/fonts directory. To install a TrueType or OpenType font, simply copy the TTF file to this location. If there is a corresponding BDF file, copy it to the same location for a “what you see is what you get” display in the editing pane of the Text property editor.

Setting Line Attributes

You can control how lines are aligned, as well as how multiple lines are treated.

Line Alignment



The line alignment options control how the subcurves are aligned with the 3D object’s center.

If **Multiple Lines** is on, you can set these options on individual lines. The first line determines how the curves are aligned with the 3D object’s center. The relative position of the other lines is determined by the **Width**.

Single versus Multiple Lines

The Lines options determine how multiple text lines are treated:

- If **Single Line** is on, curves are created on a single line in XSI. Word-wrapping in the editing pane is ignored. Any text after a line break or paragraph return is also ignored (and is not displayed in the editing pane). Pressing Enter is equivalent to clicking **Apply**.
- If **Multiple Lines** is on, curves are created on multiple lines according to the line breaks, paragraph returns, and word-wrapping shown in the editing pane.

You can specify a **Width** in points for word-wrapping and line alignment; the corresponding width in the scene is controlled by the **Text vs Font Size** options in the Text to Curve Converter—see *Controlling Curve Size* on page 45.

Importing RTF Files

You can import text saved as an RTF (rich text format) file. After you import an RTF file, you can still modify the text in the editing pane. XSI does not reference the RTF file; modifying the original RTF with a word processor does not affect the scene.

Not all attributes of the rich text format are supported. Any formatting that is not supported is ignored.

To import an RTF file

1. Choose one of the commands on the **Create** > **Text** menu of the Model toolbar.
2. On the Text property page, choose **File** > **Open.**
3. Use the Open File browser to find and select an RTF file.

If character formatting is stored in the paragraph character at the end of a line, you must turn on **Multiple Lines** to see the effect.

Saving RTF Files

You can save the contents of the editing pane as an RTF file to use in another application or import into a different scene.

To save an RTF file

Choose one of the following commands from the Text property editor's command bar:

- **File** > **Save**. If you have previously saved the contents of the editing pane, it is saved again with the same name and the previous version is overwritten on disk. Otherwise, a browser opens for you to specify a path and file name.
- **File** > **Save As**. A browser opens for you to specify a path and file name.

Controlling Conversion to Curves

The parameters in the Text to Curve Converter property page control the size and the spacing of the curves. In addition, you can create partition clusters for each character.

Curve objects created from text typically contain multiple subcurves. Because of this, there are certain limitations. In particular, some of the **Create** commands on the Modeling toolbar do not work with such curve objects. For example if you extrude such a curve along an axis, only the first subcurve is extruded.

Controlling Curve Size

There are two methods for controlling the size of curves:

- **Fit: Text Height Defined in Softimage Units** scales the entire text block (including multiple lines) to the height specified by **Fit Size**. The relative size of characters is preserved.
- **Ratio: Points per Softimage Unit** sets the size of individual curves based on the point size of the corresponding characters. Use the **Ratio** parameter to adjust the number of points per Softimage unit.

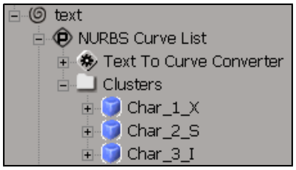
Controlling Spacing

The options in the **Spacing** box allow you to compress or expand the spacing between characters and lines.

Creating Partition Clusters

You can automatically create one subcurve cluster per character. This makes it easier, for example, to select subcurve clusters and extract new curve objects using **Create** > **Curve** > **Extract from Subcurves** to make separate curve objects for each character.

The clusters are automatically named based on the character position and the character itself, as shown in the illustration below:



```
⊟ text
   ⊟ NURBS Curve List
      ⊞ Text To Curve Converter
      ⊟ Clusters
         ⊞ Char_1_X
         ⊞ Char_2_S
         ⊞ Char_3_I
```

If you change the text after creating partition clusters, they are not updated automatically. You must re-create the partition clusters again.

To create partition clusters

- Choose **Create Partition Clusters** on the Text to Curve Converter property editor.

Exploding into Separate Curve Objects

When converting text to curves, you can create a hierarchy of separate curve objects. There is one new object for each character, and all objects are parented under a common null.

To explode text into a hierarchy of curves

- Choose **Explode Curves** on the Text to Curve Converter property editor.

  Each character is automatically extracted and parented under a new null object. The null object is selected in branch mode.

  A modeling relation exists between the unexploded curve and the extracted objects, so modifying the options like Spacing on the Text to Curve Converter property editor affects the extracted objects as well.

  However if you change the text after exploding the curves, you should re-explode them.

Controlling Conversion to Polygon Meshes

The options for controlling how text is converted to planar or solid polygon meshes are exactly the same as for converting any curve to a polygon mesh. This procedure is described in full detail in *Converting Curves to Polygon Meshes* on page 306.

Chapter 3 Building 3D Objects

Building 3D Objects

This chapter describes some of the basic concepts and procedures that underpin modeling in SOFTIMAGE|XSI.

- The operator stack is fundamental. It is the history of all operations you have applied to an object, and is divided into construction regions. You can go back and change parameters in an operator, change the order of operators, or freeze all or part of the stack.
- Modeling relations exist between objects in your scene. If you create a surface from a series of curves, you can modify a curve and the changes are automatically reflected in the surface.
- Immediate mode lets you model without creating a history or modeling relations. It is equivalent to freezing an object's operator stack after every operation.

The Process of Modeling

Modeling is the task of creating the objects that you will animate and render. This section provides an overview of the process as well as a couple of the concepts you will encounter when modeling in XSI.

Overview of Modeling

To model, you start with an initial object and you modify it in some way.

Getting the Initial Object

You can create an initial object in any of the following ways:

- Getting a primitive curve, surface, or polygon mesh object from the **Get** > **Primitive** menu.
- Creating text.
- Drawing a curve or polygon.
- Building an object from others in your scene, for example, revolving a profile curve to create a surface.
- Importing an object from SOFTIMAGE|3D or another software package.

Modifying the Initial Object

After you have an initial object, you can modify it. For each modification, an operator is added to the object's operator stack. Deformation operators change the shape of the object by moving points in space, while topology operators add or remove components like points, edges, and polygons.

When you are satisfied with your object, you can freeze it; this removes the accumulated operator stack, thus simplifying your scene data and improving performance.

Understanding the operator stack is fundamental to modeling in XSI. It is a major difference between XSI and SOFTIMAGE|3D. For more information, see *Operator Stack* on page 51.

Picking

Many operations require that you specify one or more elements as inputs. For example, to create a Loft surface you need to specify a series of profile curves. The action of specifying inputs is called *picking*.

The process of picking elements is standard for all operations. You can:

- Select the inputs first and then choose a command.

  or

- Choose a command and then pick the inputs.

  or

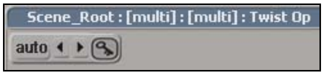
- Select some inputs, then choose a command and pick the remaining inputs.

Some operations require that the inputs be picked in a specific order. For example, to revolve a curve around another curve, you must first pick the curve to be revolved and then pick the curve to revolve around.

In most cases, when an object is an input for an operation, a modeling relation exists between the input and the output. For more information, see *Modeling Relations* on page 66.

Modifying Multiple Objects

You can select multiple objects for many operations. For example, you can select several surfaces and apply a Twist deformation. Each surface gets its own Twist operator, but they can all be controlled from the same Twist property editor—the title bar says “multi” to indicate a multiple selection.


Scene_Root : [multi] : [multi] : Twist Op

Note that you can’t do this with operators that have multiple inputs because it is ambiguous which inputs should be used for which operator. For example, Loft allows you to specify multiple curves for the profiles—you can’t create two loft surfaces at the same time because it is not clear which curves should be used for each surface.

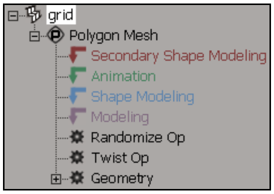
Operator Stack

The operator stack (also known as *modifier stack*) is fundamental to modeling in SOFTIMAGE|XSI. It is the history of all the operators (such as deformations and topology modifications) that have been applied to an object, and is divided into regions for modeling, animation, and shapes. Operators propagate their effects upwards through the stack, with the output of one operator being the input of the next. At any time you can go back and modify or delete operators in the stack.

The operator stack is maintained only if immediate mode is off. For more information, see *Immediate Mode* on page 71.

Viewing and Modifying Operators

You can view the operator stack of an object in an explorer if **Operators** is active in the Filters menu. The operator stack is under the first subnode of an object in the explorer, typically named **Polygon Mesh**, **NURBS Surface Mesh**, **NURBS Curve List**, **Null**, and so on.



For example, suppose you get a primitive polygon mesh grid, apply a twist, then randomize the surface. The operator stack shows the operators that have been applied. You can open the property page of any operator and modify values. Any changes you make are passed up through the history and reflected in the final object.

For example, you can:

- Change the size of the grid.
- Change the number of subdivisions.
- Change the angle, offset, and axis of the twist in Twist Op.
- Change the random displacement parameters in Randomize Op.

Opening Property Editors

You can display the property editor for any operator in the stack and modify its parameters.

Some operators, such as MoveComponent, do not have associated property editors because they have no editable parameters.

To open the property editor of a single operator

You can open the property editor of a specific operator using either an explorer view or the transient explorer available from the **Selection** button in the main command area.

- If you are using an explorer view, make sure that **Operators** is active in the Filters menu. The object with the operator does not need to be selected.

- If you are using the **Selection** button, you must first select the object with the operator.

In either case, do one of the following:

- Click the icon of an operator.

  or

- Select an operator and press Enter.
  - If a property editor is open and the object that “owns” that property is not selected, then by default the object is displayed in light blue in wireframe mode.
  - If an unselected object inherits the property through propagation, it is displayed in pink.

To open the property editor of the last operator applied to an object

1. Select the object.
2. Press Ctrl+End or choose **Edit** > **Properties** > **Last Operator in Stack**.

To open the property editors of all modeling operators applied to an object

Do one of the following:

- In a schematic or explorer view, right-click an object and choose **Modeling Properties**. The object does not need to be selected.

  or

- In a 3D view, Alt+right-click (or Ctrl+Alt+right-click on Linux) on an object and choose **Modeling Properties**. Again, the object does not need to be selected.

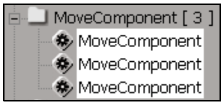
  or

- Select the object then choose **Edit** > **Properties** > **Modeling Properties** from the Edit panel.

A single property editor opens with each operator’s parameters available on separate tabs.

Grouped Operators

If you apply the same operation to an object several times in a row, the multiple occurrences of the same operator are automatically grouped together under an expandable folder in the stack and the number of occurrences is shown in square brackets.


MoveComponent [3]
MoveComponent
MoveComponent
MoveComponent

This simplifies the view in the explorer—for example, you see a single MoveComponent folder node instead of hundreds of MoveComponent nodes. Clicking on the folder name selects all the operators in the folder.

Deleting Operators

To delete an operator in the stack, select it and press Delete.

Operator Classification

Operators are divided into several categories, based on the type of information they change:

- **Generators** create an object. These include most of the operators available from the **Get** > **Primitive** menu, as well as those under **Create**. Examples include Geometry (primitive), CurveCreation, Loft, NURBS to Mesh, and so on.
- **Topology operators** change the geometry, for example, by adding or removing components. These include most of the operators available under **Modify**. Examples include Curve Shift, Insert Surface Knot, Subdivide Polygon, and so on.
- **Deformations** modify the shape of an object by changing the positions of its points. These include the operators available from the **Deform** menu, as well as the Move Component and Envelope operators.

Construction Modes and Regions

The construction history is divided into four regions: **Modeling**, **Shape Modeling**, **Animation**, and **Secondary Shape Modeling**. The purpose of these regions is to keep the construction history clean and well ordered by allowing you to classify operators according to how you intend to use them. For example, when you apply a deformation, you might be building the object's basic geometry (Modeling), or creating a shape key for use with shape animation (Shape Modeling), or creating an animated effect (Animation), or creating a shape key to tweak an enveloped object (Secondary Shape Modeling).

You add operators to specific regions by selecting a construction mode before you perform a modeling operation, as described in *Setting Construction Modes* on page 56. If you unintentionally added an operator in the wrong mode, you can move the operator to the correct region as described in *Changing the Order of Operators* on page 60.

As you work, you can display the final result (the result of all operators in all regions) or the just the current mode (the result of all operators in the current region and those below it) as described in *Displaying the Results of Different Regions* on page 57.

By default, each view shows information about the current construction mode and display mode. You can toggle this feedback on or off as described in *Getting Feedback on the Current Modes* on page 58.

About the Construction Regions

Each of the four construction regions is intended for a different purpose:

- The **Modeling** region is at the bottom. You can use this region to define the basic shape and topology of an object using both deformations and topology operators.

  This is often the region you want to freeze for performance reasons. You can freeze the Modeling region without affecting the other regions as described in *Freezing the Operator Stack* on page 63.

  You can include animated operators in the Modeling region, if desired. However, the animation will be lost if you freeze the modeling. In addition, if there is shape animation above an animated deformation in the stack, there is a danger that shapes will "pop" when polygons intercollide.

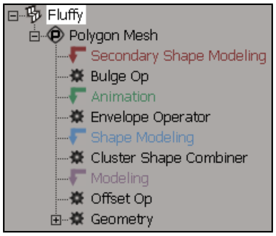
- The **Shape Modeling** region is above the Modeling region. In addition to containing the Cluster Shape Combiner node, this region lets you apply deformations to define shape keys for shape animation. When you save or store a shape key in Shape Modeling mode, the effect of the operators in this region are stored in the shape key. When you create a shape clip by saving or applying a key, only the operators in this region are removed—this protects animated deformations that may be present in the Animation region, such as envelopes, from being removed.

- The **Animation** region is above the Shape Modeling region. You can use it for animated deformations that you don't want to be removed when you freeze the modeling or apply a shape, for example, deformations such as envelopes, waves, cages, lattices, spines, or even a simple Push with an animated amplitude.

- The **Secondary Shape Modeling** region is at the top. You can use it to create shape keys for things like adding muscle bulges on top of an enveloped character. Like the Shape Modeling region, the effect of the operators in this region are stored when you save or store a shape key in Secondary Shape Modeling mode, and the operators in this region are removed when you create a shape clip by saving or applying a key.

  Note that all shape keys are applied at the level of the Cluster Shape Combiner at the bottom of the Shape Modeling region, including shapes created in Secondary Shape Modeling mode. This means that XSI must calculate the inverse deformation of any operators in the Animation region when it creates the shape key. As a result, shapes work well if there is only an envelope operator in the Animation region, but they might not always give the desired results when there are non-envelope deformations in the Animation region. This is unavoidable because not every deformation has an inverse.

In explorers, the tops of each region are indicated by special markers that act as placeholders. The markers can be moved like other operators as described in *Changing the Order of Operators* on page 60. However, you cannot change the order of the markers by moving one marker above or below another one. In addition, the Cluster Shape Combiner must always be at the bottom of the Shape Modeling region.



```
Fluffy
  Polygon Mesh
    Secondary Shape Modeling
    Bulge Op
    Animation
    Envelope Operator
    Shape Modeling
    Cluster Shape Combiner
    Modeling
    Offset Op
    Geometry
```

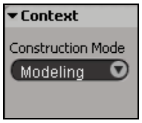
When you open or import a scene or model created in SOFTIMAGE|XSI version 3.5 or earlier:

- If an object has shape animation, the position of the Cluster Shape Combiner node determines the start of the Shape Modeling region.
- If an object has no shape animation, all operators are placed in the Modeling region.

You can then move operators to specific regions as described in *Changing the Order of Operators* on page 60.

Using Construction Modes

Here is a quick overview of the workflow for using construction modes:



Context

Construction Mode: Modeling

1. Set the construction mode as described in *Setting Construction Modes* on page 56.
2. Continue modeling objects by applying new operators.

   New deformations are applied at the top of the current region, and new topology modifiers are always applied at the top of the Modeling region.

   You can change operators' regions after they have been applied by moving operators in the stack, but the usual caveats still hold (see *Changing the Order of Operators* on page 60).
3. At any time as you work, you can display the final result (the result of all operators in all regions) or the just the current mode (the result of all operators in the current region and those below it) as described in *Displaying the Results of Different Regions* on page 57. You can even have different displays in different views so, for example, you can see and move points in one view in Modeling mode while you see the results after enveloping and other deformations in another view.

By default, each view shows information about the current construction mode and display mode. You can toggle this feedback on or off as described in *Getting Feedback on the Current Modes* on page 58.

Setting Construction Modes

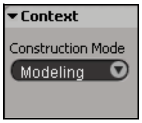
The current construction mode determines the construction region to which new deformation operators are added. You can choose the mode based on what you want to do.

For example, suppose that you want to apply a deformation by spine:

- If you are sculpting an object's basic shape with the spine deformation and intend to freeze it when you are finished, use Modeling mode.
- If you will be using the spine deformation to store shape keys for the basic shape, use Shape Modeling mode.
- If you will be animating the object by animating the spine curve's shape, use Animation mode.
- If you will be using the spine deformation to store shape keys such as muscle bulges on top of an enveloped object (that is, in relation to the envelope's animation), use Secondary Shape Modeling mode.

Topology operators are always added at the top of the Modeling region, no matter what construction mode is current.

To set the construction mode



▾ Context

Construction Mode

Modeling

- Choose the desired **Construction Mode** on the Context panel below the toolbar on the left of the interface.

  New deformations will be added to the top of the corresponding construction region until you change modes again.

To change construction mode automatically

The **Construction Mode - Enable Auto Update** option in the Modeling preferences controls whether the construction mode changes automatically when you perform certain operations. When this option is on, the following changes happen automatically:

- Applying a topology operator automatically switches to Modeling mode.
- Applying an envelope and responding "Yes" when prompted to apply it in the Animation region automatically switches to Animation mode.
- Saving or storing a shape key automatically switches to Shape Modeling mode when you are in Modeling mode, or Secondary Shape Modeling mode when you are in Animation mode.

Displaying the Results of Different Regions

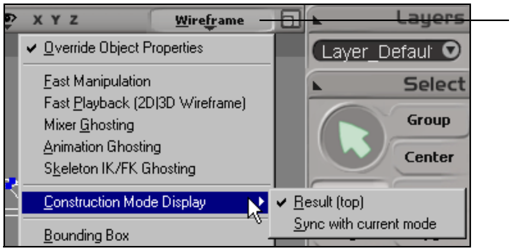
The construction display mode determines how objects are displayed in 3D views. The possible values are:

- **Result (top)** always shows the final result of all operators, no matter which construction mode is current.
- **Sync with construction mode** shows the result of the operators in the current construction region and below.

You can continue to apply deformation and topology operators no matter which region is displayed. You can even have different displays in different views so, for example, you can see and move points in one view in Modeling mode while you see the results after enveloping and other deformations in another view.

To set the construction display mode for a view

- Do one of the following:
  - From the view's Display menu, choose the desired option from the **Construction Mode Display** submenu.



or

  - From the view's Display menu, choose **Display Options**. On the Display Mode tab of the Camera Display property editor, set **Construction Level Viewing Mode** to the desired value.

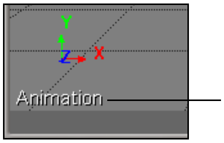
To set the construction display mode for all open views

1. Choose **View** > **Display Options** (**All Cameras**) from the main menu.
2. On the Display Mode tab of the Camera Display property editor, set **Construction Level Viewing Mode** to the desired value.

Getting Feedback on the Current Modes

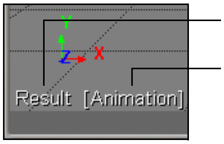
You can see information about the current construction mode and display mode in the 3D views. You can toggle this feedback on or off as desired.

- If the construction mode and display mode are the same, the mode is displayed in the lower left corner of a view.



Construction mode and display mode are the same.

- If the construction mode and display modes are different, the display mode is shown first and the construction mode is displayed in square brackets.



Display mode

Construction mode

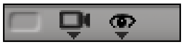
To toggle the construction mode feedback for a single 3D view

1. Open the Camera Visibility property editor for a particular view by doing one of the following:

   - Click the eye icon to open the Show menu, and choose **Visibility Options**.

     or

   - Press Shift+s while the mouse pointer is over the view.



Eye icon

2. On the Stats tab of the of the Camera Visibility property editor, toggle **Show Construction Level**.

To toggle the construction mode feedback for all open 3D views

1. Choose **View** > **Visibility Options** (**All Cameras**) from the main menu.
2. On the Stats tab of the of the Camera Visibility property editor, toggle **Show Construction Level**.

Special Considerations for Topology Modifications

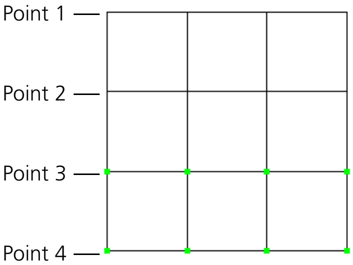
There are a few things you should be aware of when there are operators that modify the topology in the stack.

Animating or Modifying Generators and Topology Operators

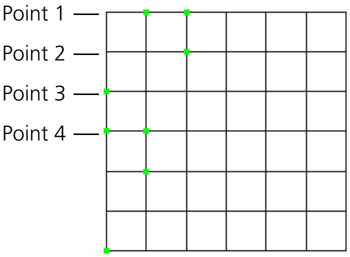
When a generator or topology operator exists below other operators in the stack, you must be very careful when modifying or animating its parameters. If the amount of geometry changes, then the indices of components will also change. This can have an adverse affect on other operators higher in the stack if they rely on clusters or component indices.

For example, suppose you get a primitive polygon mesh grid and then create a cluster on it. If you reopen the primitive Geometry property editor and increase the Subdivisions, the cluster shifts because the new points are being added at a point in the stack earlier than where the cluster is defined—this also affects any operator such as MoveComponent that may use the cluster as an input.

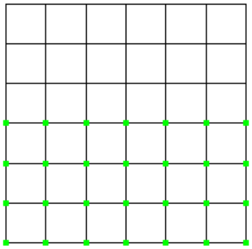
However, if you apply a SubdividePolygon operator after you create the cluster, the cluster does not shift because the new points are created after the cluster.



Point cluster on a polygon mesh.



Increasing the subdivisions in the primitive shifts the cluster.



Subdividing polygons does not shift the cluster. New points are added to the cluster based on their adjacency.

Topology Operators and Shape Animation

If you intend to modify or animate the parameters of a topology operator on an object with shape animation, then you should apply that topology operator after authoring all your shapes. The topology operator will be inserted in the modeling region below the Cluster Shape Combiner, but the shapes will update correctly. The shape animation does not require the topology operator to be re-evaluated, and you can freeze the topology operator without affecting the shapes.

For example, you can take a shape-animated object and apply polygon reduction to create one or more low-resolution versions of the object. In this way, all the shapes and transitions are preserved. However, if you apply polygon reduction first and then create the shape animation, the shapes are not correct when you change the amount of reduction.

Modifying Clusters Below Topology Operators

When you modify existing clusters or cluster-based properties, you may get no results or incorrect results if there are topology operators higher up in the stack. Such situations include:

- Adding or removing components from a cluster that is created below a topology operator in the stack. This includes clusters that are created automatically when you apply deformations or properties to selected components.
- Modifying weight map values (including envelope weights) when there is a topology operator above the weight map creation point or Envelope Operator in the stack.
- Modifying vertex colors when there is a topology operator above the Color at Vertices map creation point in the stack.

In these situations, you should simply disable all topology operators above the cluster creation point before modifying the cluster or property. When you have finished, re-enable the top of the stack and the object updates properly.

For more information about disabling operators, see *Disabling the Top of the Stack* on page 62.

Changing the Order of Operators

You can change the order of operators in an object’s stack by dragging and dropping them in an explorer view. You must always drop the operator onto the operator or marker that is immediately below the position where you want the dragged operator to go.

Be aware that you might not always get the results you expect, particularly if you move topology operators or move other operators across topology operators, because operators that previously affected certain components may now affect different ones. In addition, some deformation operators like MoveComponent or Offset may not give expected results when moved because they store offsets for point positions whose reference frames may be different at another location in the stack.

When you try to drag and drop an operator, XSI evaluates the implications of the change to make sure it creates no dependency cycles in the data. If it detects a dependency, it will not let you drop the operator in that location. Moving an operator up often works better than moving it down—this is because of hidden cluster creation operators on which some operators depend.

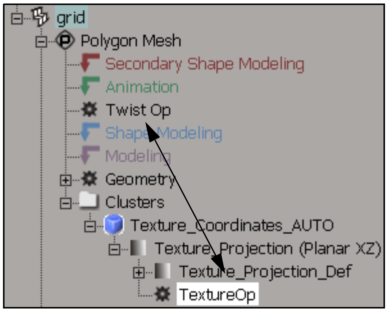
If you drag an operator to a new location and then drag it back to its original location, you might not get the original results. This is due to internal dependencies on clusters. However, if you undo after dragging an operator, you will always get the original results.

One exception to these rules is the Surface Continuity Manager operator (SCM Fixer Op2): you cannot move this operator and it is always at the top of the stack. Another exception is the Cluster Shape Combiner: it is always at the bottom of the Shape Modeling region and cannot be moved.

You can even drag and drop to reorder operators in different stacks on the same object, such as the modeling stack and a texture projection.

For example, if you applied a Twist deformation and then a texture projection, the texture projection “reads” from above the Twist. As a result, the texture slips when the Twist parameters are modified. In this case, you can drag the Twist Op node onto the TextureOp node to place the Twist above the texture projection so that the texture sticks.

In the other direction, you could drag the TextureOp node onto the top of the modeling stack so that you can freeze the modeling stack without freezing the projection.



Drag and drop to change the order of the operators.

Changing the Order of Operators Using Script Commands

When you move an operator by dragging and dropping in an explorer, the **CopyPaste** command is logged to the command history.

However if you want to move operators using scripts, you can also use the **SwapOps**, **MoveOperatorAfter**, and **MoveOperatorBefore** commands. If these commands fail, they provide more precise error messages than CopyPaste.

For more information about scripting in general, see *Chapter 2: Scripts* in the *Customization* guide.

Disabling the Top of the Stack

If you need to insert new operators in the middle of the stack, you can disable the stack from any operator up to the top. New operators are added to the active portion of the stack until you re-enable the top of the stack.

Disabling the stack is temporary. If you save and reload the scene, the stack is re-enabled.

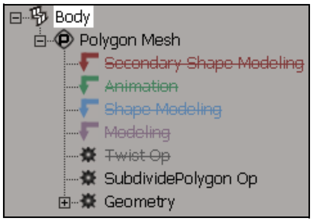
Follow these guidelines when disabling the top of the stack; otherwise, you may get unpredictable results.

- You should re-enable the entire stack before moving operators.
- Do not disable generators like Loft.
- While the top of the stack is disabled, do not perform any operations involving clusters that were created above the point of deactivation. Doing so may create data dependency cycles.
- Do not freeze a disabled stack, and then undo.

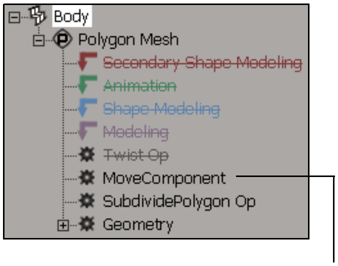
To disable the top of the stack

- In an explorer view, right-click the lowest operator you want to disable and choose **Disable from Here**.

  The operator and everything above it no longer affect the object and appear dimmed with strike-through text.



Body
Polygon Mesh
~~Secondary Shape Modeling~~
~~Animation~~
~~Shape Modeling~~
~~Modeling~~
~~Twist Op~~
SubdividePolygon Op
Geometry



Body
Polygon Mesh
~~Secondary Shape Modeling~~
~~Animation~~
~~Shape Modeling~~
~~Modeling~~
~~Twist Op~~
MoveComponent
SubdividePolygon Op
Geometry

New operators go under the disabled ones.

You can continue to disable other operators even lower in the stack to disable in stages.

To re-enable the top of the stack

- In an explorer view, right-click the bottom-most disabled operator and choose **Enable from Here**.

  The operator and those above are enabled up to the top of the stack or, if you disabled the stack in stages, up to the previously disabled operator.

Freezing the Operator Stack

When you are satisfied with an object, you can freeze all or part of its operator stack. This removes the current history—as a result, the object requires less memory and is quicker to update. You can no longer go back and change values, but you can still undo operations.

You can freeze the modeling region, freeze the entire object, or freeze the bottom of the stack from a particular operator down.

- Freezing removes any animation on the modeling operators (for example, the angle of a Twist deformation, and so on). The values at the current frame are used.
- For hair objects, the Hair Generator and Hair Dynamics operators are never removed.

To freeze the modeling region

1. Select an object.
2. Do one of the following:

   - Click **Freeze M** on the Edit panel.

   or

   - Choose **Edit** > **Operator** > **Freeze Modeling**.

   The modeling region is collapsed, including any weight maps and texture projections that were applied in the modeling region.

To freeze the entire object

1. Select an object.
2. Do one of the following:

   - Click **Freeze** on the Edit panel.

   or

   - Choose **Edit** > **Operator** > **Freeze Selected**.

   The entire construction history is collapsed, including the modeling stack as well as any texture operator stacks and weight painting stacks.

To freeze the bottom of the operator stack

- In an explorer, do one of the following:

  - Right-click an operator and choose **Freeze**.

  or

  - Select an operator in the stack, and then click **Freeze** on the Edit panel.

  or

- Select an operator in the stack, and then choose **Edit** > **Operator** > **Freeze Selected**.

The selected operator and everything below it in the stack is collapsed, including any weight maps and texture projections that were applied below the selected operator.

Collapsing Deformation Operators

Sometimes, it is useful to "freeze" certain operators in the stack without freezing earlier operators that are lower in the stack. For example, you might have many MoveComponent operators that are slowing down your scene, but you don't want to lose an animated deformation or a generator (if your object has a modeling relation that you want to keep).

Another situation where this would be useful is when you need to change the behavior of particular deformations. For example, if you apply a Bulge deformation, scale some points, and adjust the Bulge operator's amplitude, you'll see that the scaling is multiplied by the bulge effect. If you want an additive effect, you need to somehow "freeze" the MoveComponent operator created by scaling the points.

In these cases, you can collapse several deformation operator into a single Offset operator. The Offset operator is a single deformation that contains the net effect of the collapsed deformations at the current frame. The Offset operator has no parameters and no property editor.

When collapsing operators, here are some things you should know:

- Only deformations can be collapsed. You cannot collapse topology operators.
- You cannot collapse across topology operators. If a topology operator is in the middle of a set of deformation operators to be collapsed, two Offset operators are created: one above the topology operator and one below it.
- You cannot collapse across points in the stack where other operators read information, even though you cannot see these points in the stack.

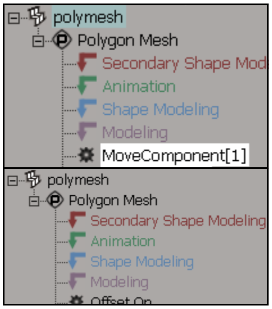
  For example, if you apply a Randomize deformation, create a texture projection, apply a Push deformation, then select the deformations and collapse them, two Offset operators are created because the texture projection is reading data from the point in-between. If you freeze the texture projection, you can then collapse the deformations into a single Offset operator.
- If the collapsed deformations contained any animated parameters, the animation is lost when you collapse them.
- If the collapsed deformations had any object inputs (such as envelope or spine deformers), that relation is lost after you collapse them.

- If you modify operators below an Offset operator, the overall effect might change.

  For example, let's say you have a Lattice deformation, followed by a Twist, and then you collapse the Twist into an Offset operator. If you then modify the lattice, you'll notice that the Offset operator still applies the same relative offset to each point, but depending on how you modify the lattice, it is no longer really a "twist".

- Offset operators are deformations like any other and can also be collapsed.

To collapse deformation operators

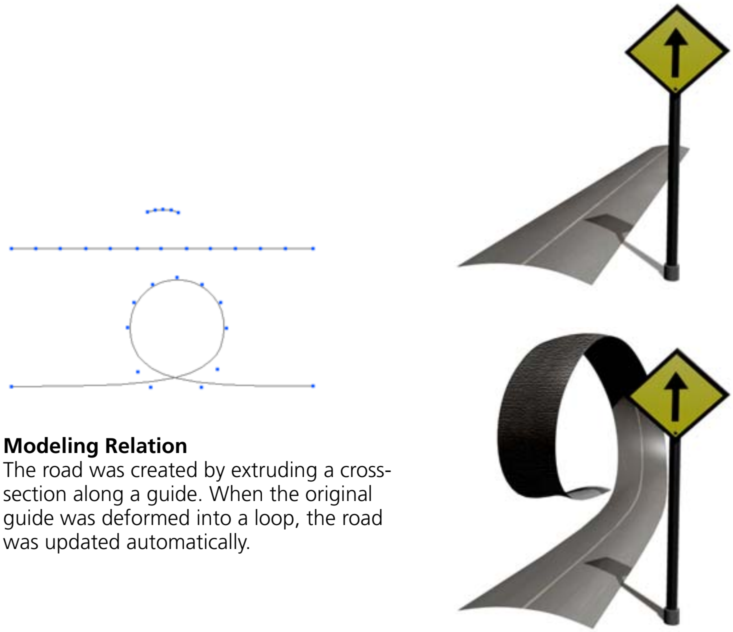


1. Select one or more deformation operators in the stack.
2. Choose **Edit** > **Operator** > **Collapse Operators**.

   Each sequence of consecutive selected deformations is collapsed into a single Offset Op.

Modeling Relations

When you generate an object from other objects, a modeling relation is established. For example, if you create a surface by extruding one curve along another curve, the resulting surface is linked to its generator curves. If you modify the curves, the surface updates automatically. The modeling relation is sometimes called *construction history*.

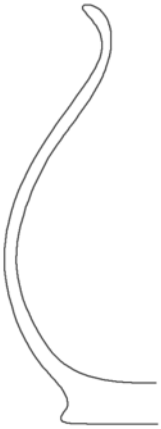


Modeling Relation
The road was created by extruding a cross-section along a guide. When the original guide was deformed into a loop, the road was updated automatically.

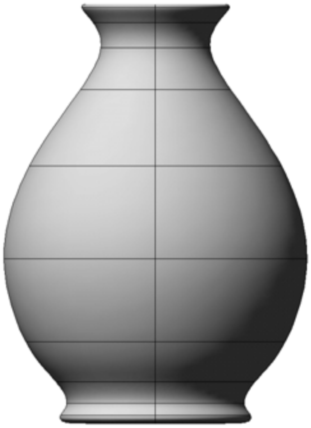
You can modify the generated object in any way you like, for example, by moving points or applying a deformation. When you modify the generators, the generated object is updated and your modifications to it are preserved—the object does not "snap" back to its generated shape the way it does in SOFTIMAGE|3D.

- When you are working in Immediate mode, the operator stack is not kept and there is no modeling relation. See *Immediate Mode* on page 71.
- If you delete the input objects, the generated object is removed as well. To avoid this, freeze the generated object (see *Freezing the Operator Stack* on page 63) or at least the generator (*Freezing Generators and Deleting Inputs* on page 68) before deleting the inputs.

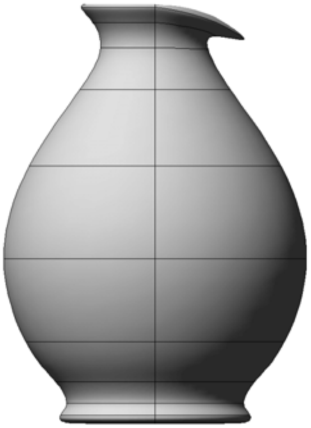
  If you accidentally delete an input, simply press Ctrl+z to undo and restore it and any generated objects.



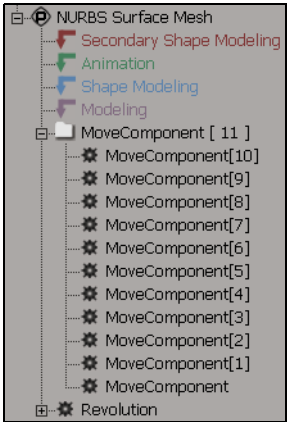
1 The original curve



2 Surface created by revolution.

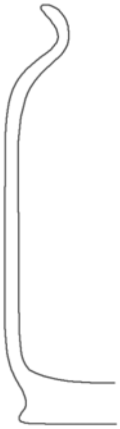


3 Revolved surface modified by moving points.

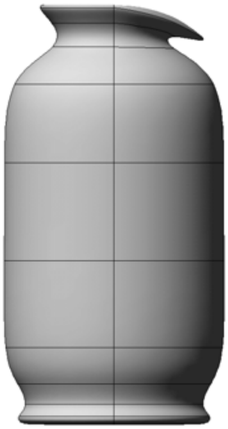


```
⊟ NURBS Surface Mesh
    Secondary Shape Modeling
    Animation
    Shape Modeling
    Modeling
  ⊟ MoveComponent [ 11 ]
      MoveComponent[10]
      MoveComponent[9]
      MoveComponent[8]
      MoveComponent[7]
      MoveComponent[6]
      MoveComponent[5]
      MoveComponent[4]
      MoveComponent[3]
      MoveComponent[2]
      MoveComponent[1]
      MoveComponent
  ⊞ Revolution
```

4 MoveComponent operators are added to the stack after the Revolution.



5 The original curve is modified.



6 The revolved surface automatically updates to reflect the new curve shape. The modifications you made to the surface after revolving are preserved.

Managing Generator Inputs

There are several commands that help you manage the inputs for generated objects. You can:

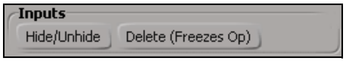
- Automatically hide or unhide the inputs for a generated object. This simplifies the display of your scene in the 3D views, but still allows you to modify the inputs later if necessary.
- Freeze the generator and delete the inputs. This simplifies the scene once you are satisfied that you won't need to modify the inputs again.
- Select all the inputs for a generated object.

Hiding and Unhiding Inputs

You can automatically hide all the inputs for a generated object to simplify the display of your scene in the 3D views, and then unhide the inputs when you need to modify them.

To hide or unhide input objects for a generated object

Do one of the following:



- In the generator operator's property editor, click **Hide/Unhide** in the Inputs box.

  or

- In an explorer, right-click on the generated object or the generator operator node and choose **Hide/Unhide Generator Inputs**.

  or

- In a 3D view or a schematic view, Alt+right-click (Ctrl+Alt+right-click on Linux) on the generated object and choose **Hide/Unhide Generator Inputs**.

  or

- Select the generated object, and then choose **Edit** > **Operator** > **Hide/ Unhide Generator Inputs**.

Freezing Generators and Deleting Inputs

You can freeze the generator and delete the inputs. This simplifies the scene once you are satisfied that you won't need to modify the inputs again.

To freeze generators and delete their inputs

Do one of the following:

- In the generator operator's property editor, click **Delete (Freezes Op)** in the Inputs box. The property editor closes automatically.

  or

- In an explorer, right-click on the generated object or the generator operator node and choose **Freeze Generator & Delete Inputs**.

  or

- In a 3D view or a schematic view, Alt+right-click (Ctrl+Alt+right-click on Linux) on the generated object and choose **Freeze Generator & Delete Inputs**.

  or

- Select the generated object, and then choose **Edit** > **Operator** > **Freeze Generator & Delete Inputs**.

Selecting Inputs

At any time, you can quickly select all the inputs for a generated object.

To all the inputs for a generator

Do one of the following:

- In an explorer, right-click on the generated object or the generator operator node and choose **Select Generator Inputs**.

  or

- In a 3D view or a schematic view, Alt+right-click (Ctrl+Alt+right-click on Linux) on the generated object and choose **Select Generator Inputs**.

  or

- Select the generated object, and then choose **Edit** > **Operator** > **Select Generator Inputs**.

Generator Inputs Preference

The **Generator Inputs** option in the Modeling preferences determine whether the generator input objects are kept, hidden, or deleted automatically. This preference applies only when an object is generated without opening its property editor—that is, if **Popup Property Editors on Node Creation** is off in your Interaction preferences, or if it is on and you press Ctrl while choosing a menu item.

You can set **Generator Inputs** to the following values:

- **Keep** does not change the visibility of the inputs, and maintains the modeling relation with the generated object.
- **Hide** turns off the view and render visibility of the inputs, but keeps the modeling relation. This simplifies the display of your scene, but still allows you to unhide and modify the inputs if you need to later.
- **Delete** freezes the generator operator and deletes the inputs, thereby breaking the modeling relation.

For more information about preferences, see *Chapter 26: User Preferences* in the *Fundamentals* guide.

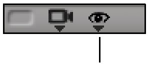
Substituting Inputs

Many modeling operations allow you to substitute one input object for a different one. For example, you can change the profile or guide curve of an extrusion. This works best when the object is an input to only one operator.

To substitute inputs

1. Select the input object you want to replace.
2. Choose **Edit** > **Operator** > **Substitute Input**.
3. Pick a new object to be the input.

Showing Modeling Relations in the 3D Views



To see modeling relations in a 3D view, click the eye icon (Show menu) and make sure that **Relations** is on.

If the selected object has a modeling relation, it is linked to its input objects by dotted white lines. A white label on the dotted line identifies the type of relation (such as wave or revolution) and the name of the input object.

You can click the dotted line or the label to select the corresponding operator.

Optimizing Performance with Modeling Relations

When a modeling relation exists, a large amount of data may need to be updated if the input objects are transformed or deformed. If you need to maintain the modeling relation, you can improve performance by turning off geometry updates for playback.

For more information about the playback options, see *Optimizing Playback* in Chapter 3 of the *Animation* guide.

Breaking the Modeling Relation

To break the modeling relation, freeze the generated object’s operator stack as described on page 63.

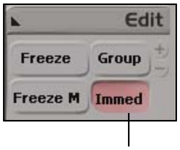
- If you delete the generator (input) objects without freezing the operator stack of the generated (output) object first, the output object is removed from the scene. For more information, see *Freezing the Operator Stack* on page 63.
- If this happens accidentally, press Ctrl+z to undo.

Immediate Mode

Immediate mode provides a way of modeling without building an operator stack. Using this mode is equivalent to freezing the selected objects after every operation.

Immediate mode is especially useful when moving points or adding and extruding polygon components. The stack does not fill up with a large number of operators, and therefore, scenes are smaller and quicker to update.

To activate immediate mode



Immediate mode

- Click the **Immed** button on the Edit panel in the main command area.

Whenever you apply an operator in immediate mode, its property editor opens with OK and Cancel buttons. Make all the changes you want before clicking OK to confirm that you are satisfied with the results. If you are editing multiple objects (for example, if you select several objects and choose **Deform** > **Twist**), the same values are applied to all objects.

Be very careful when using immediate mode: when it freezes the stack after each operation, it removes any previously applied operators.

In particular, you will lose any animation applied to operators' parameters, such as the amplitude of a deformation. For more information, see *Freezing the Operator Stack* on page 63.

Immediate mode may be slow on very large models because it freezes after every action. In such cases, it may be better to freeze manually every so often.

Commands and Immediate Mode

If you want to be able to replay commands from the history, don't use immediate mode. When you get a primitive or apply an operator in immediate mode, the corresponding property editor opens with OK and Cancel buttons. If you change a parameter value while the property editor is open, no **SetValue** command is logged.

As a result, you will get different results when you play back commands from the history.

Chapter 4 Generating Objects from Curves

Generating Objects from Curves

You can create surfaces and polygon meshes from curves in several ways:

- Extruding an open curve along two guides (Birail).
- Picking a series of curves in U and V (Curve Net).
- Extruding a curve along an axis.
- Extruding a curve along another curve.
- Extruding with two profiles, morphing one into the other as it runs along a rail curve.
- Picking the four curves to use as boundaries (Four Sided).
- Creating a series of profiles and lofting them.
- Revolving a curve around an axis or another curve.

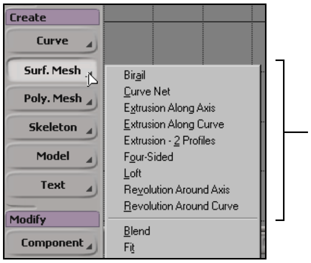
For information about creating curves to work from, see *Chapter 11: Curves* on page 359.

For information about how you can modify the generated polygon meshes and surfaces further, see *Chapter 6: Deformations* on page 131, as well as *Section II • Polygonal Modeling* on page 179 and *Section III • NURBS Modeling* on page 345.

About Building from Curves

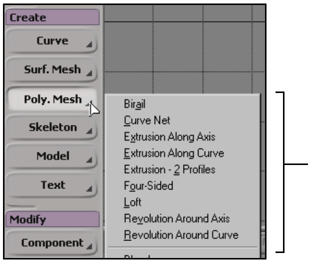
There are two sets of commands for generating surfaces and polygon meshes from curves:

- The first group of commands in the **Create** > **Surf. Mesh** menu on the Model toolbar create NURBS surface meshes consisting of one subsurface.



Create surface from curves

- The first group of commands in the **Create** > **Poly. Mesh** menu on the Model toolbar create polygon meshes.



Create polygon mesh from curves

The commands and the procedures on these two menus are the same. The only difference is the type of object that is created.

The sections that follow include some basic information that is common to all or most of the operators that generate objects from curves.

Picking Input Curves

In the procedures that follow, you are not limited to curve objects for building surfaces. You can select any combination of isolines, knot curves, boundaries, surface curves, and trim curves as well as curve objects. For example, you can create a loft surface that joins two surface boundaries while passing through other curves.

Most of the generators, like extrusion, revolution, and so on, do not work with curve objects composed of multiple subcurves such as those obtained from text or imported EPS files—what happens is that only the first subcurve is used. However, you can extract the subcurves as separate curve objects first, as described in *Extracting Curve Segments* on page 369.

For a quick overview of using filters to select different types of component curves on surfaces, see *Selecting Surface Components* on page 390. For more more information about selecting components, see *Chapter 8: Selecting* of the *Fundamentals* guide.

Controlling Subdivisions

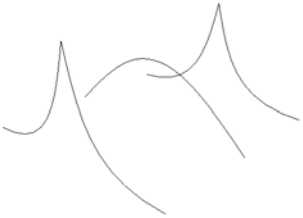
Most of the operators that generate objects from curves, like extrusion, revolution, and so on, use a standard set of controls to determine the subdivisions (knot curves) of the result.

- If **Subdivision Type** is **Per Span**, the basic subdivisions are based on the "natural" subdivisions as determined by the shapes of the inputs. You can subdivide each segment further using the **U** and **V span** sliders.

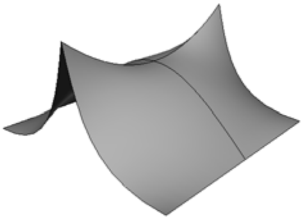
  In this case, discontinuities resulting from multiknots in the inputs are preserved in the generated objects.

- If **Subdivision Type** is **Absolute**, you can use the **U** and **V abs** sliders to set the exact number of subdivisions in the resulting object.

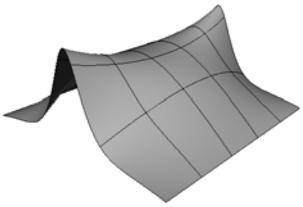
  However, the result is automatically reparameterized and discontinuities are not preserved.



Input curves for a Loft operation



Subdivision Type = Per Span
Result uses the minimum number of subdivisions to follow the inputs—in this case, 2x2.



Subdivision Type = Absolute
You can set the Subdivision sliders to the desired amounts—like 5x5. However, discontinuities are smoothed.

Controlling Degree of Surfaces

When you build a NURBS surface, the result can be linear or cubic in either U or V, depending on whether it is built from linear or cubic inputs in each direction. For example, if you use Curve Net (page 79) and all your U input curves are cubic and all your V input curves are linear, the result is cubic in U and linear in V.

If you use a mixture of cubic and linear inputs in the same direction, then the result will always be cubic in that direction.

For those operations where you specify inputs for only one direction, like Loft, Extrusion Along Axis or Revolution, the V direction of the result is always cubic.

In all cases, you can use Fit Surface to create a new surface with the degrees you desire—see *Fitting Surfaces* on page 395.

Controlling the Direction of Normals

The direction in which curves are drawn determines the direction in which the normals point on the resulting object. Remember to draw curves counterclockwise for the proper alignment of normals.

If the normals of the resulting object are pointing in the wrong direction, you can invert a surface as described on page 399, invert a polygon mesh as described on page 275, or invert one or more of the input curves as described on page 380.

Managing Input Objects

Whenever you generate a new object from other input objects, you can manage the inputs, for example, you can hide or delete them. See *Managing Generator Inputs* on page 68.

Using Birail

The Birail operator lets you extrude an open-profile curve by running its endpoints along two rails, or guide curves. This is sometimes called *guided extrusion*. The distance between points determines which end of which rail is connected to each end of the profile.

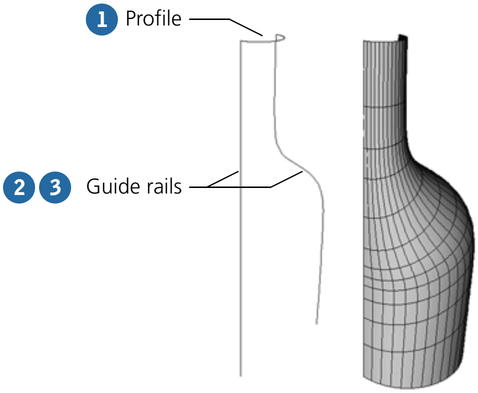
Birail works only with an open profile curve. For information on how to open a closed curve, see *Opening and Closing Curves* on page 381.

To create a surface or polygon mesh using Birail

1. Select the profile curve, then add the two rail curves to the selection. You can select curve objects as well as U or V isolines, U or V knot curves, boundaries, surface curves, and trim curves.
2. Choose one of the following commands from the Model toolbar:
   - **Create** > **Surf. Mesh** > **Birail**
   - **Create** > **Poly. Mesh** > **Birail**

   If you didn't select all the curves in step 1, you can pick the remaining ones now in order. This creates an object and the Birail property editor opens.
3. Adjust the parameters as desired. For more information, refer to Online Help by clicking on the ? in the property editor.

Birail



Using Curve Net

You can create a surface or polygon mesh by picking a series of curves in U and V. Curve Net is similar to Loft, but you can control the detail in both the U and V directions. For best results, make sure the curves intersect each other as closely as possible, particularly at the boundaries.

To create a surface or polygon mesh using Curve Net

1. Optionally, select one or more U curves in order. You can select curve objects as well as U or V isolines, U or V knot curves, boundaries, surface curves, and trim curves.
2. Choose one of the following commands from the Model toolbar:
   - **Create** > **Surf. Mesh** > **Curve Net**
   - **Create** > **Poly. Mesh** > **Curve Net**
3. Pick the remaining U curves in order.

   - If you make a mistake, Ctrl+click to “unpick” the last curve. Repeat to “unpick” multiple curves.
   - With the Free Form selection tool, you can pick the curves in a single sweep. Be careful not to accidentally select a V curve.

4. Right-click to indicate that you have finished picking U curves.
5. Pick the V curves in order.
6. Right-click. The Curve Net property editor opens.
7. Adjust parameters as necessary:
   - Specify the number of **Subdivisions** in **U** and **V**.
   - Set **Feature Match** to follow the detail of the input curves more closely. Note that this may significantly affect performance.

Curve Net treats closed curves as if they were open curves with overlapping boundaries. When using Curve Net with closed curves, it is important to align their boundaries and pick the transverse curves in order, starting at the boundary.

Extruding Curves along Axes

You can create a surface or polygon mesh by extruding a curve along an axis.

To extrude a curve along an axis

1. Select the curve to be extruded. You can select a curve object, U or V isoline, U or V knot curve, boundary, surface curve, or trim curve.
2. Choose one of the following commands from the Model toolbar:

   - **Create** > **Surf. Mesh** > **Extrusion Along Axis**
   - **Create** > **Poly. Mesh** > **Extrusion Along Axis**

   If you did not select a curve in step 1, you can pick one now. This creates an object and the Extrude Curve property editor opens.
3. Adjust the parameters as desired. For more information, refer to Online Help by clicking on the ? in the property editor.

Extruding Curves along Curves

You can create a surface or polygon mesh by extruding a profile curve along a guide or rail curve. This and similar operations are sometimes called *rail* or *sweep*. If the input curves are closed, the resulting object will be closed in the corresponding direction; if they are open, you have the option to close the result.

To extrude a curve along a curve

1. Select the profile curve first, then add the guide curve to the selection. You can select curve objects as well as U or V isolines, U or V knot curves, boundaries, surface curves, and trim curves.

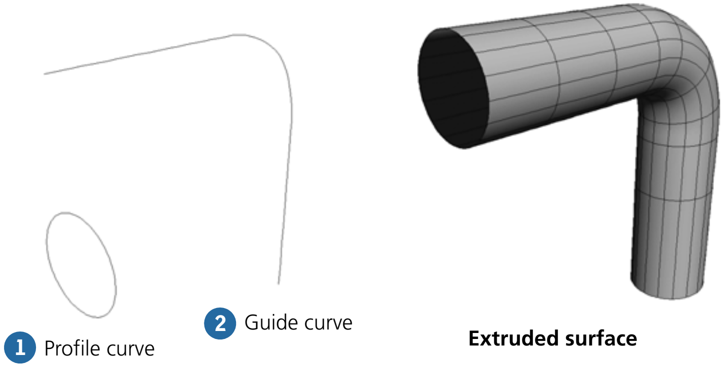
2. Choose one of the following commands from the Model toolbar:

   - **Create** > **Surf. Mesh** > **Extrusion Along Curve**

   - **Create** > **Poly. Mesh** > **Extrusion Along Curve**

   If you didn't select a profile and guide curve in step 1, you can pick them now. An object is extruded and the Extrude Curve property editor opens.

3. Adjust the parameters as desired. For more information, refer to Online Help by clicking on the ? in the property editor.



Extruding with Two Profiles

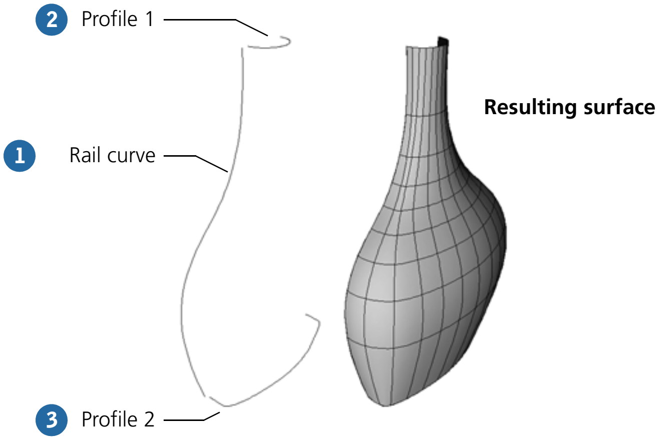
When you create a surface or polygon mesh by extruding with two profiles, the first profile morphs into the second profile as it runs along a single rail curve. The distance between points determines which end of which profile is connected to each end of the rail.

To extrude with two profiles

1. Select the rail curve, then add the two profiles to the selection. You can select curve objects as well as U or V isolines, U or V knot curves, boundaries, surface curves, and trim curves.
2. Choose one of the following commands from the Model toolbar:

   - **Create** > **Surf. Mesh** > **Extrusion - 2 Profiles**
   - **Create** > **Poly. Mesh** > **Extrusion - 2 Profiles**

   If you didn't select the curves in step 1, you can pick them now. This creates an object and the Extrusion 2 Profiles property editor opens.
3. Adjust the parameters as desired. For more information, refer to Online Help by clicking on the ? in the property editor.



Picking Boundaries Using Four-Sided

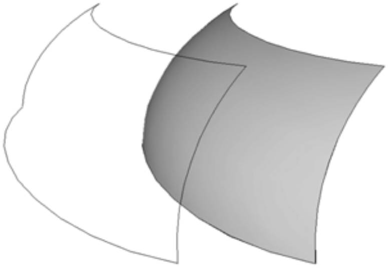
You can create a simple surface or polygon mesh by picking four curves to define the surface's boundaries. For best results, make sure that their ends meet as closely as possible.

To pick boundaries using Four-sided

1. Select the first curve, then add the remaining curves to the selection in clockwise order. You can select curve objects as well as U or V isolines, U or V knot curves, boundaries, surface curves, and trim curves.
2. Choose one of the following commands from the Model toolbar:

   - **Create** > **Surf. Mesh** > **Four Sided**
   - **Create** > **Poly. Mesh** > **Four Sided**

   If you didn't select the curves in step 1, you can pick them now. This creates an object and the FourSided property editor opens.
3. Set the desired number of subdivisions in U and V.



Four sided

Lofting Curves

You can create a series of profile curves, then use the Loft command to create a surface or polygon mesh with the corresponding cross-sections. This procedure is sometimes called *skinning*.

- If the input curves are closed, the resulting surface will be closed.
- If they are open, you have the option to close the result.
- After you have created a surface by lofting, you can add and remove input curves.

To loft curves

1. Select one or more curves in the series in order. You can select curve objects as well as U or V isolines, U or V knot curves, boundaries, surface curves, and trim curves.
2. Choose one of the following commands from the Model toolbar:
   - **Create** > **Surf. Mesh** > **Loft**
   - **Create** > **Poly. Mesh** > **Loft**
3. Pick the remaining profile curves in order.

   If you make a mistake, you can unpick curves in order using Ctrl+click. You can also cancel the loft operation by pressing Esc.

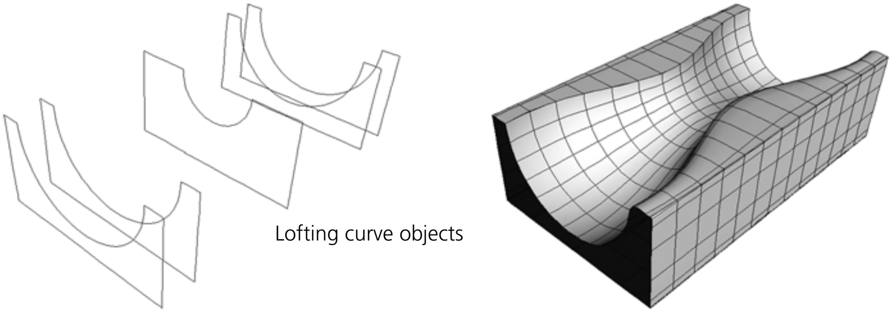
With the Free Form selection tool, you can pick curve objects in a single sweep.

Loft does not work if there are overlapping curves. If you created curves by duplicating them, make sure that two or more don't overlap.

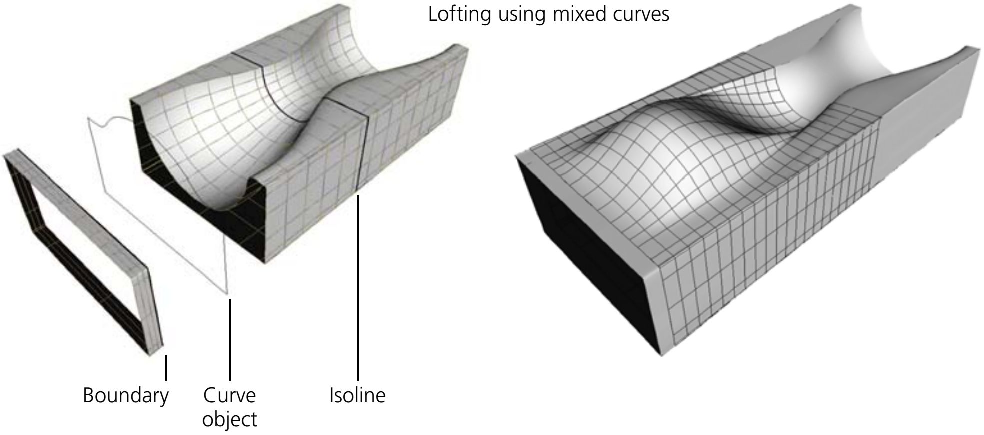
4. Right-click to indicate that you have finished picking profile curves. The Loft property editor opens.
5. Set the parameters as desired:
   - Specify the number of **Subdivisions** in **U** and **V**.
   - Use the **Start** and **End Surface** parameters only when the first and last curves are on surfaces (boundaries, knot curves, or isolines). They determine how the continuity of the loft surface matches that of the input surfaces.

     For more information, refer to Online Help by clicking on the ? in the property editor.

You can quickly create a low-resolution version of surfaces by lofting selected isolines.



Lofting curve objects



Lofting using mixed curves

To add a curve

1. Select the input curve after which you want to add a new curve.

   To add a curve before the first one, use **Surface** > **Modify** > **Extend to Curve** instead of this procedure. For more info, see *Extending Surfaces* on page 402.

2. Choose **Edit** > **Operator** > **Add Input**. A picking session starts.

3. Pick a curve. The curve is added to the Loft operator and the generated object is rebuilt according to the current settings.

   Note that if the curve is an input for more than one loft operation, the new curve will be added to the last one.

To remove a curve

1. Select a curve to remove. If you select multiple curves, only the first one will be removed.
2. Choose **Edit** > **Operator** > **Remove Input**. The curve is removed from the Loft operator and the generated object is rebuilt according to the current settings.

   If the curve is an input for multiple loft operations, it is removed from the last one.

Revolving Curves around Axes

You can create a surface or polygon mesh by revolving a curve around a global axis.

- If the profile curve is closed, the resulting surface will be closed in V.
- If it is open, you have the option to close the result. The surface is always closed in U if it sweeps through a full 360 degrees.

To revolve a curve around an axis

1. Select the curve to be revolved. You can select a curve object, U or V isoline, U or V knot curve, boundary, surface curve, and trim curve.
2. Choose one of the following commands from the Model toolbar:

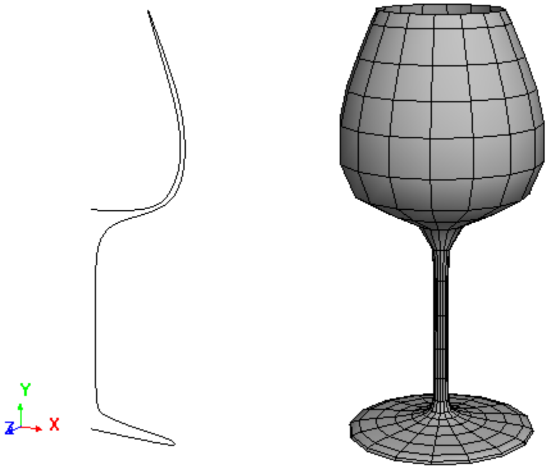
   - **Create** > **Surf. Mesh** > **Revolution Around Axis**

   - **Create** > **Poly. Mesh** > **Revolution Around Axis**

   If you did not select a curve in step 1, you can pick one now. This creates an object and the Revolution property editor opens.
3. Adjust the parameters as desired.

   In particular, you can specify the axis to extrude along or select a combination of axes to revolve around the resulting diagonal.

   For more information, refer to Online Help by clicking on the ? in the property editor.



Revolving Curves Around Curves

You can create a surface or polygon mesh by revolving a curve around an axis defined by another curve.

- If the profile curve is closed, the resulting surface will be closed in V.
- If it is open, you have the option to close the result. The surface is always closed in U if it sweeps through a full 360 degrees.

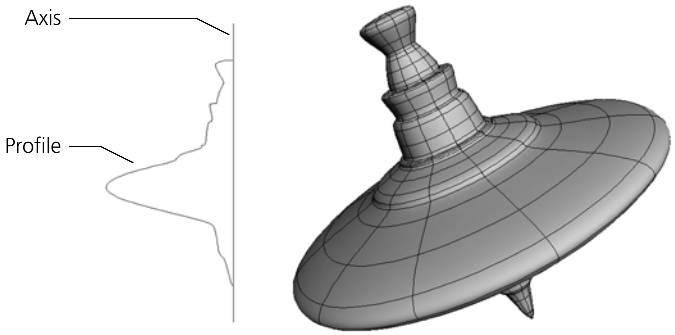
To revolve a curve along a curve

1. Select the curve to be revolved, then add a curve to be revolved around to the selection. You can select curve objects as well as U or V isolines, U or V knot curves, boundaries, surface curves, and trim curves.
2. Choose one of the following commands from the Model toolbar:

   - **Create** > **Surf. Mesh** > **Revolution Around Curve**
   - **Create** > **Poly. Mesh** > **Revolution Around Curve**

   If you didn't select the curves in step 1, you can pick them now. This creates an object and the Revolution property editor opens.
3. Adjust the parameters as desired. For more information, refer to Online Help by clicking on the ? in the property editor.

Revolution



Chapter 5 Manipulating Components

Manipulating Components

You can model an object by transforming its points, edges, and polygons. This chapter covers the basic tools and techniques for doing the following:

- Translating, rotating, and scaling components.
- Using the Move Point tool.
- Manipulating components relative to their neighbors.
- Manipulating points with a proportional falloff.
- Manipulating points symmetrically.
- Using volume deformers to manipulate all points in a volume of 3D space with a proportional falloff.
- Using Push Paint to sculpt an object by painting.
- Manipulate clusters by applying a center and transforming it.

About Component Manipulation

In XSI, you can select and transform points, edges, and polygons directly on a geometric object. This lets you shape the geometry into the form that you want and is an integral part of modeling.

It is important to know that although these techniques use the transformation tools, they are actually deformations. Each time you manipulate components, a deformation operator is added to the object's stack. Some of these operators, like MoveComponent, don't have any parameters and therefore do not display a property editor; others, like Proportional, do have parameters that you can modify.

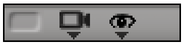
For information about deformations in general, see *Chapter 6: Deformations* on page 131.

Viewing Components and Attributes

You can display or hide components like points, edges, and polygons in the 3D views using exactly the same way as other components and attributes. A quick summary is provided here—for complete details about viewing, see *Chapter 5: Viewing 3D Objects* in the *Fundamentals* guide.

To display or hide components in a single 3D view

Do one of the following:



- Click the eye icon (Show menu) and choose the corresponding component option. You can press Shift to keep this menu open while you toggle multiple items.

  or

- Press Shift+s, or click the eye icon and choose **Visibility Options**. The Camera Visibility property editor contains many more options than the menu, for both selected and unselected objects.

To display or hide components in all open 3D views

Do one of the following:

- Choose **View** on the main menu and then choose the corresponding component option.

  or

- Choose **View** > **Visibility Options** (**All Cameras**). The Camera Visibility property editor contains many more options than the menu, for both selected and unselected objects.

Selecting Components and Clusters

You can select and work with points, edges, and polygons in the same way as any other type of component using the filters on the Select panel. Selection is fully described in *Chapter 8: Selecting* of the *Fundamentals* guide and is quickly summarized here.

The specific mouse buttons and modifier keys that you use to perform different actions—like selecting, adding to the selection, toggling, and deselecting—depend on the selection interaction model. For complete details, see *Selecting Components Using the Different Interaction Models* in Chapter 8 of the *Fundamentals* guide.

Summary of selection filters

Component selection filters determine what type of component you can select on the active geometric objects. The most commonly used filters are available as buttons on the Select panel, and additional filters are available from the Filter menu below the buttons.

The component selection filters are context-sensitive; they change depending on what type of object is active. For example, the **Point** button is available for polygon meshes, curves, surfaces, and lattices, while the **Edge** and **Polygon** buttons are available only for polygon meshes.

Summary of selection tools

The selection tools determine the mouse interaction used select objects. You can activate a selection tool with the items on the **Select** > **Tools** menu. Alternatively, you can press one of the following keys:

- **F7** for the **Rectangle** selection tool.
- **F8** for the **Lasso** selection tool.
- **F9** for the **Freeform** selection tool.
- **F10** for the **Raycast** selection tool.
- **Shift+F10** for the **Rectangle-Raycast** tool.
- **F11** for the **Paint** selection tool (points, polygons, and edges only).

Summary of selection modes

Selection modes are combinations of a filter and tool mapped to a single key for convenience. Use any of the following keys:

- **t** for **Rectangle Point** mode.
- **e** for **Rectangle Edge** mode.
- **i** for **Raycast Edge** mode.
- **y** for **Rectangle Polygon** mode.
- **u** for **Raycast Polygon** mode.
- **'** (apostrophe) for **Rectangle Tip** mode (hair only).

Basics of Transforming Components and Clusters

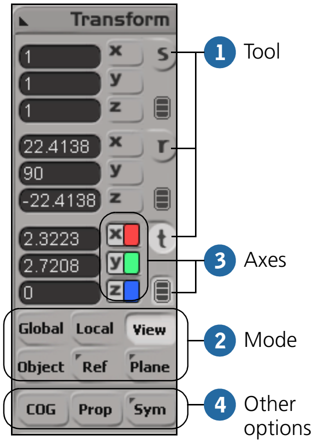
You can scale, rotate, and translate selected components and clusters using the tools on the Transform panel.

For information about transforming objects instead of components, see *Chapter 16: Transformations* in the *Fundamentals* guide.

You may not get the desired results if you try to control the center of transformation by moving the object center multiple times while transforming components.

Instead of moving the object center, you can control the center of transformation in different ways by activating COG, by transforming in Ref mode, or by setting the pivot.

Transforming Components Interactively



The Scale, Rotate, and Translate tools let you transform components and clusters interactively by dragging the mouse in the 3D views.

To transform components and clusters interactively

1. Select the components and clusters to transform, and activate a transformation tool:

   - To translate, press **v** or click the **t** button on the Transform panel.

     or

   - To rotate, press **c** or click the **r** button on the Transform panel.

     or

   - To scale, press **x** or click the **s** button on the Transform panel.

2. If desired, change the manipulation mode. The manipulation mode controls the reference frame used when you scale, rotate, and translate elements. Each tool remembers the last mode used.

   For a description of each mode, see *Component Translation Modes* on page 101, *Component Rotation Modes* on page 104 or *Component Scaling Modes* on page 106. The numeric boxes on the Transform panel always show the components' transformation values in terms of the current reference frame.

3. If desired, you can restrict the interaction to specific axes. See *Specifying Axes* on page 94.

4. If desired, activate any other options:

   - **COG** (center of geometry) changes the reference frame used to manipulate the components and clusters. See *Component Translation and COG Mode* on page 102, *Component Rotation and COG Mode* on page 105, and *Component Scaling and COG Mode* on page 107.

- **Prop** (proportional modeling) affects nearby points with a falloff based on distance. See *Using Proportional Modeling* on page 112.
- **Sym** (symmetry) affects symmetrical points if a symmetry map has been applied. See *Manipulating Components Symmetrically* on page 117.

5. If desired, you can set the manipulation pivot. The pivot defines the position around which elements are rotated or scaled. When translating with snapping, the pivot is the position that snaps to the target. See *Setting the Manipulation Pivot* on page 95.

6. Manipulate the components using the mouse in the 3D views:

   - If you are using the SRT manipulators, click and drag on different parts (*handles*) of the displayed transformation axes to transform the objects in different ways.

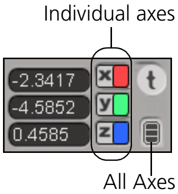
     For general information about the SRT manipulators, see *Transforming Interactively with the SRT Manipulators* on page 96. For specific information about each manipulator, such as the handles used by different tools, see *Using the Translate Manipulator* on page 101, *Using the Rotate Manipulator* on page 104, or *Using the Scale Manipulator* on page 107.

   - If you are not using the SRT manipulators, click and drag the mouse anywhere in the 3D views. The function of the left, middle, and right mouse buttons depends on the tool, mode, and other options—check the mouse/status line at the bottom of the main window for details. You can also drag while pressing multiple mouse buttons simultaneously.

- You can change manipulation modes and other options even while you work with a transformation tool in supra mode. Keep the **x**, **c**, or **v** key pressed while clicking buttons on the Transform panel.
- As an alternative to the Translate tool, you can translate individual points with the Move Point tool. See *Moving Points* on page 108.
- You can use snapping while translating or moving components. See *Component Translation and Snapping* on page 103.

Specifying Axes

When transforming interactively, you can specify which axes are active using the **x**, **y**, and **z** icons in the Transform panel. For example, you activate rotation in Y only or deactivate translation in Z. Active icons are colored and inactive icons are gray.



-2.3417
-4.5852
0.4585

- Click an axis icon to activate it and deactivate the others.
- Shift+click an axis icon to activate it without affecting the others.
- Ctrl+click an axis icon to toggle it.
- Click the All Axes icon to activate all three axes.

- Ctrl+click the All Axes icon to toggle all three axes.

Setting the Manipulation Pivot

When transforming elements interactively, you can set the manipulation pivot using the Alt key. The pivot defines the position around which elements are rotated or scaled. When translating with snapping, the pivot is the position that snaps to the target.

The same pivot is used for all selected elements. The pivot is normally reset when you change the selection. However, you can lock and reset the pivot position manually.

To set the manipulation pivot

- With a transformation tool active, do one of the following:
  - Alt+click in a 3D view. The pivot snaps to the closest point, edge midpoint, polygon midpoint or object center among the objects with selected components.

    If Ref or Plane manipulation mode is active, the current reference plane is automatically updated. For information about reference planes, see *Chapter 17: Reference Planes* in the *Fundamentals* guide.

    or

  - Alt+middle-click to place the pivot freely in the view plane of a 3D view. You can drag the mouse before releasing the middle mouse button to adjust the pivot's location.

    You can also activate snapping to snap the pivot to any available target type. For more information, see *Chapter 18: Snapping* in the *Fundamentals* guide.

To lock the manipulation pivot

Pivot icon ◉ ▽

1. With a transformation tool active, press and hold Alt. The pivot's icon appears.
2. Click on the triangle below the pivot's icon. A menu opens.
3. Choose **Lock**.

   The pivot remains at its current location, even if you change the selection or activate another tool, until you reset it.

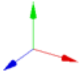
To reset the manipulation pivot

1. With a transformation tool active, press and hold Alt. The pivot's icon appears.
2. Click on the triangle below the pivot's icon. A menu opens.
3. Choose **Reset**.

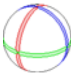
   The pivot returns to the default location for the current manipulation mode.

Transforming Interactively with the SRT Manipulators

Transformation Axes



Translate



Rotate

Scale

The SRT manipulators let you transform elements interactively by clicking and dragging on different parts (*handles*) of the displayed transformation axes. You can also use gestural manipulation or perform a default transformation.

To enable or disable the SRT manipulators

- Do one of the following:
  - Choose **Transform** > **Enable Transform Manipulators**.

    or

  - Choose **Transform** > **Transform Preferences**, and then toggle **Use 3D Manipulators**.

To use the manipulator handles directly

1. Move the mouse pointer over the manipulator handle you want. Different tools have different handles:
   - For a description of the Translate manipulator handles, see *Using the Translate Manipulator* on page 101.
   - For a description of the Rotate manipulator handles, see *Using the Rotate Manipulator* on page 104.
   - For a description of the Scale manipulator handles, see *Using the Scale Manipulator* on page 107.

   The pointer updates to reflect the axes or plane of manipulation.
2. Click and drag.

To use gestural manipulation

- Middle-click and drag anywhere in a 3D view.

  The axis that most closely matches the mouse movement over the first few pixels is used for the transformation.

  Gestural manipulation works only if **Click Outside Manipulator** is set to **No Interaction** or **Simple Transform** in the Transform preferences. See the next section, *To choose the behavior when clicking outside of the manipulators*, for more information about these options.

To choose the behavior when clicking outside of the manipulators

You can choose how the transform tools behave when you click outside of the manipulators.

1. Choose **Transform** > **Transform Preferences**.
2. Set **Click Outside Manipulator** to one of the following options:

- **No Interaction:** There is no effect when you click outside the manipulator.
- **Select Tool:** Clicking outside the manipulator performs a selection using the current selection filter and the last-used selection tool.
- **Simple Transform:** Clicking outside the manipulator performs a transformation using a default manipulation mode: View for the Translate and Rotate tools, or Uni for the Scale tool.
- **SI3D-style Transform:** Clicking outside the manipulator performs a transformation as if **Enable Transform Manipulators** is off. The left, middle, and right mouse buttons transform the selected elements differently, depending on the current manipulation mode set on the Transform panel.

To change options

When using SRT manipulators, you can open a context menu by right-clicking over the manipulator in a 3D view. The commands on this menu let you change the manipulation mode and set other options without moving the mouse pointer over to the Transform panel.

Transforming Interactively without the SRT Manipulators

When you are transforming elements interactively without using the SRT manipulators (that is, when **Transform** > **Enable Transform Manipulators** is off), the different mouse buttons manipulate the object in different ways.

The exact way depends on the manipulation mode as well as other options, but you can always look at the mouse/status line at the bottom of the XSI window to see what each button will do. Usually the different mouse buttons transform the object along different axes or planes, but if a single axis is active they transform at a slow, medium, or fast rate.

Transforming Components and Clusters in Local Mode

Local transformations are performed along the components' or clusters' own reference axes. Unlike objects, this reference frame is computed dynamically: it is the average local reference frame of the selected components.

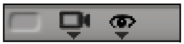
Note that as you transform components, their local axes may get recomputed. Therefore, you might get slightly different results with one larger transformation than with two or more smaller ones.

There are a couple of options that affect how this local reference frame is calculated and used:

- **Transform Local Components Independently** on the **Transform** menu controls whether adjacent components are treated separately or together. See *Transforming Components Independently* on page 98.

- **COG** controls whether non-adjacent components are treated separately or together when translating—see *Component Translation and COG Mode* on page 102. It also controls the pivot point used for rotating and scaling components and clusters—see *Component Rotation and COG Mode* on page 105 and *Component Scaling and COG Mode* on page 107.

To show the local reference frame of selected components



1. Do one of the following:

   - For a single 3D view, choose **Visibility Options** from the eye icon (Show menu) or press Shift+s.

   or

   - For all open 3D view, choose **View** > **Visibility Options** (**All Cameras**) from the top menu.

   The Camera Visibility property editor opens.

2. Activate **Cluster Reference Frame** on the Attributes tab.

To show multiple cues when transforming components separately

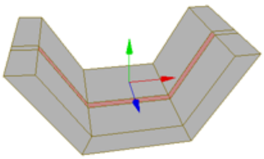
1. Open the Preferences window by choosing **File** > **Preferences**.
2. Expand **Tools** and click **Transform**.
3. Activate the **Show Multiple Transformation Axes** option.

Transforming Components Independently

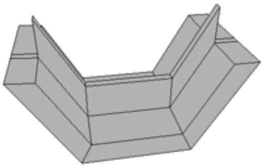
Transform Components Independently in Local Mode on the **Transform** menu controls how adjacent components are scaled, rotated, and translated in **Local** mode. It has no effect when using other manipulation modes.

- When off, adjacent components are transformed according to their averaged reference frame in Local mode and a single MoveComponent operator is applied.

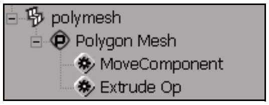
Transform Local Components Separately = Off



Adjacent components share an average reference frame.



This is the result of translating the polygons along the Y axis of their common reference frame.



polymesh
Polygon Mesh
MoveComponent
Extrude Op

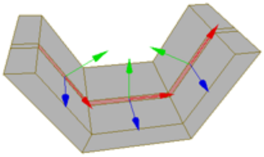
A single MoveComponent operator is applied.

However, disjoint clusters of connected components are treated separately and one MoveComponent operator is applied per cluster.

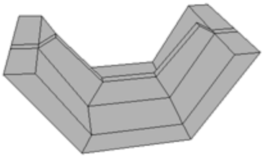
- When on, adjacent components are not treated as a single cluster in Local mode. Instead, each component is transformed separately according to its individual reference frame, one after the other, and a separate MoveComponent operator is applied for each component.

  Because the transformations are applied sequentially when **Transform Local Components Independently** is on, the local reference frame of one component may change before it gets transformed. This means that you might not get good results in some situations, such as when scaling edges and so on.

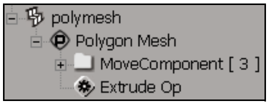
Transform Local Components Separately = On



Adjacent components each use their own local reference frame. Note that Display Multiple Transformation Axes is also on.



This is the result of translating the polygons along their Y axes individually.



Multiple MoveComponent operators are applied.

When **COG** is on, it always takes precedence: all components are treated as a single cluster, whether adjacent or not, and a single MoveComponent operator is applied.

Setting Values Numerically

As an alternative to transforming objects interactively, you can enter numerical values in the boxes on the Transform panel.

- In Global mode, values are relative to the scene origin.
- In Ref mode, values are relative to the active reference plane.
- In all other modes, values are relative to the object's center.

Like other input boxes, you can enter specific or relative values for one or more selected objects. For multiple objects, you can also use random values or specify a range of linear values. See *Entering Values in a Text Box* on page 86.

When you set values, selected components are transformed based on their center of geometry, keeping their relative positions to each other. The exception is when entering translation values for points when **COG** is off and **Transform > Collapse Points For Snapping/Absolute Translation** is on: in this case, all the points are set individually to the specified values.

Animating Component and Cluster Transformations

You cannot animate component and cluster transformations directly. Instead, you can use a deformer such as a cluster center (see *Cluster Centers* on page 127) or volume deformer (see *Manipulating Points by Volume* on page 121) and animate it. Another alternative is to use shape animation (see *Chapter 4: Shape Animation* in the *Nonlinear Animation* guide).

When working in Animation construction mode, you should create a cluster and transform that rather than transforming selected components or moving points directly. Otherwise, the MoveComponent operator may not update properly if you later modify the topology.

This is not an issue for MoveComponent operators in the Shape Modeling or Secondary Shape Modeling regions. Once the MoveComponent operator has been baked into a shape, the shape will update properly when the topology is modified.

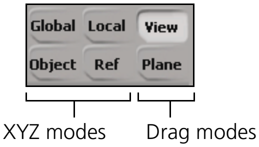
Translating Components and Clusters

You can translate selected components and clusters by pressing **v** or clicking the **t** button in the Transform panel to activate the Translate tool.

This section describes the manipulation modes and other options that are available for translating components and clusters. For the general procedure of transforming components and clusters, see *Basics of Transforming Components and Clusters* on page 93.

Component Translation Modes

Translation Modes

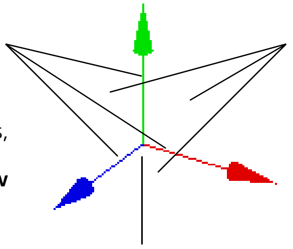


The translation modes are divided into two groups: the XYZ modes and the drag modes. When you are not using the SRT manipulators, the Global, Local, Object, and Ref modes use the left, middle, and right mouse buttons to transform along the separate X, Y, and Z axes respectively. The View and Plane modes use the left mouse button to “drag” across the corresponding plane.

- **Global** translations are performed along the scene’s global axes.
- **Local** translations are performed along the axes of the components’ or clusters’ own reference frame. For some more information that is specific to Local mode, see *Transforming Components and Clusters in Local Mode* on page 97.
- **View** transformations are performed with respect to the viewing plane of the 3D view.
- **Object** transformations are performed in the local coordinate system of the object that contains the components.
- **Ref** and **Plane** transformations can be performed in any default plane (XY, YZ, XZ, or View) as well as any arbitrary reference plane that you specify. For more information, see *Chapter 17: Reference Planes* of the *Fundamentals* guide.

Using the Translate Manipulator

When using SRT manipulators to translate, there are seven handles that you can click and drag:



Click and drag on a single axis to translate along it.

Double-click to toggle between a single active axis and all three axes, or Shift+double-click to toggle activeness of a single axis (If **Allow Double-Click to Toggle Active Axes** is on in Transform preferences).

Click and drag between two axes to translate along the corresponding plane.

Click and drag on the center to translate in the viewing plane.

In addition to dragging the handles, you can:

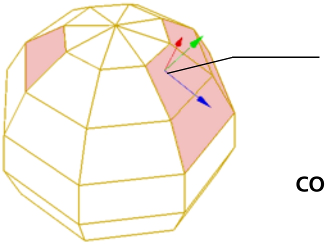
- Middle-click and drag anywhere in the 3D views to translate along the axis that most closely matches the drag direction.
- Click and drag anywhere in the 3D views (except on the manipulator) to perform different actions, depending on the setting for **Click Outside Manipulator** in the **Tools** > **Transform** preferences. See *To choose the behavior when clicking outside of the manipulators* on page 96.
- Right-click over the manipulator to open a context menu, where you can set the manipulation mode and other options.

For information about using the SRT manipulators in general, see *Transforming Interactively with the SRT Manipulators* on page 96.

Component Translation and COG Mode

COG (center of geometry) controls whether non-adjacent components are translated together or separately.

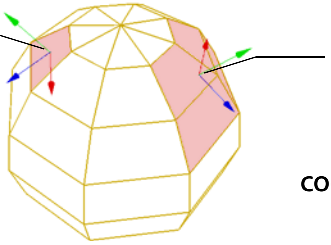
- When it is on, all components are translated along the axes of their average reference frame in Local mode.



All components share a single average reference frame.

COG = on

- When it is off, non-adjacent components are translated separately along the axes of their own reference frames in Local mode. Adjacent components are translated together as a single cluster (if **Transform** > **Transform Components Independently in Local Mode** is also off).



Non-adjacent components have their own reference frame.

Adjacent components share a single average reference frame.

COG = off

Component Translation and Snapping

When snapping, you can specify which relative location snaps to the target. For points, you can also control whether they snap as a group or all collapse onto the same target.

For more information about snapping in general, see *Chapter 18: Snapping* of the *Fundamentals* guide.

Specifying the Location that Snaps to the Target

When translating, the position that snaps to the target is the pivot. When components are selected, this is normally the center of geometry. However, you can set the pivot manually as described in *Setting the Manipulation Pivot* on page 360—this is particularly useful when you are translating several selected points and want one point to snap to the target while the other points maintain their relative offsets.

Collapsing Multiple Points to the Same Target

When snapping multiple selected points, they normally maintain their relative positions, whether or not **COG** is on. However, you can force all selected points to snap individually to the same target if **COG** is off and **Transform** > **Collapse Points For Snapping/Absolute Translation** is on.

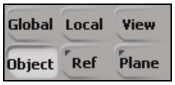
Rotating Components and Clusters

You can rotate selected components and clusters by pressing **c** or clicking the **r** button in the Transform panel to activate the Rotate tool.

This section describes the manipulation modes and other options that are available for rotating components and clusters. For the general procedure of transforming components and clusters, see *Basics of Transforming Components and Clusters* on page 93.

Component Rotation Modes

Rotation Modes



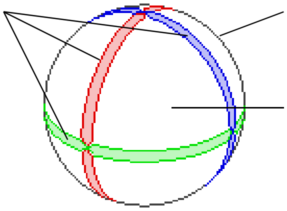
The rotation mode determines the axis of rotation.

- **Global** rotations use one of the global axes.
- **Local** rotations use one of the local cluster reference frame axes. For some more information that is specific to Local mode, see *Transforming Components and Clusters in Local Mode* on page 97.
- **View** rotations use an axis perpendicular to the viewing plane of the 3D view you are manipulating in.
- **Object** rotations use one of the parent object's local axes.
- **Ref**, or reference, mode lets you rotate elements using the X, Y, or Z axis of another element or an arbitrary reference plane. For more information, see *Chapter 17: Reference Planes* of the *Fundamentals* guide.
- **Plane** mode lets you rotate elements using an axis (by default, Y) of another element or an arbitrary reference plane. For more information, see *Chapter 17: Reference Planes* of the *Fundamentals* guide.

When you are not using the SRT manipulators, the Global, Local, Object, and Ref modes use the left, middle, and right mouse buttons to rotate around the X, Y, and Z axes respectively. The View and Plane modes use the left, middle, and right buttons to rotate about the plane normal (by default, the Y axis) at slow, medium, or fast speed.

Using the Rotate Manipulator

When using SRT manipulators to rotate, there are five handles that you can click and drag:



Click and drag on a single ring to rotate around that axis.

Double-click to toggle between a single active axis and all three axes, or Shift+double-click to toggle activeness of a single axis (If **Allow Double-Click to Toggle Active Axes** is on in Transform preferences).

Click and drag on the silhouette to rotate about the viewing axis.

Click and drag on the ball to rotate freely.

In addition to dragging the handles, you can:

- Middle-click and drag anywhere in the 3D views to rotate about the axis that most closely matches the drag direction.
- Click and drag anywhere in the 3D views (except on the manipulator) to perform different actions, depending on the setting for **Click Outside Manipulator** in the **Tools** > **Transform** preferences. See *To choose the behavior when clicking outside of the manipulators* on page 96.
- Right-click over the manipulator to open a context menu, where you can set the manipulation mode and other options.

For information about using the SRT manipulators in general, see *Transforming Interactively with the SRT Manipulators* on page 96.

Component Rotation and COG Mode

COG (center of geometry) determines the default pivot point used for rotating components and clusters.

- When it is on, components rotate about the center of their average reference frame in all modes.
- When it is off, components rotate about the center of their own reference frame in Local mode—adjacent components share an average reference frame if **Transform** > **Transform Components Independently in Local Mode** is off.

  In Global, Object, View, and Plane modes, they rotate about the center of the object that contains them. In Ref mode, they rotate about the origin of the reference plane.

You can also set the pivot manually—see *Setting the Manipulation Pivot* on page 95.

Scaling Components and Clusters

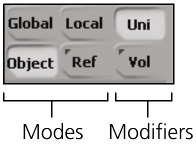
You can scale selected components and clusters by pressing **x** or clicking the **s** button in the Transform panel to activate the Scale tool.

This section describes the manipulation modes and other options that are available for scaling components and clusters. For the general procedure of transforming components and clusters, see *Basics of Transforming Components and Clusters* on page 93.

You can align components by scaling them to zero along a particular axis. One way to do this quickly is to press Ctrl while scaling to snap by increments. For more information about incremental snapping, see *Incremental Snapping* in Chapter 18 of the *Fundamentals* guide.

Component Scaling Modes

Scaling Modes



Scaling components moves them away from or towards a pivot point. The scaling mode determines the axes along which to scale. If you scale non-uniformly in any mode other than Object, you may get a shearing effect.

- **Global** mode scales along one of the global axes.
- **Local** mode scales along one of the local cluster reference frame axes. Note that as you transform components, their local axes may get recomputed.
- **Object** mode scales along one of the parent object’s local axes.
- **Ref**, or reference, mode scales along the X, Y, or Z axis of another element or an arbitrary reference plane. For more information, see *Chapter 17: Reference Planes* of the *Fundamentals* guide.

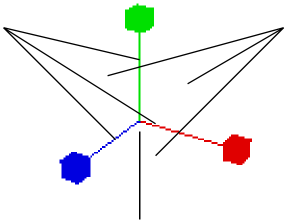
In addition to the four modes, there are two modifiers that affect how components are scaled:

- **Uni**, or uniform, scales along all active axes at the same time with a single mouse button.

  This avoids the annoyance of trying to drag the mouse while pressing two or three buttons. You can activate and deactivate axes as described in *Specifying Axes* on page 94. You can also temporarily turn on **Uni** by pressing Shift while scaling.
- **Vol**, or volume, scales along one or two local axes, while automatically compensating the other axes so that the volume of a bounding box around the selected elements remains constant.

Using the Scale Manipulator

When using SRT manipulators to scale, there are seven handles that you can click and drag:



Click and drag on a single axis to scale along it.

Double-click to toggle between a single active axis and all three axes, or Shift+double-click to toggle activeness of a single axis (If **Allow Double-Click to Toggle Active Axes** is on in Transform preferences).

Click and drag along the diagonal between two axes to scale both those axes uniformly.

Click and drag the center left or right to scale all active axes uniformly.

In addition to dragging the handles, you can:

- Middle-click and drag anywhere in the 3D views to scale along the axis that most closely matches the drag direction.
- Click and drag anywhere in the 3D views (except on the manipulator) to perform different actions, depending on the setting for **Click Outside Manipulator** in the **Tools** > **Transform** preferences. See *To choose the behavior when clicking outside of the manipulators* on page 96.
- Right-click over the manipulator to open a context menu, where you can set the manipulation mode and other options.

For information about using the SRT manipulators in general, see *Transforming Interactively with the SRT Manipulators* on page 96.

Component Scaling and COG Mode

COG (center of geometry) determines the default pivot point used for scaling components and clusters.

- When it is on, components scale about the center of their average reference frame in all modes.
- When it is off, components scale about the center of their own reference frame in Local mode—adjacent components share an average reference frame if **Transform** > **Transform Components Independently in Local Mode** is off.

  In Global and Object modes, they scale about the center of the object that contains them. In Ref mode, they scale about the origin of the reference plane.

You can also set the pivot manually—see *Setting the Manipulation Pivot* on page 95.

Moving Points

As an alternative to translating points, you can use the Move Point tool. The Move Point tool lets you translate whatever point is under the mouse pointer just by clicking and dragging—you do not need to select the point first.

The Move Point tool uses the same manipulation modes as the Translate tool—for more information, see *Component Translation Modes* on page 101. When you activate the tool, it automatically remembers the last mode used.

You can change the mode and other options even while using the Move Point tool in supra mode simply by keeping the **m** key pressed while clicking buttons on the Transform panel.

You can use **Prop** to activate proportional modeling so that neighboring points are also adjusted with a falloff based on distance—see *Using Proportional Modeling* on page 112. You can also use **Sym** for symmetry—see *Manipulating Components Symmetrically* on page 117.

The Move Point tool also makes use of the **Raycast in Shaded Display Mode** option in the **Tools** > **Selection** preferences. When that option is on, the Move Point tool ignores points that are occluded (hidden behind other elements) in Hidden Line Removal, Constant, Shaded, Textured, and Textured Decal display modes. However, the option is ignored if **Override Object Properties** is off in the Display type menu of a 3D view.

Using the Move Point Tool

As an alternative to selecting and translating points, you can move points individually using the Move Point tool:

1. Select a geometric object.
2. Activate the Move Point tool by doing one of the following:
   - Choose **Modify** > **Component** > **Move Point** tool from the Model toolbar.

     or

   - Quickly press and release **m** to activate the Move Point tool in sticky mode.

     or

   - Press and hold **m** to move points in supra mode.
3. In a 3D view, position the mouse pointer over a point on the object, then click and drag to move it.

   On polygon meshes, you can press Alt to snap and weld one point to another while moving it. See *Welding Vertices* on page 221.

To move a point along its own normal, you can translate along the Y axis in **Local** manipulation mode. If you find that the normal updates as you move the point, you can use a reference plane instead:

1. Select the point.
2. Activate the Translate tool.
3. Activate **Ref** mode.
4. Alt+click on the point. This sets the pivot and, because Ref mode is already active, updates the current reference frame.

   Don't do this while the Move Point tool is active because the Alt key is used to weld points in that case (polygon meshes only).
5. Translate the point along Y.

Setting the Picking Region Size

You can set the size of the picking region that determines whether a point is "under" the mouse pointer when using the Move Point tool.

1. Choose **Transform** > **Transform Preferences**.
2. Set **Move Point Pick Size** to the desired value in pixels.

Notes and Tips for Moving Points

Here is some information that may be useful when using the Move Point tool.

Moving Points That Overlap Others

Polynode bisectors can be useful when selecting or moving points that overlap others on a polygon mesh object.

To select or move a point that overlaps others

1. Open the Camera Visibility property editor of a 3D view, for example, by pressing Shift+s.
2. On the Components tab, set **Show Polynode Bisectors** to **Show While Drawing Points**.

   With this option, bisectors are displayed as small purple lines at the corners of polygons whenever points are visible.
3. Activate point selection (for example, press **t** for Rectangle Point Selection) or the Move Point tool (press **m**).
4. Click on a bisector to affect the corresponding point.

Moving Points in Animation Mode

When working in Animation construction mode, you should create a cluster and transform that rather than transforming selected components or moving points directly. Otherwise, the MoveComponent operator may not update properly if you later modify the topology.

This is not an issue for MoveComponent operators in the Shape Modeling or Secondary Shape Modeling regions. Once the MoveComponent operator has been baked into a shape, the shape will update properly when the topology is modified.

Relative Mode

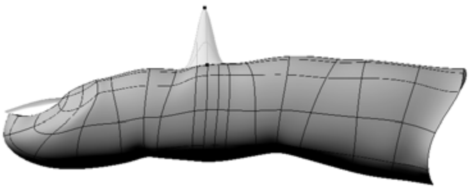
You can use relative mode to store component transformations in terms of the component's own coordinate system, rather than the parent object's coordinate system.

No matter which manipulation mode is used for transforming components, the results are stored relative to the object center by default. If you move the object, the components' offsets are maintained. However if you deform the object, the transformations may not be the same relative to the surrounding components. This may cause undesired results.

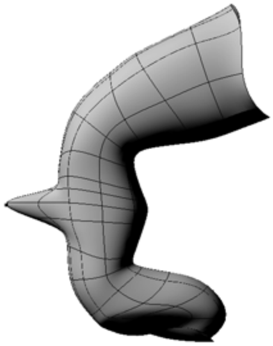
In relative mode, the transformations are stored in terms of the component's local coordinate system. If the object is deformed, the components maintain their transformation relative to the surrounding components.

To activate relative mode

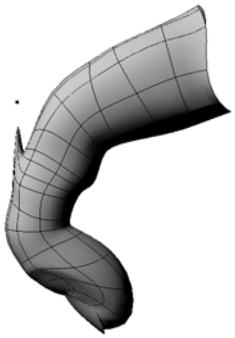
- Turn **Modify** > **Component** > **Relative Mode** on before moving components.



Point moved on an object.



Relative Mode On
Point is offset relative to surrounding points when object is deformed.



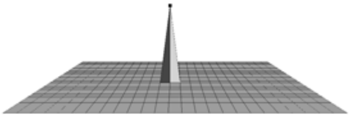
Relative Mode Off
Point is offset relative to object's center when object is deformed.

Remember that when you transform components, their local axes may get recomputed.

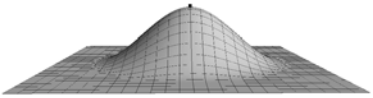
Using Proportional Modeling

When you move or transform points and point clusters, you can use proportional modeling. When this option is on, neighboring points are affected as well, with a falloff that depends on distance. Points that will be affected are highlighted in the 3D views. You can set the proportional options for future edits, as well as modify them for previous edits.

Proportional modeling is sometimes known as “magnet” or “soft selection”.



Proportional modeling off

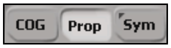


Proportional modeling on

When rotating in proportional mode, affected points spiral in towards the center of rotation.

To activate proportional modeling

Do one of the following:



- Click the **Prop** button on the Transform panel.

  or

- Choose **Modify** > **Component** > **Proportional** from the Model toolbar.

Points that are affected by the proportional falloff are highlighted, and when using the Move Point tool, the Distance Limit is displayed as a circle.

Proportional Modeling Settings

The proportional modeling settings control things such as the distance limit for affected points and the shape of the falloff profile. You can modify all of these settings in the Proportional property editor, and in addition, you can change the distance limit interactively as you work.

Modifying Proportional Modeling Settings

You can modify the proportional settings before activating proportional modeling and the settings will be used for future proportional operations. You can also go back and adjust the settings used for previous operations in the operator stack.

To change proportional settings for future operations

To adjust the distance, falloff profile, and other proportional settings, do one of the following:

- Right-click the **Prop** button on the Transform panel.

  or

- Choose **Modify** > **Component** > **Proportional Setup**.

The Proportional property editor opens. The new settings will be used for future actions.

To adjust proportional settings for previous operations

1. Select the object.
2. Do one of the following:
   - Choose **Edit** > **Properties** > **Modeling Properties** from the Edit panel and click the Proportional tab. There is one Proportional property page for each MovePoint operation with proportional on.

     or

   - Choose **Selection** on the Select panel, expand a MovePoint node, and click the Proportional icon.
3. Adjust the parameters. Click the Help icon (**?**) for details.

Setting the Distance Limit

The distance limit controls the size of the virtual sphere that determines which nearby points will be affected by the proportional operation. The points that are within the distance limit are highlighted in the 3D views. You can adjust the distance limit interactively or by increments as you work—these affect the value used for future operations.

To change the Distance Limit interactively

You can change the **Distance Limit** interactively when proportional modeling is active and either the Move Point tool is active or one of the transform tools (Scale, Rotate, Translate) is active in Point mode.

- Press and hold **r** while dragging the mouse to the left to decrease the **Distance Limit** or to the right to increase it.

To change the Distance Limit by increments

You can change the **Distance Limit** by increments when proportional modeling is active and either the Move Point tool is active or one of the transform tools (Scale, Rotate, Translate) is active in Point mode. Do one of the following:

- Press the up arrow key to increase the Distance Limit by 10%.

or

- Press the down arrow key to decrease the Distance Limit by 10%.

or

- On Windows systems, you can also roll the mouse wheel up or down to increase or decrease the Distance Limit by increments of 10%. You may need to click first to make sure that the 3D view has focus.

Reference

Reference specifies the coordinate system used to determine the distance of nearby points:

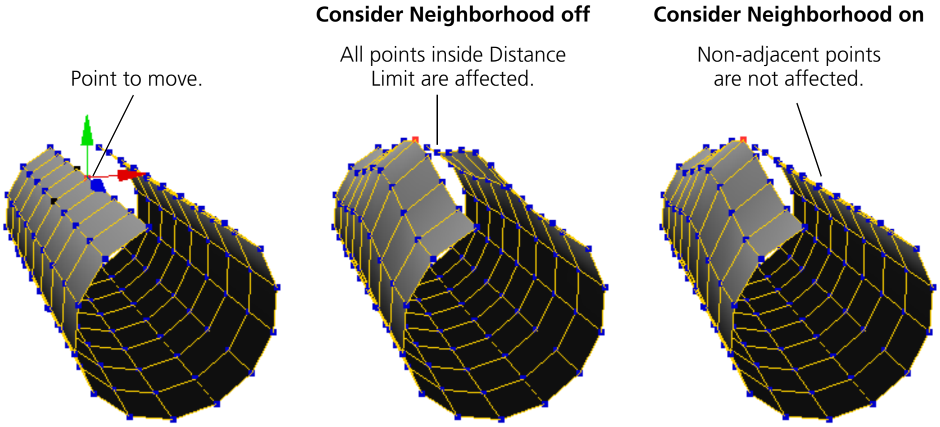
- **Global** uses the scene's coordinate system. With this option, the Distance Limit is in Softimage units.
- **Local** uses the object's local coordinate system. In particular, the Distance Limit is affected by the object's scaling, and if the object has been scaled non-uniformly, then the limit of the effect will be different in different axes.

Consider Neighborhood

Consider Neighborhood provides further control over which points are affected by considering the adjacency of affected points, in addition to their distance from the points that are being directly manipulated:

- If it is off, all points within the Distance Limit are affected.
- If it is on, a point within the Distance Limit is not affected if it is not adjacent to points that are being directly manipulated, nor adjacent to other affected points.

This option is particularly useful when working on objects that curve back on themselves.



Consider Neighborhood off

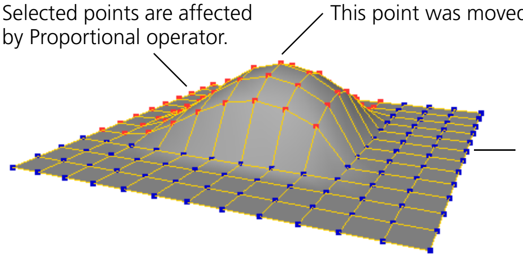
Consider Neighborhood on

Point to move.
All points inside Distance Limit are affected.
Non-adjacent points are not affected.

Tagged Points Only

Tagged Points Only limits the points that are affected by the Proportional operator to those that are currently selected. This option works only when using the Move Point tool to manipulate individual points with Proportional on; it has no effect when transforming selected points.

Tagged Points Only on



Selected points are affected by Proportional operator.

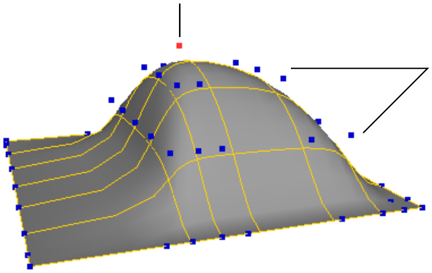
This point was moved.

Unselected points are not affected.

To use Tagged Points Only, you must activate it using **Component > Proportional Setup** before moving a point. When you modify a previous operator, this option will not appear in the Proportional property editor if it was not already on when the operator was originally applied.

Surface Options

The Surface Options apply only to NURBS surfaces. They use the NURBS parameterization instead of distance to determine which nearby points to affect.



This point was moved.

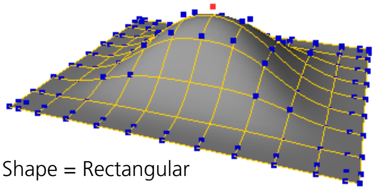
Neighboring points are affected by the same amount even though they are at different distances from the moved point.

When **Use U/V Limits for Surfaces** is on, the Surface Options are used when manipulating points on NURBS surfaces. Other types of geometry (polygon meshes, lattices, curves, hair, particles, and so on) continue to use the settings in the Proportional Options area.

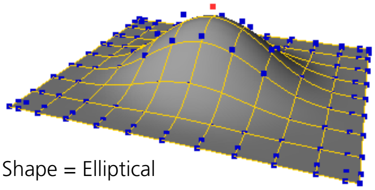
The **U** and **V Limit** options set the number of points to affect in the positive and negative directions along each parameter.

Shape controls the outline of the affected area:

- **Rectangular** produces a box shape the size of the U and V Limits.
- **Elliptical** modulates the falloff at the corners of the box. resulting in an oval area of effect.



Shape = Rectangular

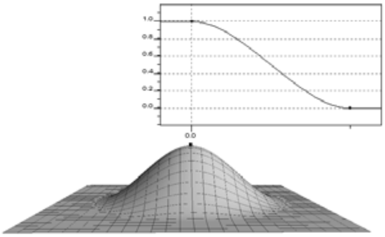


Shape = Elliptical

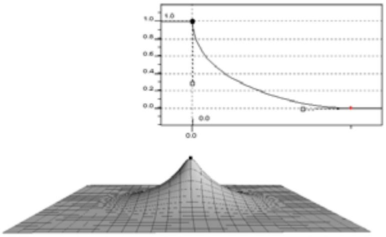
Falloff Profile

The **Falloff Profile** controls the shape of the Proportional effect by determining the amplitude of the displacement over the length of the Distance Limit or U and V Limits. A value of 1 means that points at the corresponding distance move the same amount as the selected points; a value of 0 means that they do not move at all.

You can select the curve and then edit it using the mouse and the same keyboard commands as the animation editor or right-click to display a menu. For more information, click ? to display the online help.



Using default profile



Using modified profile

Manipulating Components Symmetrically

Symmetrical manipulation lets you manipulate points and other components and maintain the symmetry of an object. It is a two-stage process:

1. Applying a symmetry map.
2. Activating **Sym** while manipulating components.

While **Sym** is active, any manipulation performed on components on one side are mirrored to the corresponding components on the other side. Components that lie directly on the plane of symmetry are “locked down”; they can be translated or moved only along the plane of symmetry.

Tools that Work with Sym Mode

Manipulation in Sym mode works with the following tools:

- Scale tool on points, edges, and polygons (**x** key)
- Rotate tool on points, edges, and polygons (**c** key)
- Translate tool on points, edges, and polygons (**v** key)
- Move Point tool (**m** key)

Applying Symmetry Maps

A symmetry map establishes the correspondence between points on either side of the plane of symmetry.

To apply a symmetry map

1. Select a geometric object, cluster, or points.
2. Choose **Get** > **Property** > **Symmetry Map** from any toolbar. A property editor opens.
3. On the Symmetry Map Generator property page, set the Symmetry Plane to the desired value: **YZ** (X = 0), **XZ** (Y = 0), or **XY** (Z = 0). The symmetry map uses the object’s local center and the chosen plane to establish a correspondence between points.

Symmetry maps are not updated after topological operations. If you make any modifications that create or remove components on the object, you should delete and reapply the symmetry map.

Activating Symmetrical Manipulation

Once you have applied a symmetry map to an object, you can activate **Sym** to manipulate components symmetrically.

To activate or deactivate symmetrical manipulation

- Click **Sym** on the Transform panel.

- Symmetry maps are not updated automatically when you add new components. However, you can delete the old symmetry map and apply a new one.
- The symmetry options that you can set by right-clicking on **Sym** have no effect when manipulating components. These options are used when drawing curves and chains, as well as when painting on weight maps.

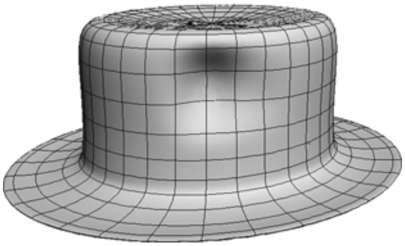
Push Paint

You can push points out or in along their normals using the Push Paint tool. This lets you sculpt an object by painting.

What the Push Paint tool actually does under the hood is:

- Create a weight map.
- Connect it to a Push deformation.
- Activate the Paint tool in a single operation.

See *Chapter 6: Deformations* on page 131 for more information about Push and other deformations in general, and see *Chapter 11: Weight Maps* in the *Fundamentals* guide for more information about weight maps and painting weights.



Applying Push Paint

To apply push paint

1. Select the element that you wish to sculpt:
   - If you select a cluster, the weight map will be applied to it.
   - If you select an object, a cluster will be created for all the points on the object and the weight map will be applied to the cluster.
   - If you tag (select) points, a cluster will be created and the weight map will be applied to the cluster.
2. Choose **Get** > **Property** > **Paint Push Tool**.

   A weight map is created and its property editor is displayed. A Push deformation is applied and the Paint tool is automatically activated.
3. In a 3D view, click and drag to push points. In normal (additive) paint mode:
   - To push points out, use the left mouse button.
   - To push points in, either use the right mouse button or press Shift while using the left mouse button.
   - To smooth, press Alt while using the left mouse button.

- To change the brush radius, press **r** while dragging with the left mouse button.
- To change the opacity, press **e** while dragging with the left mouse button. The opacity controls the amount of weight added or removed with each brush stamp.
- To change other brush properties, display the Brush Properties editor by choosing **Get** > **Property** > **Brush Properties** or pressing Ctrl+w.

When painting, you can increase performance by reducing the geometry-approximation settings. The Paint tool uses the triangulation of the object to follow its surface.

For more information, see *Geometry Approximation on NURBS* on page 356.

Modifying Push Paint

After you have used Push Paint, you can modify the effect by painting more strokes at a later time or by changing the default parameters of the underlying Push operator.

Painting More Strokes

If you want to resculpt the object at a later time, you can select the original weight map and reactivate the Paint tool.

To paint again later

1. Select the deformed object.
2. On the Select Panel, choose **Explore** > **Property Maps** and pick the associated weight map (by default, named **PushMap**).
3. Activate the Paint tool by pressing **w** or by choosing **Get** > **Property** > **Paint Tool**.

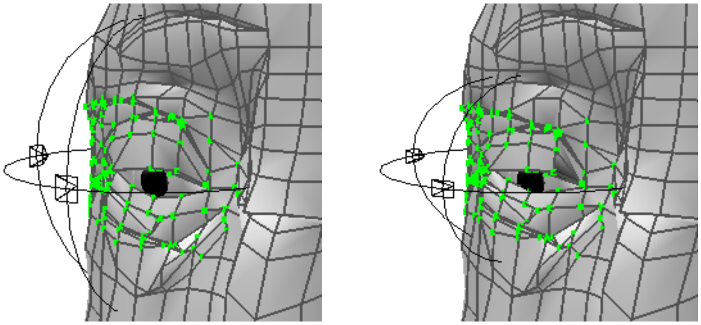
Setting Push Parameters

To change the default parameters of the Push operator

1. Select the deformed object.
2. On the Select Panel, choose **Selection** and then click the Push Op icon. The Push Op property editor opens.
3. Modify the parameters as desired. Refer to Online Help by clicking on the ? in the property editor for specific information about each parameter.

Manipulating Points by Volume

Volume deformations are a modeling tool consisting of a spherical control object that you can use to move an object's points. Only points within the control object are affected, with a proportional falloff from the center.



Scaling a volume deformer applied to a cluster.

About Volume Deformations

Volume deformations have two parts:

- A volume-deform object, or volume deformer, is a control object you can use to deform geometric objects. It is represented by a spherical icon with little "clamps" in the 3D views. Points that lie inside the sphere's volume are affected by the deformation.
- A Proportional Volume operator is applied to a geometric object when you attach and transform a volume deformer.

Volume Deformers

Volume deformers are spherical shapes that define the points that are affected on the deformed object.

To get a volume deformer

1. If desired, first select some objects, clusters, or points to deform. The control object will be automatically attached to the selected elements.
2. Choose **Get** > **Primitive** > **Control Object** > **Volume Deform** from any toolbar. A volume deformer is added to the scene.
   - If any geometric elements were selected, the volume deformer is attached to them. A Proportional Volume operator is automatically applied and its property editor opens.
   - If no geometric elements were selected, the volume deformer is initially attached to nothing and its property editor opens instead.

To set volume deformer properties

Open the volume deformer's property editor. A quick way to do this is to select the volume deformer and press Enter.

- **Radius** controls the untransformed size of the volume deformer.
- **Remember Attached Objects** controls what happens when you attach the volume deformer.
  - If it is on, the deformation is applied to the same elements (objects or clusters) as before.
  - If it is off, the deformation is applied to all visible geometric objects that are wholly or partially within the control object's volume.

    You can exclude geometric objects by hiding them before you attach the control object. Alternatively, you can attach elements manually as described in *Attaching Geometry Manually* on page 123.

Applying Volume Deformations

To deform by volume, you first use the volume deformer to define the volume of influence and then you apply the volume deformation operator.

If you want to animate the volume deformer's transformation to change the shape of the geometry, leave the volume deformer attached. Otherwise, you can use the volume deformer as a modeling tool by detaching it, defining a new volume of influence, applying another volume deformation operator, and repeating.

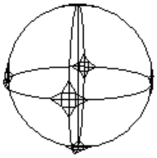
To attach or detach a volume deformer

- Select the volume deformer and do one of the following:
  - Press Shift+v.

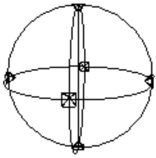
    or

  - Choose **Modify** > **Component** > **Attach/Detach Volume Deform** from the Model toolbar.

  The volume deformer's icon changes to show whether it is attached or detached. When you attach it, the geometries that are affected depend on the deformer's Remember Attached Objects parameter—see *To set volume deformer properties* on page 122.



Detached.



Attached.

To apply a volume deformation

1. Make sure that the volume deformer is not attached.
2. Define the points to affect by adjusting the size, shape, and position of the volume-deform object.

   For example, you can scale, rotate, and translate the object, as well as adjust its radius. Once the deformer is attached, it will affect only those points within its volume.

To apply a volume deformation

1. Attach the volume-deform object.
2. Scale, rotate, and translate the volume-deform control object. Points within the volume on the attached elements are deformed accordingly.

   As you continue to transform the control object, no new Proportional Volume operator is applied; instead, the current operator is updated. You can adjust its falloff and other properties as described in *To modify the current (last) operator* on page 124.
3. When you are satisfied, you can detach the volume-deform control object, and then repeat the entire procedure to continue sculpting your objects.

Attaching Geometry Manually

You can attach geometric elements to volume-deform objects manually. This is useful, for example, if you want to restrict a deformation to a particular cluster.

To manually attach geometric elements to volume-deform objects

1. Select one or more objects, clusters, or points.
2. Choose **Modify** > **Deform** > **by Volume** from the Model toolbar. This command is also available on the Animate and Simulate toolbars.
3. Pick a volume-deform control object.

   - If the control object was not attached to anything, it becomes attached to the selected elements.
   - If the control object was already attached, the selected elements are added to the list of attached geometries.

Setting Volume Deform Operator Options

The Proportional Volume operator controls parameters like the strength and falloff of the deformation. You can modify the current operator or set the defaults for future operators in your scene. Of course, you can always use an explorer to open the property editor of any previously applied operator as well.

To modify the current (last) operator

- With the volume deformer selected, choose **Modify** > **Component** > **Inspect Volume Deform** from the Model toolbar.

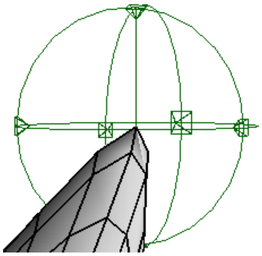
To modify settings for future operators

- Choose **Modify** > **Component** > **Volume Deform Setup** from the Model toolbar.

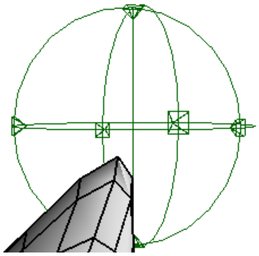
Power

Power controls how closely the deformed points follow the center of the volume deformer as it is transformed.

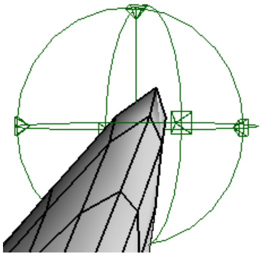
- When Power is 100, the points follow exactly.
- When Power is 50, the points follow halfway.
- When Power is greater than 100, the points overshoot.
- When Power is negative, the points move in the opposite direction.



Power = 100



Power < 100

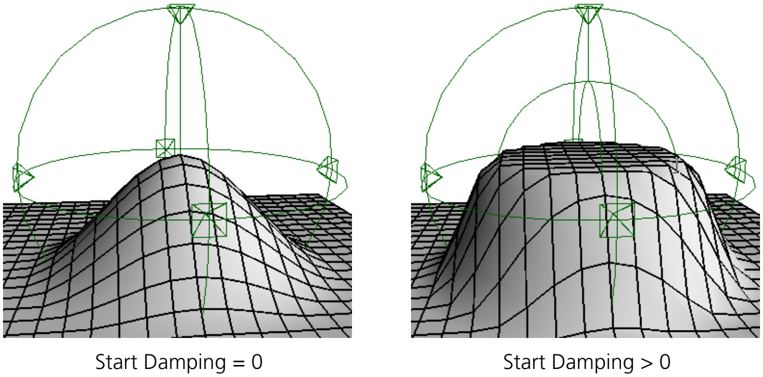


Power > 100

Start Damping

Start Damping defines a radius within which the deformation has full power and after which it falls off. Values are a percentage of the volume deformer’s radius.

- At 0, the Power falls off from the center.
- At 100, all points within the volume are deformed with full Power.
- Between these values, an inner sphere shows the radius at which the falloff begins.



Start Damping = 0

Start Damping > 0

Envelope Mode

When Envelope Mode is on, the deformation is relative to the volume deformer’s parent. This option is useful if you are using volume deformers parented to envelope deformers to create bulging muscles. The volume deformer will not affect the envelope as it moves with its parent, but only when it transforms relative to its parent.

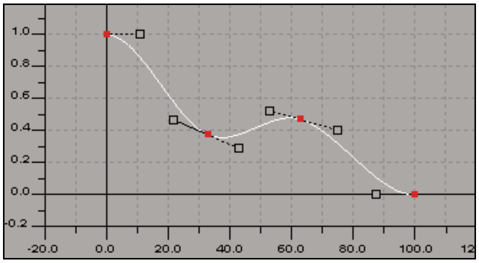
Interpolation

Interpolation controls how the deformation is calculated.

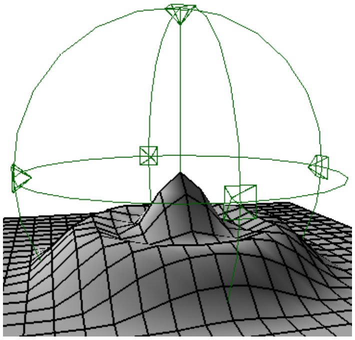
- **Linear** is faster to calculate.
- **Transform** may give better results, particularly when the volume deformer is rotated.

Falloff Profile

Falloff Profile controls how the Power falls off from the Start Damping radius to the end of the volume deformer's radius. You can edit the profile curve using the mouse and the same keyboard commands as the animation editor or right-click to display a menu.



Falloff Profile



Animating Volume Deformations

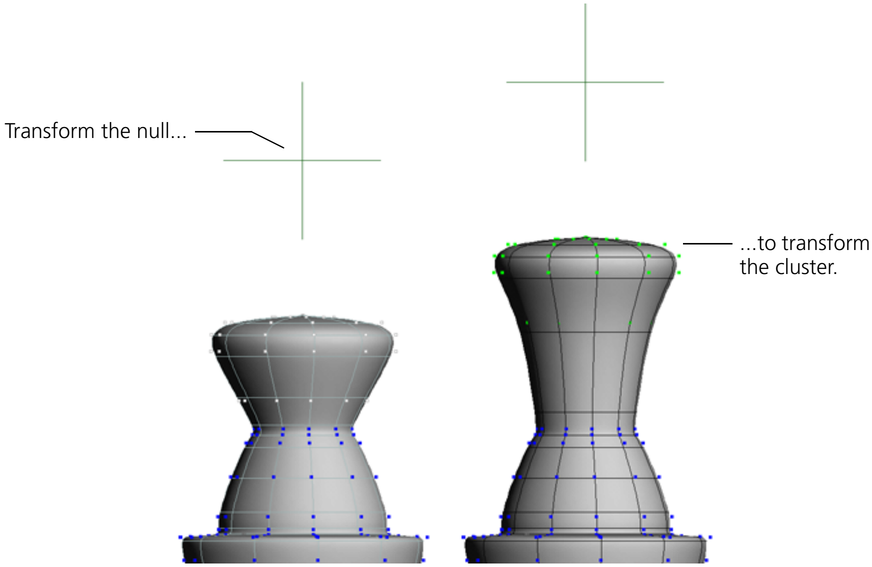
If you wish to animate an object's deformation by animating the transformations of a volume deformer, then you must leave the volume deformer attached.

Note that you can also animate the Proportional Volume operator's properties, such as Power and Damping.

Cluster Centers

You can assign the center of a cluster to a deformer like a null or other object. The cluster is constrained to the center and you can deform the object by moving the center. You can animate the deformation by animating the center. This is especially effective if you use a weight map to create a falloff that modulates the amplitude of the deformation.

You can create a null and assign it as the center when you create a cluster or you can create a cluster and assign a center later.



You can use a weight map to control the falloff of the effect across the cluster. For information about attaching weight maps, see *Chapter 11: Weight Maps* in the *Fundamentals* guide.

- In SOFTIMAGE|XSI, cluster centers are implemented as a deformation operator. You can mute the deformation, use weight maps, and so on, like any other deformation.
- To achieve the same effect as SOFTIMAGE|3D when rotating, make sure that **SI3D Rotation** is on in the Cluster Center property editor—rotations are performed around the cluster center's center.

  When this option is off, rotations are performed around the cluster's local center.

Creating a Cluster with a Center Automatically

You can automatically create a cluster, create a null for its center, and a cluster center operator all at once with the **Edit** > **Create Cluster with Center** command.

However if you want to use an existing cluster or a deformer other than a null, you must use the procedure described in *Assigning a Cluster to a Center Deformer Manually* on page 128.

To create a cluster with a center

1. Select some components (points, polygons, or edges) on an object.
2. Choose **Edit** > **Create Cluster with Center** from the Edit panel. Several things happen:
   - A cluster is created with a default name based on the type of component, for example, "Point".
   - A null is created at the geometric center of the cluster, with a default name based on the object and component type, for example, "sphere_Point_ClsCtr". The null is a child of the object and is automatically selected.
   - A Cluster Center deformation is applied.
   - A property editor opens, allowing you to change the parameters of the Cluster Center operator and the cluster.
3. Modify the scaling, rotation, and translation of the null to affect the cluster. You can animate the null's transformations to change the shape of the object over time.

Assigning a Cluster to a Center Deformer Manually

If you don't want to create a new cluster, or if you don't want to use a null as the cluster center deformer, then you must assign the cluster to a deformer manually.

To assign a center deformer to a cluster

1. Create an object to act as the cluster center deformer. You can use any type of object; you are not restricted to nulls.
2. Scale, rotate, and translate the cluster center deformer into the desired "neutral" pose. In this pose, the deformer will have no effect on the cluster.
   - You can use the **Transform** > **Align Objects** command to align the deformer with elements in the scene, such as the cluster you want to affect.

     In this case, make sure to select the cluster or components first and set **Move** to **All Objects Except the First**. For more information about the Align Objects command in general, see *Aligning Objects* in Chapter 16 of the *Fundamentals* guide.

  - If the deformer is a geometric object with points, you can freeze its transformations using **Transform** > **Freeze All Transforms** before applying the cluster center deformation.

    This can be useful because it “zeroes out” the deformer’s transformations in the neutral pose.

3. Select some components (points, edges, or polygons) or a cluster on the object you want to deform. If components are selected, a cluster will be created automatically.
4. Choose **Deform** > **Cluster Center** from the Model or Animate toolbar.
5. Pick the deformer. The Cluster Center property editor opens and the deformer is automatically selected.
6. Modify the scaling, rotation, and translation of the deformer to affect the cluster. You can animate the deformer’s transformations to change the shape of the object over time.

If you make the deformer a child of the deformed object, you can select the object in branch mode to transform it without leaving the deformer behind.

Offsetting the Cluster Center

Because Cluster Center is a deformation rather than a constraint, you cannot use **CnsComp** (Constraint Compensation) to offset the deformer from the cluster. However, you can use the Scaling, Rotation, and Translation parameters in the Cluster Center property editor to define the offset you need.

In addition, when you assign a cluster center to a cluster manually, any existing offset between the deformer and the cluster is maintained—in other words, the cluster does not snap to the deformer when you apply the deformation.

Chapter 6 Deformations

Deformations

Deformations are operators that change the shape of geometric objects. This chapter covers:

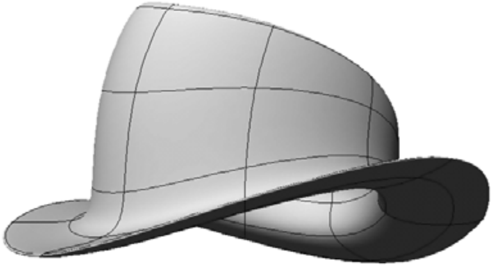
- Applying simple deformations like Bend and Twist.
- Deforming objects by curves and surfaces.
- Using lattices as a scaffolding to deform an object.
- Deforming an object by arbitrary geometric cages.
- Deforming an object by spine curves.
- Smoothing and relaxing objects.
- Shrinkwrapping one object onto another.
- Using waves to deform an object over time.
- Using Quickstretch to make an object stretch, flex, or yield as it moves.
- Muting a deformation to temporarily

Other types of deformation are described elsewhere:

- Direct manipulation of points and other components are actually implemented as deformations in XSI and are described in *Chapter 5: Manipulating Components* on page 89.
- Envelope deformations let you deform a character based on the movement of its skeleton. They are covered in *Chapter 3: Envelopes* of the *Character Animation* guide.
- Soft body dynamics let you simulate the behavior of non-rigid objects as they are subjected to forces and collisions. It is described in *Chapter 16: Soft Bodies* of the *Simulation* guide.
- Cloth lets you simulate how fabric moves and drapes. It is described in *Chapter 17: Cloth* of the *Simulation* guide.
- Freezing transformation and moving object centers are also deformations. These operations are described in *Chapter 16: Transformations* of the *Fundamentals* guide.
- Scripted operators let you create your own deformations by getting and setting the positions of an object's points. They are described in *Chapter 9: Scripted Operators* of the *Customization* guide.

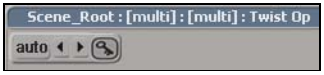
About Deformations

Deformations are powerful modeling and animation tools. You can apply deformations to objects such as polygon meshes, surfaces, and surface meshes, as well as to hierarchies and clusters of components. You can also apply deformations to lattices and particle clouds.



Deforming Multiple Objects

You can apply deformations to multiple selected objects. For example, you can select several surfaces and apply a Twist deformation. Each surface gets its own Twist operator, but they can all be controlled from the same Twist property editor—the title bar says “multi” to indicate a multiple selection.



You can still open each object’s individual Twist operator from its construction history and adjust parameters separately from the others.

You can reopen all the Twist operators and control them together by selecting them all in an explorer and pressing Enter.

Deformations on Hierarchies

An object with a hierarchy provides more possibilities for deformation animation than a single object. You can apply the deformation on the parent selected in branch mode (middle-mouse button) or tree mode (right-mouse button), in which case the deformation is transmitted to its children as if the hierarchy were a single piece.

All objects in the hierarchy share an instance of the same deformation operator. Opening the property editor from the construction history of any object and modifying parameters affects all objects in the hierarchy, and deleting the operator from any construction history deletes it from all of them.

Removing an object from the hierarchy removes the deformation from it, and adding a new object to the hierarchy applies the deformation to it.

Deformations on Groups

Like a hierarchy, applying a deformation to a group applies it to all members of the group. They all share an instance of the same deformation operator. Opening the property editor from the construction history of any member and modifying parameters affects all members, and deleting the operator from any construction history deletes it from all of them.

Adding an object to the group applies the deformation to it, and removing an object from the group removes the deformation.

Deforming with Weight Maps

If you apply a deformation when a cluster's weight map is selected, the weight map is automatically used to modulate the deformation's amplitude. Note that this technique does not work with all deformations, but in those cases you can still connect a weight map manually. For more information about weight maps in general see *Chapter 11: Weight Maps* in the *Fundamentals* guide.

To deform with weight maps

1. With a cluster's weight map selected, choose a deformation from the **Modify** > **Deform** menu on the Model toolbar.
2. In the deformation's property editor, adjust the parameters as desired.

After you have applied a deformation on a weight map, you can select the weight map again and then press **w** to add more paint strokes.

If an object has multiple maps, for example, a weight map controlling a Push deformation in addition to an envelope weight map, then you may need to select the desired weight map explicitly before you can paint on it. XSI remembers the last map selected or applied on each object.

A quick way to select the desired weight map is to select the object, then choose **Explore** > **Property Maps** from the Select panel and select the map to paint on.

Deformations in the Operator Stack

Deformations are applied as operators in the operator stack. If you do not freeze the operator stack, you can go back into the history and modify the deformations parameters—the changes percolate up through the history and are reflected in the deformed object.

Once you are satisfied with an object, you can freeze the operator stack. This discards the history and keeps only the current shape of an object, saving memory and computation time for updates.

For more information about modifying, reordering, collapsing, and freezing deformation operators, see *Operator Stack* on page 51.

Deformations and Textures

If an object has an implicit texture projection defined after a deformation, it will slip when the object is deformed. You can reorder the texture projection and deformation operators as described in *Changing the Order of Operators* on page 60. For more information about texture projections, see *Chapter 7: Projections and Supports* in the *Shaders, Lights & Cameras* guide.

Deformations and the Schematic

Objects with deformations are displayed with a letter **D** in the schematic view.

Animating Deformations

You can mark and animate many deformation parameters directly. The procedure is exactly the same as for any other animatable parameter and is described throughout the *Animation* guide. This allows you to create very simple shape animation, like an animated twist. (Of course if you freeze the operator stack, the animation is lost.)

In addition to animating deformation parameters directly, there are other tools for animating shapes:

- Shape animation using the animation mixer is described in *Chapter 4: Shape Animation* of the *Nonlinear Animation* guide.
- In particular, you can deform an object based on the orientation of another object or any other parameter as described in *Deforming a Shape Based on Poses or Orientation* in Chapter 4 of the *Nonlinear Animation* guide.
- Soft body dynamics let you simulate the behavior of non-rigid objects as they are subjected to forces and collisions. It is described in *Chapter 16: Soft Bodies* of the *Simulation* guide.
- Cloth lets you simulate how fabric moves and drapes. It is described in *Chapter 17: Cloth* of the *Simulation* guide.

Creating Smoother Deformations

Objects with large or long polygons might not deform smoothly. You can create smoother deformations by adding more geometry, or by using linear subdivision as described in *Chapter 9: Subdivision Surfaces* on page 325.

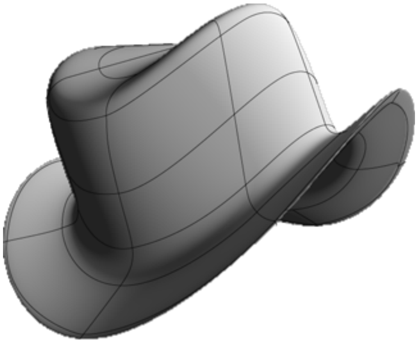
Simple Deformations

There are a variety of simple deformations you can quickly apply to objects. The parameters of each deformation are set in a single property editor and require no other input, so you can modulate the deformation easily. The deformations can be applied to objects, hierarchies, and clusters.

Simple deformations include:

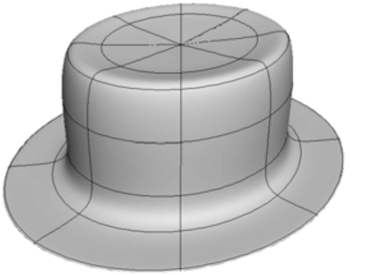
- Bend
- Bulge
- Shear
- Taper
- Twist and Vortex
- Push
- Randomize
- Shape Jitter

In addition, you can use the Push Paint tool to paint a Push deformation quickly and easily.



Applying Simple Deformations

You can apply a simple deformation to an object, hierarchy, or cluster. With clusters, you can also modulate the deformation with a weight map.



Undeformed hat

To apply a simple deformation

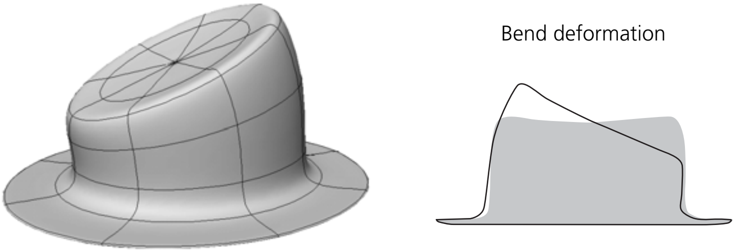
1. Select the object, hierarchy, cluster, or weight map.
2. Choose one of the simple deformations from the **Deform** menu on the Model, Animate, or Simulate toolbar: Bend, Bulge, Shear, Taper, Twist, Push, Randomize, or Shape Jitter.
3. Adjust and animate the parameters as desired. Refer to Online Help by clicking on the ? for information about every parameter on a property page.

Types of Simple Deformation

This section provides a quick visual reference of the effects you can achieve with the different types of simple deformation. For a complete description of every parameter available for a given deformation, refer to Online Help by clicking on the ? in the property editor.

Bend

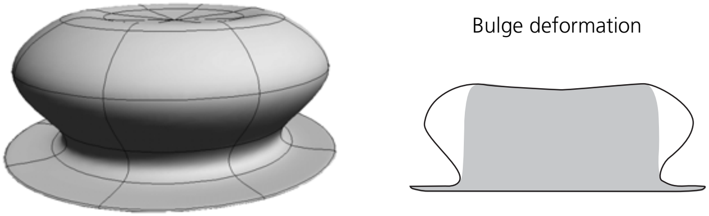
Bend folds an object. You can specify the axis that gets bent, the angle of the bend, the radius over which the bend occurs, the position where the bend starts, and the direction of the bend.



Bend deformation

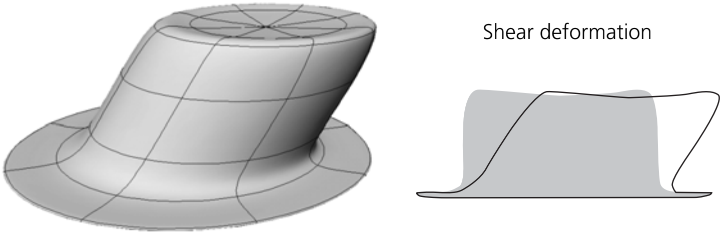
Bulge

Bulge pushes an object's points out from the center. You can specify which axes the deformation occurs on, as well as the reference axis, the amplitude, and the amplitude's profile.



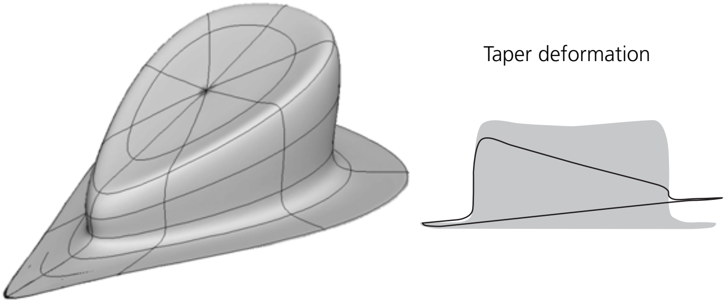
Shear

Shear pulls the ends of an object in opposite directions. You can specify which axes the deformation occurs on, as well as the reference axis, the amplitude, and the amplitude's profile.



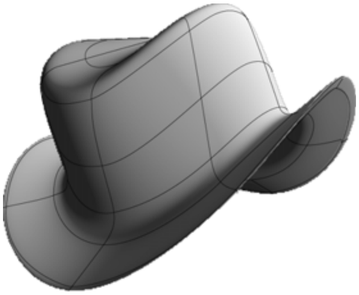
Taper

Taper gradually scales an object in one direction. You can specify which axes the deformation occurs on, as well as the reference axis, the amplitude, and the amplitude's profile.

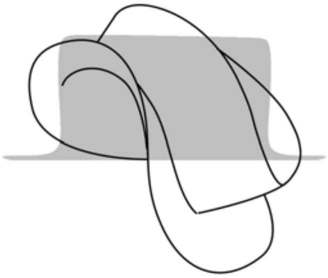


Twist and Vortex

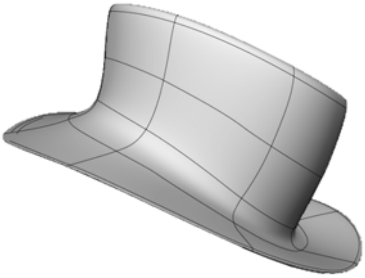
Twist progressively rotates an object in one direction. You can specify the axis or rotation, the maximum angle, and the angle’s modulating profile.



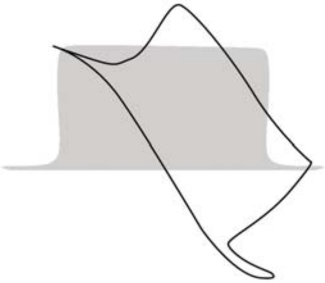
Twist deformation



You can also use the Twist deformation to create a vortex effect, where the amplitude of the rotation depends on the radial distance from the axis. The maximum angle occurs at the center of the deformation and tapers off towards the edges of the object.

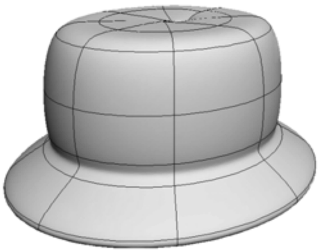


Vortex deformation

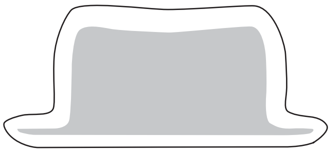


Push

Push moves points in the direction of their normals. You can specify the amplitude.



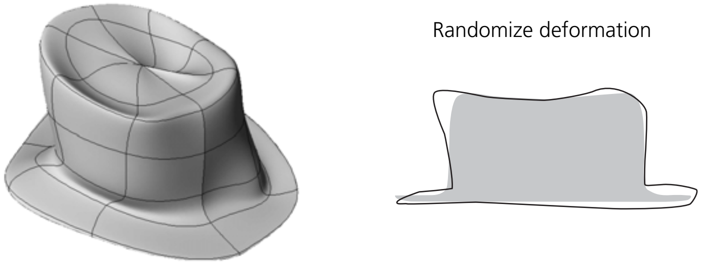
Push deformation



You can quickly create a weight map, connect it to a Push deformation, and activate the Paint tool as described in *Push Paint* on page 119. This lets you sculpt an object by painting.

Randomize

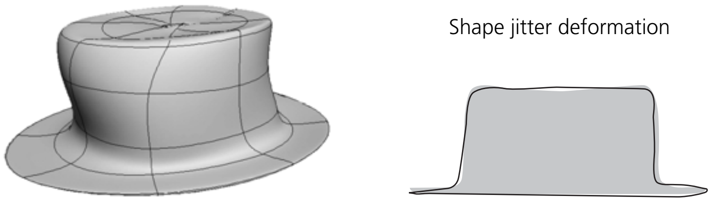
Randomize moves points around randomly. You can specify the maximal displacement in each axis, the number of repetitions, and the way that random values are generated.



Randomize deformation

Shape Jitter

Shape Jitter is like an animated Randomize.



Shape jitter deformation

The options on the Time Control tab of the Shape Jitter Op property editor control the time span of the effect:

- **Start offset** is the first frame for which the jitter begins to affect the object.
- **End** is the last frame for which the jitter affects the object.

Shape Jitter pre-allocates streams of random values for the point positions for each frame. This affects memory requirements and performance.

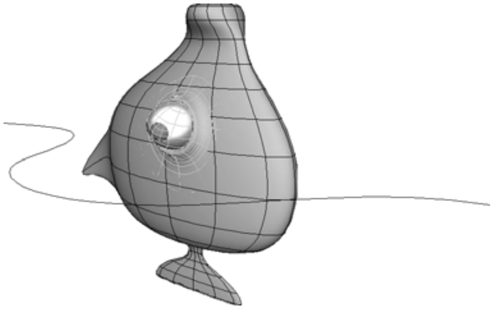
For best results, do not set the End frame of your scene to a higher value than necessary.

Curve and Surface Deformations

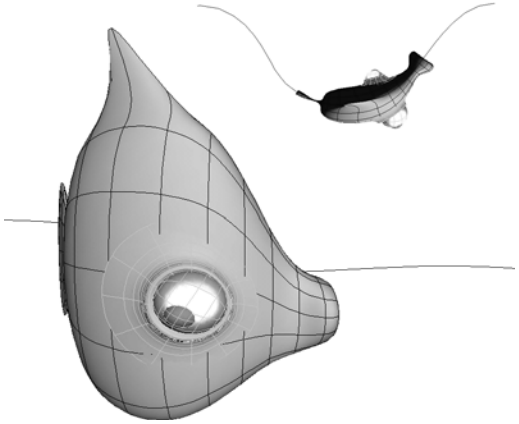
Curve and surface deformations use the U and V parameterizations of NURBS objects to twist the local space of a deformed object.

Deforming by Curves

Deformation by curve distorts an object by remapping the Y axis to any curve you pick. You can animate the object in the deformed space defined by the curve and you can also animate the shape of the curve itself.



Object and curve before the deformation is applied



Object and curve after the deformation is applied

To deform by curve

1. Create a curve using any of the available tools. Note that if the curve contains extremely sharp bends, the object may become severely distorted. For information about curves in general, see *Chapter 11: Curves* on page 359.
2. Select the object, branch, group, or model you want to deform.
3. Choose **Deform** > **by Curve** from the Model, Animate, or Simulate toolbar.
4. Pick the curve. The center of the deformed object snaps to the beginning of the curve and the Curve Deform property editor opens.
5. Select the **Axis** to deform along the curve.
6. Use the **Translation**, **Scaling**, and **Roll** parameters to move the object in deformed space. You can still use the standard SRT commands to move the object in ordinary space.

7. If you moved either the curve or the object from the global scene center before applying the curve deformation, you can compensate with the options on the Constraint page:

   - If you moved the object before deforming, make sure that **Constrain to Deformer** is on.
   - If you moved the curve before deforming, make sure that **Constrain to Deformee** is on.
   - If you moved both, make sure that both options are on.

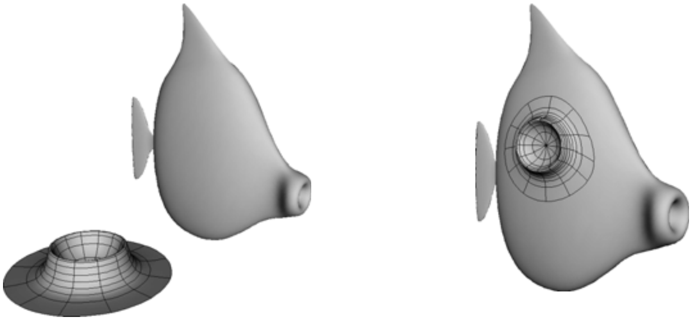
If you deformed an object by a curve, you should freeze its operator stack before using it as an envelope.

If you import an object deformed by curve in branch mode from SOFTIMAGE|3D, it might not appear correctly in SOFTIMAGE|XSI. This is because of basic differences in how deformation by curve is implemented.

However, it should still be possible to recreate the original effect by modifying the Curve Deform parameters in SOFTIMAGE|XSI.

Deforming by Surfaces

Deformation by surface distorts an object by remapping the XZ plane to the UV space of a surface you pick. You can animate the object in the deformed space defined by the surface and you can also animate the deforming surface.



Object and surface before the deformation is applied

Object deformed by the surface

You cannot use an assembled surface mesh with multiple subsurfaces as a deformer.

To deform by surface

1. Create a surface of any shape using any of the surface or deformation tools.
2. Select the object, branch, group, or model you want to deform.
3. Choose **Deform** > **by Surface** from the Model, Animate, or Simulate toolbar.
4. Pick the surface. The object deforms and the Surface Deform property editor opens.
5. Use the Translation, Scaling, and Roll parameters to move the object in deformed space. In particular, if the object is severely distorted, try using the Scale parameters to shrink it.
6. You can still use the standard SRT commands to move the object in ordinary space.
7. If you moved either the surface or the object from the global scene center before applying the curve deformation, you can compensate with the options on the Constraint page:

   - If you moved the object before deforming, make sure that **Constrain to Deformer** is on.
   - If you moved the surface before deforming, make sure that **Constrain to Deformee** is on.
   - If you moved both, make sure that both options are on.

If you deformed an object by a surface, you should freeze its operator stack before using it as an envelope.

If you import an object deformed by surface in branch mode from SOFTIMAGE|3D, it might not appear correctly in SOFTIMAGE|XSI. This is because of basic differences in how deformation by surface is implemented.

However, it should still be possible to recreate the original effect by modifying the Surface Deform parameters in SOFTIMAGE|XSI.

Curve and Surface Deformation Tips

Here are some tips and techniques related to deformations by curve or surface:

- If you deform a clone or instance and then move points on the original object, the points of the instance move in the deformed space.
- If you deform the parent of an object in branch, you can transform it in Parent mode instead of using the sliders in the property editor.
- To keep the length of a curve approximately constant as you manipulate it, deform it by another curve and manipulate the second curve instead.

Lattices

Lattices make it easy to deform a large amount of geometry at once. They allow you to deform objects by warping the 3D space around them. A lattice is a control box with a variable resolution, looking rather like scaffolding surrounding an object.

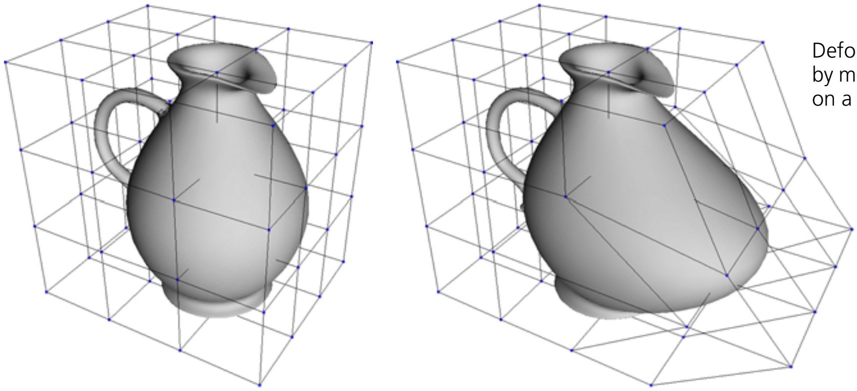
When you move a point on a lattice, its original location in space is mapped to its new location. This warps the space between points and objects that are affected by the lattice become distorted.

Lattice deformations have two parts:

- A Lattice object.
- A Lattice deformation operator on each object deformed by the lattice.

You can set the properties for these parts independently so that several objects can be deformed in different ways by the same lattice object.

- Unlike in SOFTIMAGE|3D, lattices are not a special type of geometry. You can select and transform points, move points, create clusters, assign cluster centers, apply deformations, use shape animation, and so on, just as with other 3D objects.
- Lattices do not need to be the parent of the deformed objects. However, to obtain similar results to SOFTIMAGE|3D when transforming lattices, make the deformed objects the children of the lattice.
- Scenes with lattice animation might not be imported correctly. There may be differences because transitions in XSI are between two shape clips, while in SOFTIMAGE|3D transitions are between all shape keys. You can fix this problem in the animation mixer by mixing weight curves instead of using transitions.



Deform objects by moving points on a lattice

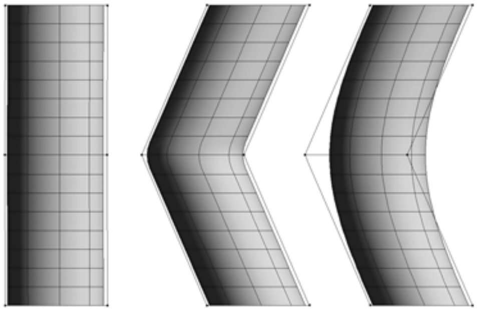
Creating and Applying Lattices

You can create a lattice and apply it to an object at the same time.

To create and apply a lattice

1. Select the object, branch, group, model, points, or clusters you want to deform.
2. Choose **Get** > **Primitive** > **Lattice**. A lattice is created to fit the object and is automatically parented to the deformed object. The lattice's property editor opens.
3. You can set the lattice's subdivisions in each axis; more subdivisions give greater resolution for the deformation.

   You can also set the interpolation type along each axis. **Curve** yields smoother deformations than **Linear**.



Undeformed

Linear interpolation

Curve interpolation

4. Deform the lattice in any way. For example, you can select and move points, use clusters, or apply any other deformation to it. You can also animate the lattice's deformation.

   As the lattice deforms, the object deforms with it.

Applying an Existing Lattice

If a lattice object already exists in the scene, you can use it to apply a deformation.

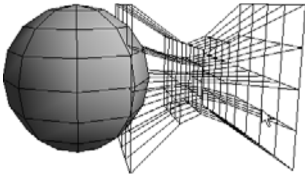
To apply an existing lattice

1. If necessary, create a new lattice by first deselecting all objects then choosing **Get** > **Primitive** > **Lattice**. A lattice is created and the Lattice property editor opens. You can set the lattice's subdivisions and interpolation type.
2. Select the object, branch, model, points, or clusters you want to deform.
3. Choose **Deform** > **by Lattice** from the Model, Animate, or Simulate toolbar.
4. Pick the lattice. The Lattice deformation property editor opens.
5. Deform the lattice in any way.

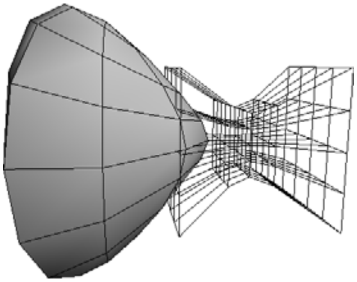
Setting the Scope of Lattice Deformations

By default, all points on an object are deformed by a lattice no matter where they are in space. You can set the **Deformation Scope** in the Lattice deformation property editor so that only those points that are within the lattice object itself are affected:

- **All Points** deforms the entire object no matter where it is. The deformation space is extrapolated beyond the lattice object's volume.
- **Falloff Relative to Lattice's BBox** deforms only those points within the influence of the lattice's bounding box. This is similar to local lattices in SOFTIMAGE|3D. With this option, you can choose which bounding box to use:
  - **Predeformed BBox** uses the lattice's original size as set on the lattice object's property editor.
  - **Deformed BBox** uses the lattice's current bounding box after any deformations have been applied.
- You can also define the **Start** and **End** of a falloff of the lattice's effect relative to the bounding box; this prevents points from jumping as soon as they enter the lattice's influence.

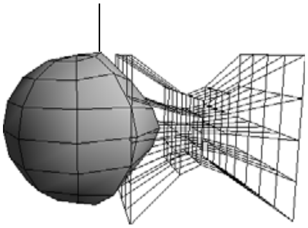


Undeformed sphere



Deformation Scope = All Points

Falloff extends beyond lattice



Deformation Scope =
Falloff Relative to Lattice's BBox

Scaling

You can control how an object is affected when the lattice object is scaled by setting the **Scaling Mode** option in the Lattice deformation property editor:

- **No Scaling**—The deformed object is not scaled when the lattice object is scaled. However, the deformed object is still affected as the lattice's points move when scaled.
- **Treat Scaling as Deformation**—The deformed object is scaled before the lattice deformation is applied.
- **Apply Scaling to Geometry (SI3D)**—The lattice deformation is applied first, then the deformed object is scaled. This is the behavior in SOFTIMAGE|3D.

Cage Deformations

Cage deformations allow you to use a low-resolution geometry to drive the deformation of a high-resolution object. They are similar to arbitrarily shaped lattices.

Each point on the deformed object is weighted to the points or polygons on the cage, similarly to how an envelope's points are weighted to its deformers. However, you cannot paint the weight assignments nor adjust them manually. You can also have multiple cage deformations on the same object, unlike envelopes.

Cage Deformations and Construction Modes

Cage deformations use a "rest" shape of the cage object to determine how to modify the deformed object, as well as how points on the deformed object are weighted to the points or polygons of the cage. Any deformations you apply to the cage on top of this rest shape affect the deformed object as well.

When you apply a cage deformation, the rest shape is defined by the current top of the modeling construction region in the cage object's geometry operator stack. The cage-deformed object is affected by any new deformations above this point in the modeling region, as well as in the other regions. If the cage object already has deformations above the modeling region, the shape of the cage-deformed object changes as soon as you apply the cage deformation.

The rest shape can be affected by modifications you make to the cage object. If you freeze all or part of the cage object's modeling region and "bake" the position in the stack where the deformation reads the rest shape, then the bottom of the stack becomes the rest shape.

When you set the **Construction Mode Display** of 3D views to **Sync with current mode**, you may or may not see the effect of the cage deformation depending on which region of the cage-deformed object contains the Cage Deform Operator. However, if the effect is visible, then the deformed object is always affected by all operators on the cage object no matter what region they are in.

For more information about construction modes in general, see *Construction Modes and Regions* on page 53.

If you apply a topology operator to the cage object, the cage deformation no longer works. It is not possible to calculate the deformation if the final topology is different from the rest shape.

Applying Cage Deformations

To apply a cage deformation

1. Select the objects, branches, groups, models, clusters, or points to be deformed.
2. Choose **Modify** > **Deform** > **by Cage** from the Model toolbar. This command is also available on the Animate and Render toolbars.
3. Pick one or more objects, branches, or groups to act as cage deformers.
4. When you have finished picking deformers, right-click in a 3D view to finish. The Automatic Cage Assignment property editor opens.
5. Set the parameters as desired. These parameters control how the points on the deformed object are weighted to points or polygons on the cage.

When you pick an object as a cage deformer, its Render Visibility is turned off in its Visibility property and its display type is set to Wireframe in its Display properties.

To see the effect of the display settings in non-wireframe views, make sure that **Override Object Properties** is off.

Cage Deformation Tips

Cage deformations can involve a large amount of data. Here are some tips for making them as fast as possible:

- Make sure that your cage has the fewest points or polygons necessary for the effect you want to achieve.
- When modeling the cage, keep it as close as possible to the geometry you want to deform.
- Assign cage objects locally to clusters if possible.
- When you are satisfied with the cage assignment, freeze the cage weights (that is, the Envelope_Weights operator under the CageWeightCls cluster).

Spine Deformations

Deformation by spine lets you change an object's shape using curves as deformers, similar to the way that you can deform envelopes by moving bones in a chain. Each curve defines a cylinder of influence with an associated radius and object points within a curve's influence are assigned to that curve. If an object point is close to two or more curves, it is weighted between them.

Spine Deformations and Construction Modes

Spine deformations use a "rest" shape of the spine curve to determine how to modify the deformed object. Any deformations you apply to the spine on top of this rest shape affect the deformed object as well.

When you apply a spine deformation, the rest shape is defined by the current top of the modeling construction region in the spine curve's geometry operator stack. The spine-deformed object is affected by any new deformations above this point in the modeling region, as well as in the other regions. If the spine curve already has deformations above the modeling region, the shape of the spine-deformed object changes as soon as you apply the spine deformation.

The rest shape can be affected by modifications you make to the spine curve. If you freeze all or part of the spine curve's modeling region and "bake" the position in the stack where the deformation reads the rest shape, then the bottom of the stack becomes the rest shape.

When you set the **Construction Mode Display** of 3D views to **Sync with current mode**, you may or may not see the effect of the spine deformation depending on which region of the spine-deformed object contains the Deform by Spine Operator. However, if the effect is visible, then the deformed object is always affected by all operators on the spine curve no matter what region they are in.

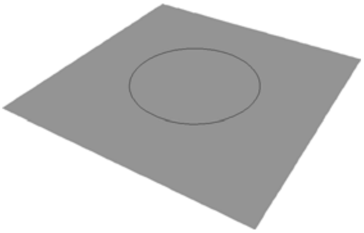
For more information about construction modes in general, see *Construction Modes and Regions* on page 53.

Applying Spine Deformations

To deform by spine

1. Create the curves for deforming the object. The curves should be as close to the object's surface as possible; otherwise, points on the object's surface may fall outside the curve's radius of influence.

   Also, note that the closer the curves' points are to the object's points, the more closely the deformation of the object's surface will follow the curves' points as you move them.



Surface and curve

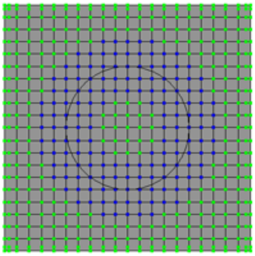
You can use the Shrinkwrap deformation to project curves onto the object's surface. Don't forget to freeze the shrinkwrapped curves before using them with Deform by Spine. For more information about Shrinkwrap in general, see *Shrinkwrap* on page 160.

2. Select the object you want to deform.

3. Choose **Deform** > **By Spine**. This command is available under Modify on the Model toolbar as well as under Deform on the Animate toolbar.

4. Pick the curves you want to act as deformers.

5. When you have finished picking curves, right-click in a geometry view. Clusters are created and colored according to the deformer and the Deform by Spine property editor appears.



Clusters on the surface

6. Adjust the values as desired:

   - **Amplitude Falloff Profile** controls the modulation of the deformation along the radius of influence from the curve to the edge of the effect.

   - **Radius** controls the base width of the cylinder of influence in SOFTIMAGE units.

   - **Longitudinal Radius Profile** controls the actual width of the cylinder of influence as a fraction of the base Radius along the percentage length of the curve.

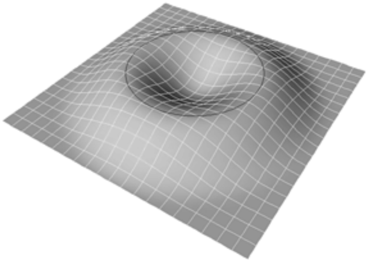
7. Select a curve deformer and use it to change the shape of the object. Any change you make to a curve deformer is reflected in the shape of the object. For example, you can:

   - Translate, rotate, or scale the curve.

- Move points, either using the Move Point tool or by selecting points and transforming them with the SRT tools.

  You can also add points to the curve for finer control. For more information, see *Modifying Curves* on page 377.

- Apply deformations like Twist and Bend to the curve.



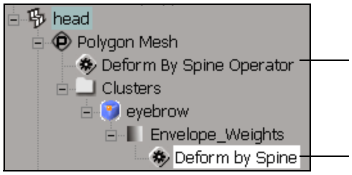
Move the curve to pull the points.

Modifying Spine Weights

Spine deformations use the same mechanism as envelopes for weighting points to deformers. You can modify the weights of points in a spine deformation either manually or by painting.

Each point on an envelope has a total weight of 100%, which is divided between the deformers to which it is assigned. For example, if a point is weighted by 75% to deformer A and 25% to deformer B, then A pulls three times as strongly as B on the point.

The Deform by Spine node is available under the Envelope_Weights property of the corresponding cluster. It allows you to modify the initial point weighting to the spine deformers.



Mute or unmute the spine deformation.

Adjust the initial weighting to deformers.

You can modify the way points are weighted to the curves in exactly the same way as for envelopes: by painting or by editing numerically.

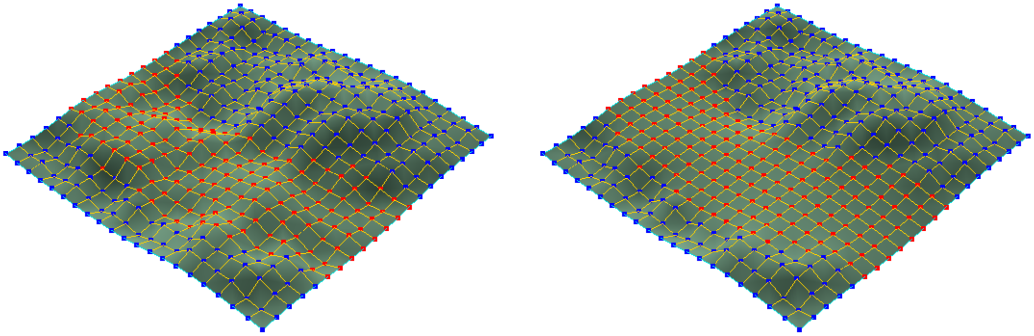
When you modify the weighting, your changes are added to the Envelope_Weights operator stack, similar to the geometry operator stack. Once you are satisfied with the deformation, you can freeze the weights to improve performance. After you have frozen the weights, you can still make modifications, but XSI cannot recalculate the initial assignment.

For a complete description, see *Modifying Envelope Weights* in Chapter 3 of the *Character Animation* guide.

Smoothing and Relaxing

The Smooth operator is a general purpose deformation that removes spikiness and other high-frequency detail. This is very useful in many situations, for example:

- Smoothing out discontinuities in the general shape of an organic object after you have tweaked it by manually moving points.
- Rounding out seams where polygons were extruded, or between parts that were modeled separately and then merged into a single object.
- Smoothing out the general shape of an object without increasing the number of points by subdividing.
- Temporarily flattening out overhanging polygons to get a good texture subprojection in tricky areas like noses and ears.
- Straightening edge loops or aligning quads so that objects deform and shade better.
- Flattening terrain around buildings and roads.



There are two commands that apply different presets of the Smooth operator: **Smooth** affects the entire shape while **Relax** aligns components along the surface. In either case, you can adjust the effect further by modifying parameters as described in the sections that follow.

Smooth works by averaging the positions of points. As a result, quads tend to become squarer, triangles tend to become more equilateral, and edge loops tend to become straighter. However, you can define a custom filter with different characteristics as described in *Smoothing with Custom Filters* on page 157.

Applying Smooth and Relax

To apply smooth and relax

1. Select the elements to be smoothed: points, edges, polygons, clusters, objects, branches, groups, or models.

   Smoothing works with any geometry type: polygon meshes, NURBS surfaces, NURBS curves, lattices, hair, and particles. However, note that:

   - With particles, you must define a custom filter with a **Distance type** of **Spatial (ignore neighborhood)**. For more information, see *Smoothing with Custom Filters* on page 157.

   - With NURBS surfaces, closed boundaries are treated as if they were connected. However, boundaries between subsurfaces in assembled surface meshes are not considered connected.

2. Choose one of the following commands from the Model, Animate, or Simulate toolbars:

   - **Modify** > **Deform** > **Smooth**. This preset is a good starting point if you want to smooth the overall shape.

     or

   - **Modify** > **Deform** > **Relax**. This preset is a good starting point if you want to align components without affecting the silhouette as much.

   The Smooth property editor opens.

3. Adjust the parameters as described in the sections that follow.

Denser areas smooth more slowly than lighter areas. In some situations, this can create protrusions when you smooth an object. If your object has a highly variable density, try using a custom filter with a **Distance type** of **Edge Path Length**. For more information, see *Smoothing with Custom Filters* on page 157.

Controlling the Amount of Smoothing

You can set the strength of the smoothing effect directly, as well as extrapolate the displacement of points, as described next. In addition, you can control the strength near unsmoothed areas using a falloff (see *Creating a Falloff Between Smoothed and Unsmoothed Areas* on page 155) and set a minimal displacement (see *Setting a Minimal Displacement* on page 157).

Controlling the Strength Directly

Strength on the General tab of the Smooth property editor controls the amount of smoothing. It represents the number of iterations that each point's position is averaged with its neighbors' positions. Decimal values interpolate between whole iterations.

You can also use a weight map by right-clicking on the connection icon. This lets you paint weights to differentiate between areas where you want more or less smoothing to occur.

For more information about using weight maps to control values in general, see *Chapter 11: Weight Maps* in the *Fundamentals* guide.

Extrapolating the Displacement

Extrapolation on the Advanced Settings tab of the Smooth property editor exaggerates the displacement of each point. Positive values move the points in the same direction as the smoothing, and negative values move points in the opposite direction.

While you can achieve some interesting effects with this parameter, it is particularly valuable as a diagnostic tool: you can temporarily adjust it to see what areas are being most affected by smoothing, and then set it back to 0.

Preserving Features While Smoothing

You can protect vertices that meet specific criteria from being smoothed. You can also control whether unselected vertices are considered when calculating average positions, as well as create a falloff between unsmoothed and smoothed areas.

Protecting Vertices that Meet Specific Criteria

The options under **Feature Preservation** on the General tab of the Smooth property editor protect vertices that meet the corresponding criteria:

- **Boundary vertices** prevents points on boundary edges from being smoothed. Because boundary edges have neighbors on only one side, they are often pulled strongly in that direction when positions are averaged. This option prevents things like mouths and eyes from becoming distorted by smoothing. It can also help when smoothing assembled surface meshes, because the boundaries of the subsurfaces are not considered connected.

- **Hard edges/vertices** prevents points and edges that have been marked as hard (using **Modify** > **Component** > **Mark Hard Edge/Vertex**) from being smoothed.

- **Crease edges/vertices** prevents points and edges with crease values set (using **Modify** > **Component** > **Set Edge/Vertex Crease Value**) from being smoothed.

When any of these options are on, you can create a falloff from the protected vertices as described in *Creating a Falloff Between Smoothed and Unsmoothed Areas* below.

Controlling Whether Unselected Vertices Are Considered

When you smooth a selected portion of an object, **Consider unselected vertices** on the General tab of the Smooth property editor controls whether the vertices in the unselected (unsmoothed) area are considered when average positions are calculated.

- When this option is on, all of a point's neighbors are considered when the average position is calculated. This allows for a transition at the borders of the smoothed region. You can refine this transition further by creating a falloff as described in the next section, *Creating a Falloff Between Smoothed and Unsmoothed Areas*.
- When this option is off, only other selected vertices are considered when calculating average positions. As a result, the selected area is smoothed as if it was disconnected from the rest of the geometry.

Creating a Falloff Between Smoothed and Unsmoothed Areas

The options under **Falloff** on the General tab of the Smooth property editor allow you to create a gradual transition between the smoothed and unsmoothed areas.

- **Radius** represents how far the falloff area extends into the smoothed region from its borders.

  By default, values are in terms of the neighborhood, where 1 represents points that are immediately adjacent to the border of the unsmoothed area, 2 represents points that are one point away from the border, and so on. Values between whole numbers are interpolated.

  If you are using a custom filter, values are in terms of the distance type you set.
- **Sharpness** controls the shape of the transition in the falloff region. Low values produce an abrupt transition at the beginning of the falloff region (near the unsmoothed areas) and high values produce an abrupt transition near the end of the falloff region. Moderate values produce a smooth transition spread out over the whole falloff region.

Adjust both sliders to achieve the desired falloff profile. If either slider is 0, there is no falloff at all.

The falloff options are unavailable unless at least one of the options above them (**Boundary vertices**, **Hard edges/vertices**, **Crease edges/vertices**, or **Consider unselected points**) is active.

Aligning Components, or Preserving the Shape While Smoothing

Because smoothing works by averaging the positions of vertices, objects tend to shrink. There are a couple of ways to minimize the shrinkage. Both of these methods tend to align components along the surface rather than smooth sharp details.

- **Reproject** on the Advanced Settings tab of the Smooth property editor projects each point back onto the surface of the unsmoothed geometry. As a result, the components become aligned in surface space but the object does not shrink as much. However, the reprojection may miss some spiky details, so the object's shape may still be smoothed down somewhat even when this option is on.

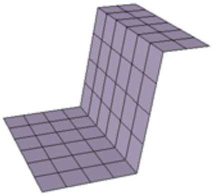
- Setting **Restrict to normal direction** to **Perpendicular** on the Advanced Settings tab of the Smooth property editor prevents each point from being displaced along its normal (local Y) and restricts the movement to its local X and Z. This option is better than **Reproject** at keeping sharp details, but is less good at aligning components in surface space.

Restricting the Smoothing Effect

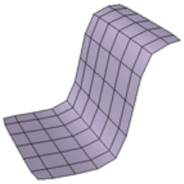
Several options on the Advanced Settings tab of the Smooth property editor allow you to restrict the smoothing effect.

Restricting the Effect by Curvature

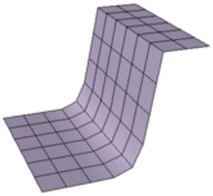
Restrict to curvature type on the Advanced Settings tab of the Smooth property editor lets you restrict the smoothing effect based on the local curvature: concave or convex, as determined by the geometric normals (user-defined normals are not considered).



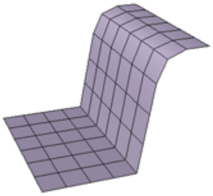
Original unsmoothed geometry.



Restrict to curvature type = None smooths the entire geometry.

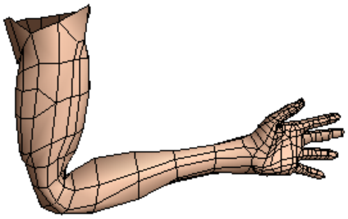


Restrict to curvature type = Concave smooths concavities.

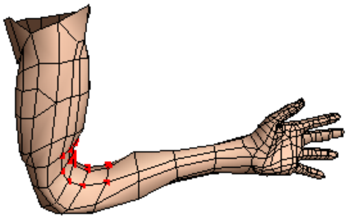


Restrict to curvature type = Convex smooths only convexities.

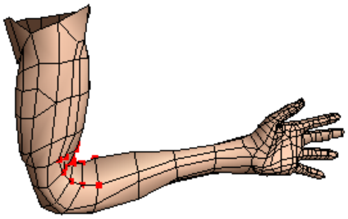
For example, you can smooth concavities to eliminate pinching that may occur in elbows and other regions:



Unsmoothed arm shows pinching in the elbow.



Smoothing selected vertices in the elbow area with no restrictions on curvature produces an undesirable effect.



Restricting the smoothing effect to **Concave** produces a much better result.

Restricting the Direction of the Effect

Restrict to normal direction on the Advanced Settings tab of the Smooth property editor controls the direction in which points are displaced:

- **None** allows points to be displaced in any direction without restriction.
- **Parallel** allows points to move only in the direction of their normals (local Y axis). This can be useful, for example, if you want to smooth the height of terrain without affecting the alignment of the map grid or roads.
- **Perpendicular** allows points to move only in the direction of their local X and Z axes. This can be useful for aligning components along the surface without smoothing protuberances and indentations. See *Aligning Components, or Preserving the Shape While Smoothing* on page 155.

Setting a Minimal Displacement

Minimum displacement on the Advanced Settings tab of the Smooth property editor is a clamp that sets the minimum distance that a point will be moved. If the difference between a point's current position and the average position of its neighbors is less than this amount, the point is not moved. This allows you to smooth noisy areas of an object without affecting areas that are already smooth enough. Values are in Softimage units.

Smoothing with Custom Filters

The Custom Filter tab of the Smooth property editor allows you to specify a different algorithm than the default smoothing. With a custom filter, each point's position is still being averaged with the positions of nearby points. However, you can control:

- The criterion, or distance type, used to measure the distance between points.
- The maximum distance with respect to that criterion for a point to be considered "nearby".
- A profile curve that weights the average over the distance used.

All the other options on the General and Advanced Settings tabs of the Smooth property editor work in conjunction with custom filters.

To use a custom filter for smoothing

1. On the Custom Filter tab of the Smooth property editor, activate **Use custom filter**.
2. Set the other options as described in the sections that follow.

Setting the Distance Type, Distance, and Computation Limits

Distance type on the Custom Filter tab of the Smooth property editor determines the criterion used to measure the distance between points, and **Distance** determines how far away "nearby" points can be.

No matter which distance type you choose, large distance values can cause long computation times because the operator evaluates many nearby points. The **Computation limit** options are a "safety valve" that prevent excessively long calculations that may occur if you set values that result in enormous numbers of points being considered "nearby" to each other. Setting values too high may result in long computation times when Distance is high, and setting values too low may prevent the Smooth operator from evaluating all nearby points.

- A **Distance type** of **Neighborhood** uses connectivity as a measure of distance. This is the same distance type used by the default smoothing algorithm, but the custom settings give you greater control over the distance and profile.

  The **Distance** between two points is the least number of edges in any path between them. Thus, adjacent points have a distance of 1, points that are separated by two edges have a distance of 2, and so on. Even though points are always separated by a whole number of edges, you can still use decimal values for the distance; values between whole numbers are interpolated, so a distance of 1.5 yields a displacement halfway between the displacements achieved by distances of 1 and 2.

  A characteristic of this distance type is that edge lengths tend to become equal at high **Strength** values.

- A **Distance type** of **Edge Path Length** uses the sum of the edge lengths in the shortest path between two points as the measure of distance. **Distance** values are in Softimage units.

  A useful characteristic of this distance type is that the ratio of edge lengths is maintained, even at high **Strength** values.

  Use **Max neighborhood depth** to prevent excessive calculations.

- A **Distance type** of **Spatial** (**consider neighborhood**) uses the distance between points in 3D space, as long as the points are connected by edge paths. **Distance** values are in Softimage units.

  Use **Max neighborhood depth** to prevent excessive calculations.

- A **Distance type** of **Spatial** (**ignore neighborhood**) uses the distance between points in 3D space, whether or not the points are connected. The **Distance** values are in Softimage units.

  This is the only distance type that can be used with particles, since the points in a particle cloud are not connected. This distance type is also useful for hair because only points on the same guide hair are connected.

  Use **Max visited vertices** to prevent excessive calculations.

Setting the Smooth Profile

The **Smooth profile** is a graph that controls how the positions of nearby points are weighted as a function of distance. The horizontal axis represents the distance of a nearby point, where 100 is the maximum distance set by the **Distance** slider. The vertical axis represents how strongly points at the corresponding distance are weighted when calculating the average position. Positive values pull a point towards its neighbors, and negative values repel it.

You can select the profile curve and edit it using the mouse and the same keyboard commands as the fcurve editor, or right-click to display a menu.

For more information about editing curves in general, see *Chapter 5: Editing Function Curves* in the *Animation* guide.

Shrinkwrap

The shrinkwrap deformation projects a wrapper object onto the surface of a target object. You can completely engulf the target, thereby giving the wrapper the same overall shape, or you can apply the wrapper onto the target like a decal.

You are not restricted in the type of objects that you can shrinkwrap. You can shrinkwrap any combination of surfaces, surface meshes, polygon meshes, and curves onto other like objects.

Shrinkwrapping surface mesh objects with multiple subsurfaces may be slow to process.

Sample Uses of Shrinkwrap

Among many things, the shrinkwrap deformation can be used to:

- Create a metamorphosis between two dissimilar objects by shrinkwrapping an object like a sphere onto one of the two objects, saving a shape key, and then repeating for the other object.
- Project a curve onto a surface. For example, you can project a path onto uneven terrain or use a shrinkwrapped curve as a modeling aid.
- Shrinkwrap surfaces onto a high-resolution polygon-mesh object to capture detail.

Shrinkwrap and the Modeling Relation

The shrinkwrap deformation maintains a modeling relation. This means that if you transform or deform the target (or the inner object, depending on the type of projection) after you have applied the deformation, the shape of the wrapper is altered accordingly.

If you freeze an object or work in immediate mode, the operator stack is lost. For more details about the operator stack in general, see *Building 3D Objects* on page 48.

You must freeze the shrinkwrapped object before deleting the target or the inner object. Otherwise, the deformation no longer has any effect.

Applying Shrinkwrap

When shrinkwrapping, there are several choices for how each point of the wrapper is projected onto the target:

- Toward an inner object until they hit the surface of the target.
- Toward the center of the target.
- Along an axis.
- Following the normals of the predeformed wrapper.

If any points of the wrapper do not hit the surface of the target using the selected projection type, they are not affected by the shrinkwrap deformation.

Shrinkwrapping toward an Inner Object

To shrinkwrap toward an inner object, you need three objects:

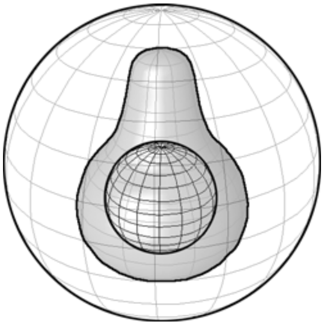
- **A wrapper**—the object that becomes deformed. Normally it should be outside the target object. If you want, the wrapper can completely surround the target.
- **A target**—the object around which the wrapper is shrunk.
- **An inner object**—the object toward which the wrapper is projected until it hits the target. It must have the same number of points as the wrapper—each point on the wrapper is projected toward the corresponding point on the inner object.

  The easiest way to create an inner object is to duplicate the wrapper, then scale the duplicate down and move it inside the target. If desired, you can deform the inner object so that there is more geometry near areas where you want high detail.

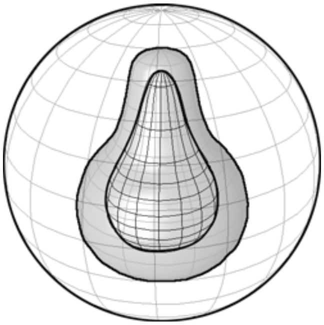
Once you have created and positioned your three objects, follow these steps.

To shrinkwrap toward an inner object

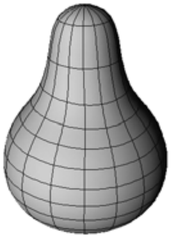
1. Select the wrapper object.
2. Choose **Deform** > **Shrinkwrap**. This command is available under **Modify** on the Model and Simulate toolbars, as well as under **Deform** on the Animate toolbar.
3. Pick the target object.
4. Pick the inner object. The Shrinkwrap property editor opens.
5. In the property editor, make sure that **Toward Inner Object** is selected and set the other parameters as desired—for information about other parameters, see *Other Shrinkwrap Controls* on page 165.



The wrapper, target, and inner object



Using a modified inner object to help the shrinkwrap



Final result

Shrinkwrapping toward the Target’s Center

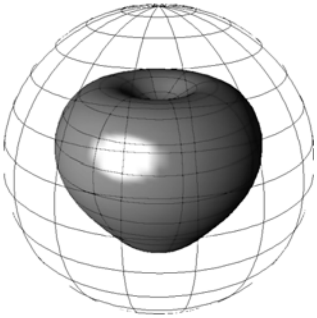
When you shrinkwrap toward the target’s center, each point of the wrapper object is projected toward the target’s center until it hits the target’s surface. In this case, you require two objects:

- **A wrapper**—the object that becomes deformed. Normally, it should be outside the target object. If you want, the wrapper can completely surround the target.
- **A target**—the object around which the wrapper is shrunk.

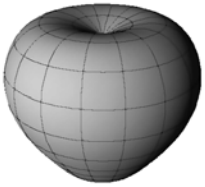
Once you have created and positioned both objects, follow these steps.

To shrinkwrap toward the target’s center

1. Select the wrapper object.
2. Choose **Deform** > **Shrinkwrap**. This command is available under **Modify** on the Model and Simulate toolbars, as well as under **Deform** on the Animate toolbar.
3. Pick the target object.
4. Right-click in a geometry view to end the picking session. (When shrinkwrapping toward the center, you do not need to pick an inner target.)
5. In the Shrinkwrap property editor that opens, make sure that **Toward Center** is selected and set the other parameters as desired:
   - If **Bounding Box Center** is off, the points of the wrapper are projected toward the center of the target’s local coordinate system. You can move the target’s center in Center mode and the wrapper’s shape is altered accordingly.
   - If **Bounding Box Center** is on, the points of the wrapper are projected toward the center of a bounding box containing the target. You can use Center mode to move the center of the target’s local coordinate system without affecting the wrapper’s deformation.



The wrapper (sphere) and target (apple)



Result of shrinkwrapping the sphere toward the apple’s center

For information about other parameters, see *Other Shrinkwrap Controls* on page 165.

Shrinkwrapping along an Axis

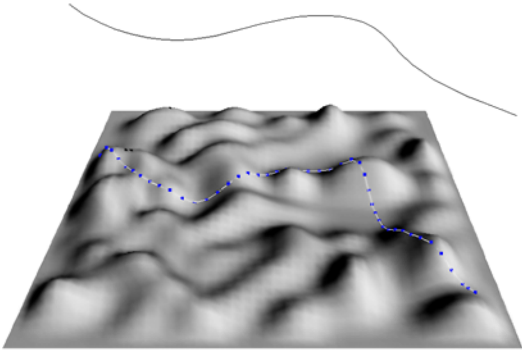
When you shrinkwrap along an axis, each point of the wrapper object is projected parallel to one of the axes of the target's local coordinate system until it hits the target's surface. As with shrinkwrapping toward the center, you require two objects:

- **A wrapper**—the object that becomes deformed.
- **A target**—the object around which the wrapper is shrunk.

Once you have created and positioned both objects, follow these steps.

To shrinkwrap along an axis

1. Select the wrapper object.
2. Choose **Deform** > **Shrinkwrap**. This command is available under **Modify** on the Model and Simulate toolbars, as well as under **Deform** on the Animate toolbar.
3. Pick the target object.
4. Right-click in a geometry view to end the picking session. (When shrinkwrapping along an axis, you do not need to pick an inner target.)
5. In the Shrinkwrap property editor that opens, select **Parallel to Axis**, then select one of the target's local axes (**Along X**, **Y**, or **Z**). Set the other parameters as desired—for information about other parameters, see *Other Shrinkwrap Controls* on page 165.



Shrinkwrapping a curve along the Y axis onto a surface

Shrinkwrapping Along the Normals

When you shrinkwrap along the normals, each point of the wrapper object is projected along the inverse of its normal direction until it hits the target's surface. This lets you, for example, shrinkwrap a belt onto a character without worrying about where the objects' centers are.

As with shrinkwrapping toward the center or along an axis, you require only two objects:

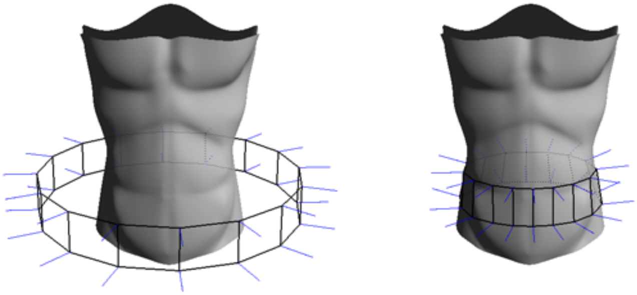
- **A wrapper**—the object that becomes deformed.
- **A target**—the object around which the wrapper is shrunk.

Once you have created and positioned both objects, follow these steps.

Shrinkwrapping does not use user-defined normals.

To shrinkwrap along an axis

1. Select the wrapper object.
2. Choose **Deform** > **Shrinkwrap**. This command is available under **Modify** on the Model and Simulate toolbars, as well as under **Deform** on the Animate toolbar.
3. Pick the target object.
4. Right-click in a geometry view to end the picking session. (When shrinkwrapping along the normals, you do not need to pick an inner target.)
5. In the Shrinkwrap property editor that opens, select **Follow Vertex Normals.** Set the other parameters as desired—for information about other parameters, see *Other Shrinkwrap Controls* on page 165.



Shrinkwrapping along the normal of each vertex

Other Shrinkwrap Controls

With any type of projection, there are two other parameters that control the shrinkwrap deformation: Reverse Projection and Amplitude.

Reverse Projection

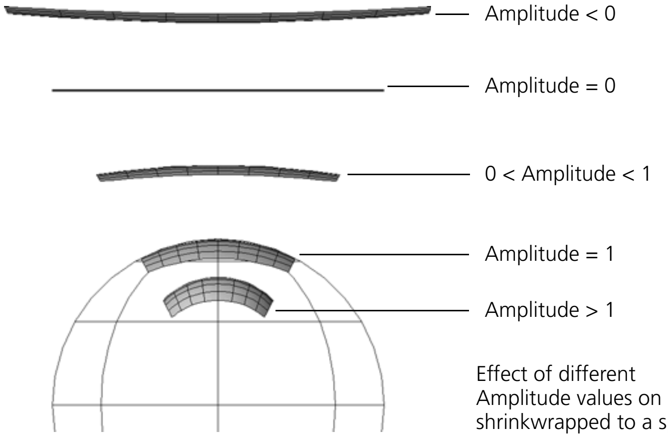
The Reverse Projection option in the Shrinkwrap property editor controls the direction in which the points of the wrapper object are projected. When on, points are moved in the opposite direction. For example, you can project the wrapper along an axis in the negative direction instead of the positive direction.

Amplitude

The Amplitude parameter in the Shrinkwrap property editor controls how far the points of the wrapper object are moved toward the target:

- When the Amplitude is 1, the points of the wrapper object are deformed to the surface of the target.
- Between 0 and 1, the points are deformed between their original positions and the target.
- Below 0, the points deform away from the target.
- Above 1, the points overshoot the surface of the target.

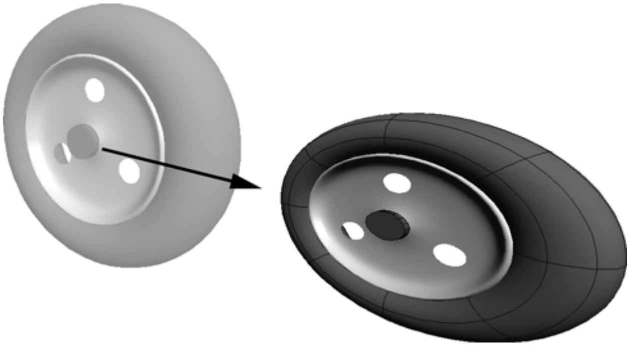
You can use a weight map to modulate the amplitude of the shrinkwrap effect across the surface of the wrapper object.



Effect of different Amplitude values on a grid shrinkwrapped to a sphere

Quickstretch

Quickstretch is an animated deformation that changes an object's shape automatically, based on its motion. Quickstretch calculates deformations on the fly, according to the object's speed and acceleration. You can make objects flex, stretch, and yield, based on their linear and rotational motion.



There are four components of motion used to calculate quickstretch deformation: linear velocity, linear acceleration, rotational velocity, and rotational acceleration. These motion components are described in more detail on page 169.

For each motion component, you can apply up to three quickstretch deformations: flexing, stretching, and yielding. The different effects of these deformation types is described on page 169.

Once you have applied quickstretch, you can see the effect by playing back the animation or by moving the object around in a geometry view.

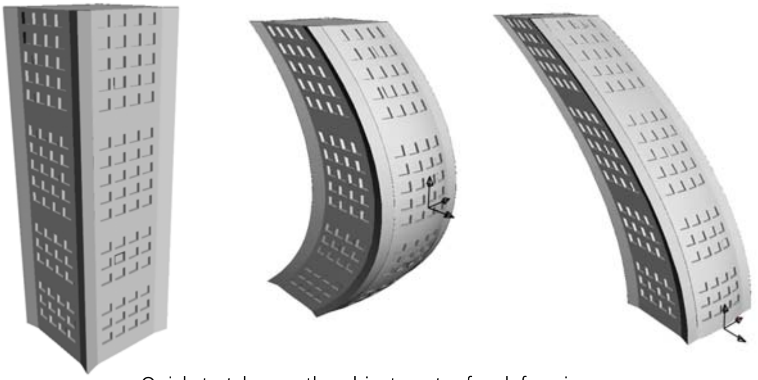
Before You Apply Quickstretch

Before you apply quickstretch, there are a couple of things you should check: object centers and subdivisions.

Object Centers

Quickstretch uses the object center as the center of the deformation. For hierarchies selected in branch mode, it uses the parent's center.

Objects with quickstretch appear to be deformed by some force—whether active or inertial—and the location where this force seems to be applied is the center of the deformation.



Quickstretch uses the object center for deforming.

Before you apply quickstretch, make sure that the object's center is located where you want the deformation to emanate from. For example, if a building faces such a strong wind that it bends a little, the deformation should originate from the ground up because you would expect the building to be attached to the ground.

If you move the center after you apply quickstretch, there is no change in the quickstretch deformation. This is because the Center operator comes after the quickstretch operator in the operator stack. You can reorder the operators in the stack (as described in *Changing the Order of Operators* on page 60) or disable the Quickstretch operator before moving the center (as described in *Disabling the Top of the Stack* on page 62).

Object Subdivisions

Quickstretch looks best with many subdivisions in the deforming object, but that can result in a great deal of computation and a heavy scene.

You can use the operator stack to change an object's subdivisions at any time, using fewer subdivisions to help you work quickly and then adding more subdivisions for the final result.

Applying Quickstretch

To create a quickstretch deformation

1. Select an object, cluster, or weight map.
2. Choose **Deform** > **QStretch**. This command is available under **Modify** on the Model and Simulate toolbars, as well as under **Deform** on the Animate toolbar. The QStretch Op property editor appears.
3. Set the parameters as desired. On the Overview property page, you can toggle the three deformation types (**Flex**, **Stretch**, and **Yield**) for each motion component (**Linear Velocity/Acceleration** and **Rotational Velocity/Acceleration**).

   These toggles are also available on the other property pages, which also provide finer control over the deformations associated with each motion component.

   - Motion components are described on page 169.
   - Quickstretch deformation types are described on page 169.

Viewing a Quickstretch Deformation

Once you have created a quickstretch deformation, you can view the effect in either of the following ways:

- By playing back the animation.
- By using the Translate tool to move the object around in a geometry view.

Since motion vectors are computed on the fly during playback of an animated sequence, the deformation that occurs is different if you play the animation backward!

Similarly, if you jump from one frame to another, a huge deformation may occur if the difference in position implies a very large velocity or acceleration. In this case, the geometry returns to normal at the next refresh.

Motion Components

Quickstretch uses the following aspects of an object’s motion to deform the object.

- Linear velocity
- Linear acceleration
- Rotational velocity
- Rotational acceleration

A moving object has speed or *velocity*. If it is changing speed, then it also has *acceleration* or *deceleration*. For example, a car moving at a steady 100 km/h has a velocity of 100 but an acceleration (and deceleration) of 0.

Velocity and acceleration can each be further divided into two types: *linear* and *rotational*. For example, a car that moves along a straight line has only a linear motion, whereas a ball that spins on the spot has only rotational motion. Regardless of the type of motion, the faster the object is moving, the more it is deformed.

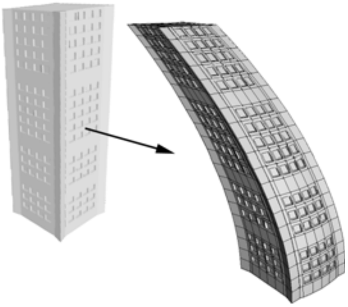
Quickstretch Deformation Types

Whichever type of motion you give to an object, it can be deformed in one or more ways: it can flex, stretch, yield, or do any combination of the three.

Each of these deformation types can be weighted independently of the others, but their effects are additive. Each effect is designed to be tweaked separately as much as possible, so that when you use them all together the resulting deformation is more predictable.

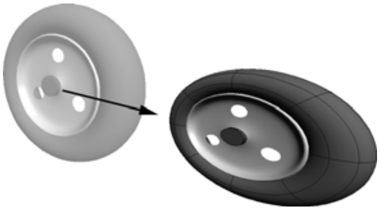
Flexing

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).



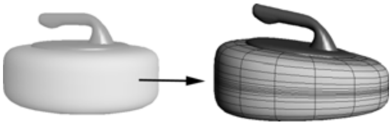
Stretching

In the case of a cartoon “squash and stretch” effect, the object usually 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.



Notes on Using Quickstretch

Here is some information that may be useful when using quickstretch.

Quickstretch and Envelopes

If you want to combine quickstretch and envelope deformations on the same object, make sure that the object is a child of the envelope deformers. For example, make the chain root the parent of the enveloped geometry. This way, when the root is translated, the envelope’s center moves with it.

Otherwise if the envelope is not a child of the root, then when the root moves the geometry follows because it is enveloped to the bones, but the center does not move so no quickstretch effect occurs.

Quickstretch and Rigid Bodies

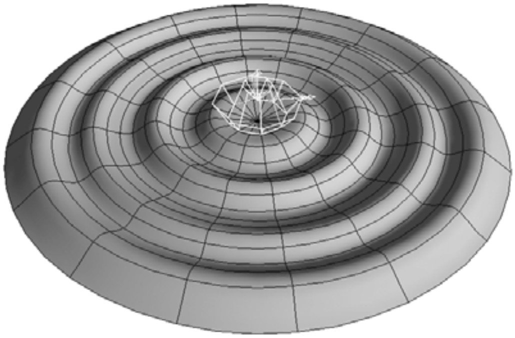
Quickstretch is not supported with rigid body dynamics.

Waves

Waves are animated deformations that travel in both time and space. You can create shock waves, water waves, and other types of natural disturbances with wave deformations.

Waves are animated deformations whose amplitude changes over time. You can:

- Create a wave control object and use it to set various properties of the wave.
- Apply a wave operator to an object so that it is deformed by the wave.



If you are familiar with SOFTIMAGE|3D, you will find that waves are much simpler to work with in XSI. For example, you can save keyframes for a wave control object's position just like for any object's position—you do not need to use a special set of commands.

In the same manner, the wave operator is treated just like any other deformation: you can move points on the deformed object without detaching the wave first.

Wave Control Objects and Wave Operators

There are two basic parts of a wave deformation: the *wave control object* and the *wave operator*.

Wave Control Objects

The wave control object controls the basic parameters that are intrinsic to the wave itself, such as its speed and shape. Each wave control object can be used to deform any number of objects in a scene.

The wave control object is represented by a wireframe icon in the geometry views; there are different icons for the different types of wave. The position of the wave control object also defines the "epicenter" of the wave and its orientation defines the direction of the waves.

Wave Operators

The wave operator deforms an object or cluster by "attaching" a wave control object. The wave operator is a node in the object's operator stack. It controls the parameters that are specific to how the wave affects a particular object, such as the spread of the deformation. You can also use a weight map to modulate the amplitude of the deformation across the surface.

Making Waves

There are two ways to create a wave deformation: you can create and apply a wave at the same time or you can create a wave object first and then manually apply a wave operator later.

To apply a wave deformation to points, clusters, or weight maps, you must use the manual method.

Creating and Applying Waves

If objects are selected when you create a wave control object, the object is created at the object's center and a wave operator is automatically applied. However, this method does not work if points, clusters, or weight maps are selected.

To create and apply a wave to objects

1. Select one or more objects, branches, or models.
2. Choose **Get** > **Primitive** > **Control Object** > **Wave**.

   The wave control object is created (as represented by a wireframe icon in the geometry views) and its property editor opens. The wave object is automatically a child of the deformed object.

   In addition, Wave operators are automatically applied to the objects that were selected.

Applying Waves Manually

If a wave object already exists in your scene, you apply it manually to scene elements. This is the only way to apply wave operators to points, clusters, and weight maps.

To apply a wave to elements

1. If necessary, create a wave object without applying it to anything by choosing **Get** > **Primitive** > **Control Object** > **Wave** with nothing selected.
2. Select one or more objects, branches, models, points, point clusters, or weight maps.
3. Choose **Deform** > **Wave**. This command is available under **Modify** on the Model and Simulate toolbars, as well as under **Deform** on the Animate toolbar.

4. Pick the wave object. Wave operators are applied to the selected elements and the Wave Op property editor opens.

Modifying Wave Control Objects

You can use the wave property editor to set the wave's basic characteristics such as its shape and speed—these are described in the sections that follow. This way, as you change the various parameters in the editor, you can see the effect on an object's deformation.

You can also scale, rotate, and translate the wave control object to define its center and the direction of the waves.

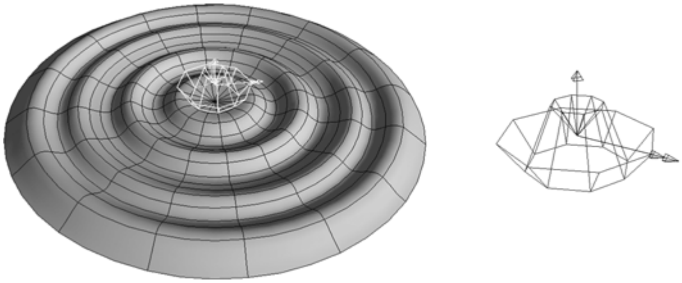
Setting the Wave Shape

There are three parameters that control the wave shape: **Type**, **Displacement Direction**, and **Profile**.

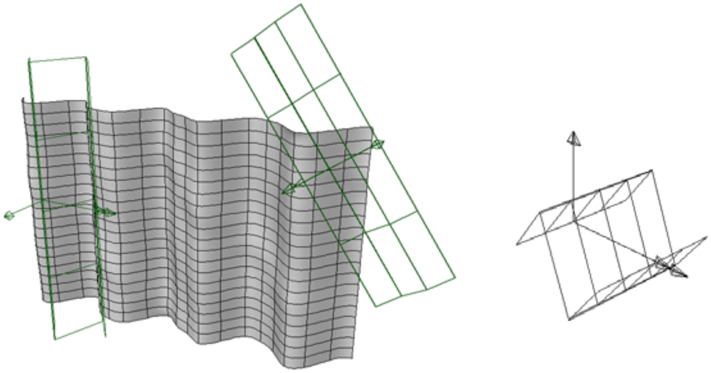
Setting the Wave Type

The Type parameter on the General tab of the wave property editor controls how the wave moves through space. There are three options: circular, planar, and spherical. Each type of wave is represented by a different icon in the geometry views and is shown in the following illustrations.

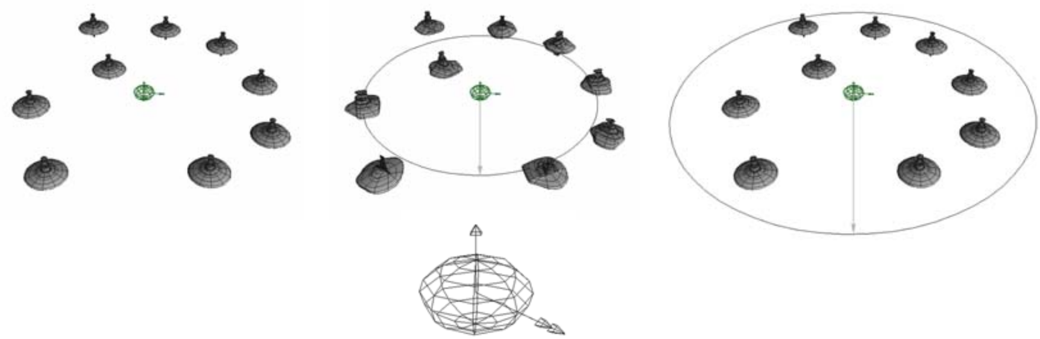
- **Circular**—The waves move out from a point in a circular, planar pattern, like those from a pebble dropped in still water.



- **Planar**—The waves move out from a line in a straight, planar pattern, like waves hitting a beach or a flag fluttering in the wind.



- **Spherical**—The waves moves out from a point spherically in all directions, like the shock waves of an explosion.



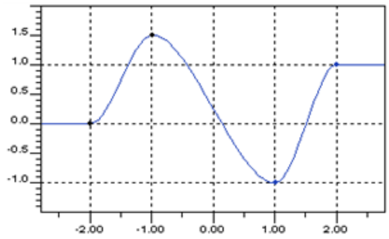
Setting the Displacement Direction

The **Displacement Direction** determines which way the points of deformed objects move when they are displaced:

- **Up** displaces points along the local Y axis of the wave object.
- **Direction** displaces points in the direction in which the wave is moving.
- **Normal** displaces each point along the normal of the deformed object at that point.

Setting the Wave Profile

The shape of the wave's displacement is controlled by the Amplitude Profile curve on the Profiles property page of the wave property editor. You can edit the profile using the mouse and the same keyboard commands as the fcurve editor or right-click to display a menu.



Controlling Periodicity

Periodicity allows the wave to repeat continuously. There are two kinds of periodicity:

- **in Space** repeats the shape of the wave's profile between the first and last key points indefinitely.
- **in Time** repeats the entire wave effect, including the result of **Amplitude** and **Spread** as set in the operator's property editor (not the object's).

  The wave deformation begins again after the **End** frame set on the Time Control tab of the operator's property editor.

Controlling Speed

You can control the velocity and acceleration at which the wave profile moves through space using the **Velocity** and **Acceleration** parameters in the wave control object's property editor. You can even use a negative velocity to make the wave move backward.

Controlling Falloff

You can make the strength of the wave effect fall off after a certain vertical distance from the wave control object. The effect decays linearly between the **Vertical Falloff Start** and **End** values as measured along the wave's local Y axis.

Transforming the Wave Control Object

Translations and rotations may be applied to wave control objects in the same manner as ordinary objects. The wave's effect on objects changes in the obvious way. For example, if you move a circular wave along a grid, you change the point from which the wave emanates.

You can apply scaling as well. The icon changes shape accordingly, but the wave's effect on objects changes only in the following ways:

- For scaling in X and Y, the effect appears as if the profile curve were scaled correspondingly.
- Scaling in Z does not change the effects of circular or spherical waves, but it does change the Z-extents for planar waves. The planar-wave icon makes this quite clear.

Modifying Wave Operators

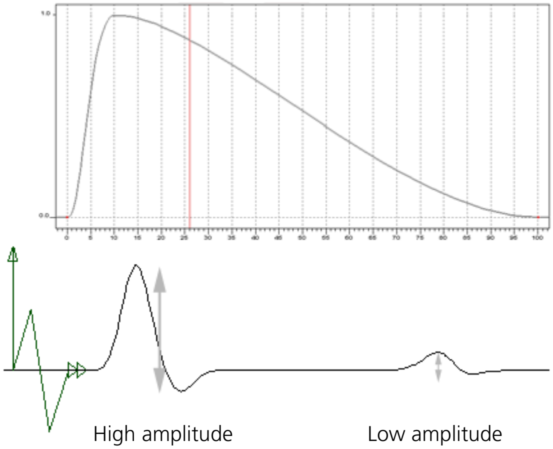
The wave operator deforms an object or cluster by “attaching” a wave control object. The wave operator is a node in the object’s operator stack that controls the parameters that are specific to how the wave affects a particular object, such as the amplitude and spread of the deformation. By default, both the Amplitude and Spread are animated with a function curve.

To make sure that the wave operator gets re-evaluated to update the object’s deformation at every frame, at least one of its parameters must be animated. A single keyframe (that is, a flat, constant-value fcurve) is enough.

By default, both Amplitude and Spread are animated, so the wave operator will be re-evaluated unless you remove the animation.

Controlling Amplitude

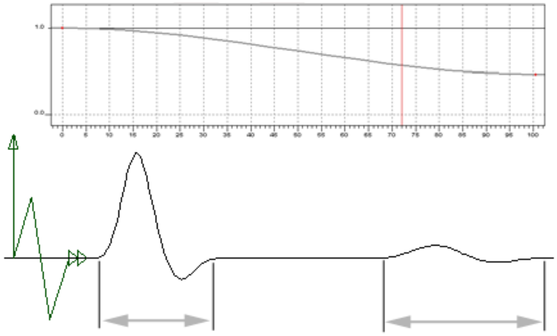
Amplitude provides an overall scaling factor for the wave profile’s height over time. By default it starts at 0, rises sharply to 1, and then decays slowly back to 0. This corresponds to a wave rising rapidly on a surface and then slowly receding back to nothing.



Controlling Spread

Spread gives an overall inverse scaling factor to the X values of the wave profile. For example, a value of 0.5 stretches the profile by 2 horizontally, which has the effect of spreading the wave.

By default, the spread is the constant value 1. Certain types of waves, such as water waves, spread out as they move. For such waves, you should edit the spread function curve to start at 1 and end at a smaller value such as 0.5.



Wave spreads out over time

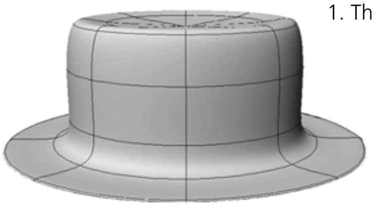
Controlling the Start and End

The options on the Time Control tab of the Wave Op property editor control the time span of the wave effect:

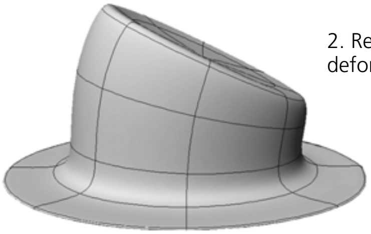
- **Start offset** is the first frame for which the wave begins to affect the object.
- **End** is the last frame for which the wave affects the object.

Muting Deformations

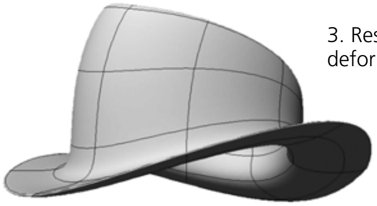
All deformations can be muted. This temporarily turns off the deformation, preventing it from affecting the objects on which it is applied. This can be useful to improve the speed of interaction in large scenes or to make other selections and adjustments.



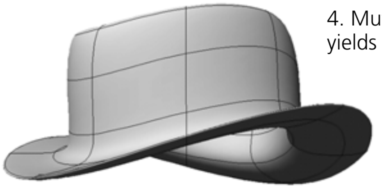
1. The original hat



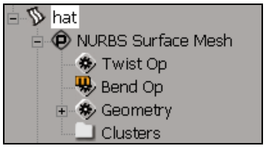
2. Result of a deformation (bend)



3. Result of a second deformation (twist)



4. Muting the Bend yields a twist.



When an operator is muted, the letter **M** appears on its icon in explorer views.

To mute or unmute a deformation

- Do one of the following:
  - Open the deformation’s property editor and toggle the **Mute** option.

    or

  - Right-click the deformation in an explorer view and choose the **Mute** item.

    Note that even if you have selected multiple deformations, only the deformation operator under the mouse pointer in the explorer view is affected by the Mute item on the context menu.

Section II • Polygonal Modeling

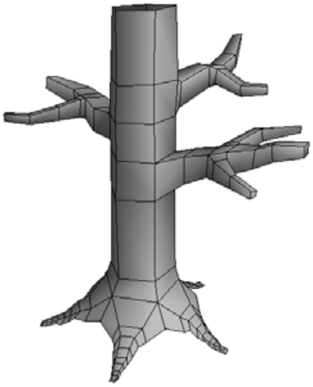
Chapter 7 Fundamentals of Polygonal Modeling

Fundamentals of Polygonal Modeling

Polygon meshes are one of the basic renderable geometry types in SOFTIMAGE|XSI. They are ideally suited for modeling non-organic objects with hard edges and corners, but they can also be used to approximate smooth, organic objects. Polygon meshes are particularly used for games development because of the requirements of most game engines. Polygon meshes are also the basis of subdivision surfaces.

This chapter discusses some basic concepts of polygonal modeling. It also discusses some special selection commands that apply only to polygon meshes, in addition to how to control the viewing and rendering properties of polygon meshes, such as smoothness, using geometry approximation settings.

The chapters that follow describe how to create and modify polygon meshes and subdivision surfaces.

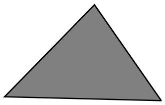


Polygon Meshes

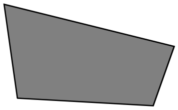
When working with polygon meshes, there are some basic concepts you should understand.

Polygons

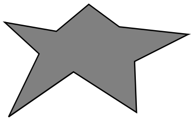
A polygon is a closed 2D shape formed by straight edges. The edges meet at points called *vertices*. There are exactly the same number of vertex points as edges. The simplest polygon is a triangle.



Triangle



Quad



N-gon

Polygons are classified by the number of edges or vertices. Triangles and quadrilaterals (or quads) are the most commonly used for modeling. Triangles have the advantage of always being planar, while quads give better results when used as the basis of subdivision surfaces. Certain game engines require that objects be composed entirely of triangles or quads.

Polygon Meshes

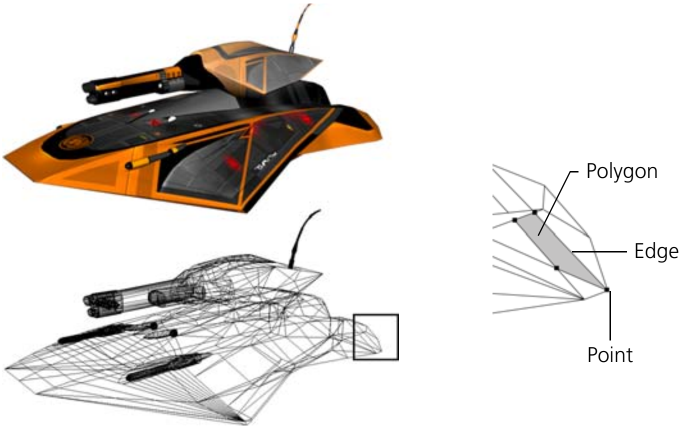
A polygon mesh is a 3D object composed of one or more polygons. Typically these polygons share edges to form a three-dimensional patchwork.

A polygon mesh sphere

However, a single polygon mesh object can also contain discontiguous sections that are not connected by edges. These disconnected polygon “islands” can be created by drawing them directly or by combining existing polygon meshes.

Types of Polygon Mesh Components

Polygon meshes contain several different types of component: points (vertices), edges, and polygons.



- **Points** are the vertices of the polygons. Each point can be shared by many adjacent polygons in the same mesh.
- **Edges** are the straight line segments that join two adjacent points. Edges can be shared by no more than two polygons.

  Edges that are not shared represent the boundary of the polygon mesh object and are displayed in light blue if **Boundaries and Hard Edges** are visible in a 3D view.
- **Polygons** are the closed shapes that make up the "tiles" of the mesh.

Viewing Polygon Mesh Components and Attributes

You can display or hide polygon mesh components, like points and boundaries as well as other attributes, in the 3D views in exactly the same way as other components and attributes. A quick summary is provided here, but for complete details about viewing, see *Chapter 5: Viewing 3D Objects* in the *Fundamentals* guide.

To display or hide components in a single 3D view

Do one of the following:

- Click the eye icon (Show menu) and choose the corresponding component option. You can press Shift to keep this menu open while you toggle multiple items.

  or
- Press Shift+s, or click the eye icon and choose **Visibility Options**. The Camera Visibility property editor contains many more options than the menu, for both selected and unselected objects.

To display or hide components in all open 3D views

Do one of the following:

- Choose **View** on the main menu, and then choose the corresponding component option.

  or

- Choose **View** > **Visibility Options** (**All Cameras**). The Camera Visibility property editor contains many more options than the menu for both selected and unselected objects.

Selecting Components on Polygon Meshes

You can select and work with points, edges, and polygons in the same way as any other type of component using the filters on the Select panel. Selection is fully described in *Chapter 8: Selecting* of the *Fundamentals* guide, and is quickly summarized here.

In addition, there are some commands that apply only to polygon mesh components and are described in this chapter—see *Selecting Polygon Mesh Components* on page 190.

To select components

1. Select the object that owns the components you want to select.

   To select components on multiple objects, select all the associated objects first.

2. Set the selection filter to the desired component type. See *Summary of selection filters* on page 186.

3. If desired, change the selection tool. See *Summary of selection tools* on page 186.

4. Click and drag on the components in a 3D view.

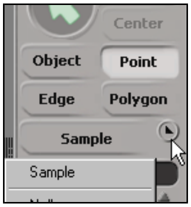
   The specific mouse buttons and modifier keys that you use to perform different actions—like selecting, adding to the selection, toggling, and deselecting—depend on the selection interaction model.

   For more complete details, see *Selecting Components Using the Different Interaction Models* in Chapter 8 of the *Fundamentals* guide.

To select different types of component simultaneously, first select the components of one type, press Shift while changing the selection filter, and then add the other components to the selection.

Summary of selection filters

Polygon Component Filters



Component selection filters determine what type of component you can select on the active geometric objects. The most commonly used filters are available as buttons on the Select panel, while more filters are available from the Filter menu below the buttons.

The component selection filters are context-sensitive; they change depending on what type of object is active. The illustration at left shows the component filters available when a polygon mesh is selected.

There are additional filters on the Filter menu that allow you to select, for example, border points or edges. These are examples of custom filters. For information about creating your own filters, see the SDK documentation.

Summary of selection tools

The selection tools determine the mouse interaction used select elements. You can activate a selection tool with the items on the **Select** > **Tools** menu.

Alternatively, you can press one of the following keys:

- **F7** for the **Rectangle** selection tool.
- **F8** for the **Lasso** selection tool.
- **F9** for the **Freeform** selection tool.
- **F10** for the **Raycast** selection tool.
- **Shift+F10** for the **Rectangle-Raycast** tool.
- **F11** for the **Paint** selection tool (points, polygons, and edges only).

When selecting edges with the Rectangle or Lasso tools, the **Rectangle Select: Surround Edges** preference controls whether you need to enclose all or part of an edge to select it.

When selecting polygons with the Raycast tool, the **Raycast Select: Include Backfacing Polygons** preference controls whether you can select polygons whose normals face away from the camera.

- When this option is on, you can select a backfacing polygon by clicking on a silhouette edge that is shared by that polygon.
- When it is off, you can select only polygons whose normals face the camera.

To access these preferences, choose **File** > **Preferences** from the main menu, then expand **Tools** and click **Selection**.

Summary of selection modes

Selection modes are combinations of a filter and tool mapped to a single key for convenience. Use any of the following keys:

- **t** for **Rectangle Point** mode.
- **e** for **Rectangle Edge** mode.
- **i** for **Raycast Edge** mode.
- **y** for **Rectangle Polygon** mode.
- **u** for **Raycast Polygon** mode.

Selecting Points that Overlap Others

Polynode bisectors can be useful when selecting or moving points that overlap others on a polygon mesh object.

To select or move a point that overlaps others

1. Open the Camera Visibility property editor of a 3D view, for example, by pressing Shift+s.
2. On the Components tab, set **Show Polynode Bisectors** to **Show While Drawing Points**.

   With this option, bisectors are displayed as small purple lines at the corners of polygons whenever points are visible.
3. Activate point selection (for example, press **t** for Rectangle Point Selection) or the Move Point tool (press **m**).
4. Click on a bisector to affect the corresponding point.

Planar and Non-planar Polygons

When an individual polygon on a polygon mesh is completely flat, it is called *planar*. All its vertices lie in the same plane, and are thus *coplanar*. Planar polygons give better results when rendering.

Planar polygon on the ground plane with normals visible.

Non-planar polygon created by moving a point below the ground plane.

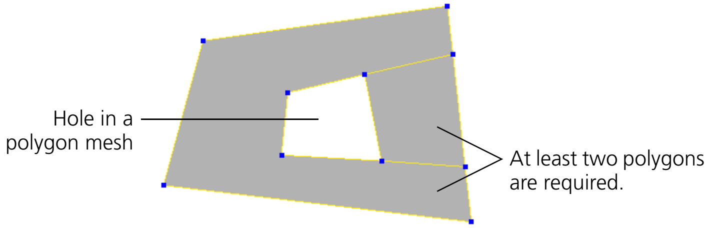
Triangles are always planar because any three points define a plane. However, quadrilaterals and other polygons can become non-planar, particularly as you move vertices around in 3D space. When objects are automatically tessellated before rendering, non-planar polygons are divided into triangles.

When adding new polygons, there is a preference setting that prevents you from drawing non-planar polygons: see *Adding and Editing Polygons (Drawing)* on page 214.

Valid Meshes

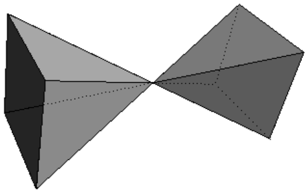
SOFTIMAGE|XSI has strict rules for valid polygon mesh structures and won't let you create an invalid mesh. Some of the rules are:

- Every point must belong to at least one polygon.
- Every edge must belong to at least one polygon.
- A given point can be used only once in the same polygon.
- All edges of a single polygon must be connected to each other. Among other things, this means that you cannot have a hole in a single polygon. To get a hole in a polygon mesh, you must have at least two polygons.



- Edges cannot be shared by more than two polygons. Tri-wings are not supported.
- XSI does support one case of non-manifold geometry. A single point can be shared by two otherwise unconnected parts of a single mesh object.

  If you export geometry from XSI, remember that such geometry may not be considered valid by other applications.



A non-manifold geometry that is valid in XSI.

Materials and Textures on Polygon Meshes

In addition to basic materials and textures, polygon meshes support a couple of specialized shading features:

- **Local materials and textures on polygon clusters**: This lets you apply a material or texture to specific polygons on an object. For more information about local materials and textures, see *Assigning Materials to Polygon Selections and Polygon Clusters* in Chapter 3 of the *Shaders, Lights & Cameras* guide.
- **Vertex colors**: This is a shading technique that stores color information in the vertices, allowing for fast shading in certain game engines. For more information about color at vertices, see *Chapter 12: Working with Vertex Colors* of the *Shaders, Lights & Cameras* guide.

For more information about materials and textures in general, see the *Shaders, Lights & Cameras* guide.

Preserving Textures While Modeling

Most modeling operations on polygon meshes are texture-safe and preserve existing texture projections. However, you may encounter some problems where there are texture seams.

For example, if you collapse an edge that crosses a texture seam, the new merged point uses the average texture UVs of the previous two points and the result may not be visually correct.

Selecting Polygon Mesh Components

This section describes some special commands that can be used only to select points, edges, and polygons on polygon mesh objects.

- **Select** > **Grow Selection** adds to the selection those components that are both adjacent to and of the same type as the current selection.
- The commands on the **Select** > **Select Adjacent** submenu select the components of the specified type that are adjacent to the current selection.
- **Select** > **Select Edge Loop** selects connected edges passing through vertices where four edges meet.
- The commands on the **Select** > **Select n-Sided Polygons** submenu allow you to select polygons based on their number of sides: triangles, quads, and others.

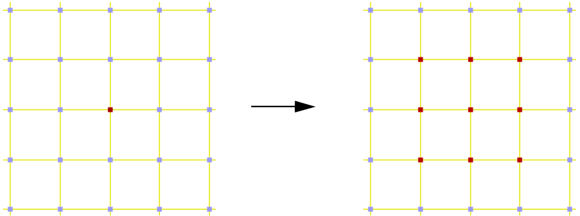
For a description of the standard methods that can be used to select any type of component, see the summary *Selecting Components on Polygon Meshes* on page 185 of this guide or *Selecting Components* in Chapter 8 of the *Fundamentals* guide.

Growing the Selection

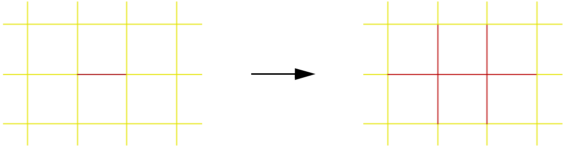
When polygon mesh components are selected, you can add adjacent components of the same type to the selection by growing it.

To grow the selection

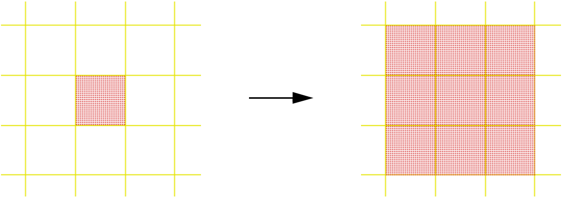
1. Select some polygons, edges or points on a polygon mesh object.
2. Choose **Select** > **Grow Selection** from the Select panel or press Shift+=.
   - If points are selected, then all points that are share a polygon are added to the selection.



   - If edges are selected, then all edges that share a point are added to the selection.



- If polygons are selected, then all polygons that share a point are added to the selection.

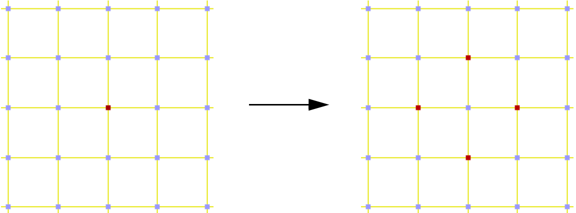


Selecting Adjacent Components

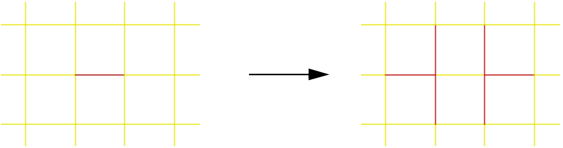
When polygon mesh components are selected, you can select the components of a specific type that are adjacent to the current selection.

To select adjacent components

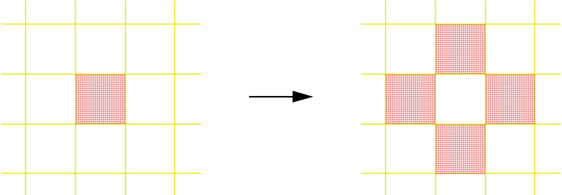
1. Select some polygons, edges, points, or clusters on a polygon mesh object.
2. Choose one of the items on the **Select** > **Select Adjacent** submenu on the Select panel: **Points**, **Edges**, or **Polygons**.
   - If you choose a different type of component than the current selection, all components of that type that share or are shared by the current selection become selected. For example, if polygons are selected and you choose **Points**, then all points that are shared by the polygons become selected.
   - If points are selected and you choose **Points**, then all points that share an edge with the selected ones become selected.



   - If edges are selected and you choose **Edges**, then all edges that share a point with the selected ones become selected.



- If polygons are selected and you choose **Polygons**, then all polygons that share an edge with the selected ones become selected.



Selecting Edge Loops

The **Select** menu has commands for selecting edge loops and parallel edge loops on polygon meshes.

As an alternative to these commands, you can use the range and loop selection features of the component selection tools. See *Selecting Ranges and Loops of Components* in Chapter 8 of the *Fundamentals* guide.

Selecting Edge Loops

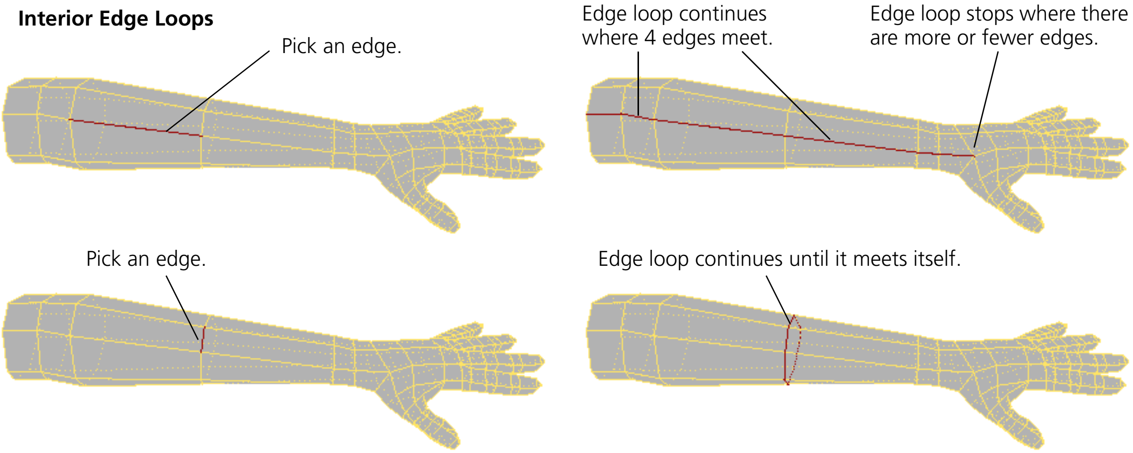
Edge loops are lines of connected edges. Edge loops are useful in polygon modeling because they define the “flow lines” or “contour lines” of your object.

There are two commands for selecting edge loops on the **Select** menu:

- **Select Edge Loop (no corners)**
- **Select Edge Loop (around corners)**

For interior edges, these commands do the same thing. At a vertex where four edges meet (+ junctions), the edge loop continues along the middle of the three other edges. The edge loop stops at vertices where two, three, five or more edges meet, or where it meets itself on the other side of the object.

Interior Edge Loops



Pick an edge.

Edge loop continues where 4 edges meet.

Edge loop stops where there are more or fewer edges.

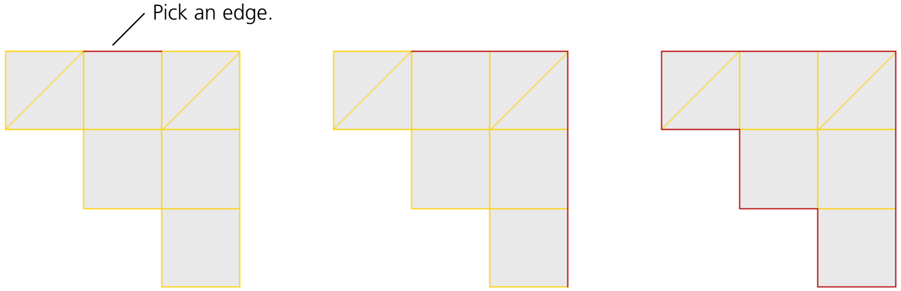
Pick an edge.

Edge loop continues until it meets itself.

For boundary edges, the commands behave differently:

- **Select Edge Loop (no corners)** extends the selection along the top of T and Y junctions.
- **Select Edge Loop (around corners)** selects the entire connected boundary.

Boundary Edge Loops



No Corners
Passes through T and Y junctions; stops anywhere else.

Around Corners
Selects entire connected boundary.

To select edge loops

1. Choose **Select** > **Select Edge Loop (no corners)** or **Select** > **Select Edge Loop (around corners)**.
2. Pick a polygon mesh object. The object becomes active for edge loop selection.
3. Pick one or more edge loops as desired:

   - Click or middle-click on an edge to select the corresponding edge loop.
   - Shift+click on an edge to add the corresponding loop to the selection.
   - Ctrl+click on an edge to toggle the corresponding loop.
   - Ctrl+Shift+click on an edge to remove the corresponding loop from the selection.
   - Activate additional polygon mesh objects for edge loop selection by picking them with the left or middle mouse button.

4. When you have finished, right-click or press Esc to terminate the picking session.

Selecting Parallel Edge Loops

Parallel edge loops consist of the non-adjacent edges that are opposite each other along a strip of quads.

In some cases, **Select Parallel Edge Loop** also selects edges on non-quads if they are parallel enough.

To select parallel edge loops

Parallel edge loops are strips of unconnected edges along a row of quadrilateral polygons.

1. Choose **Select** > **Select Parallel Edge Loop**.
2. Pick a polygon mesh object. The object becomes active for edge loop selection.
3. Pick one or more edge loops as desired:
   - Click or middle-click on an edge to select the corresponding edge loop.
   - Shift+click on an edge to add the corresponding loop to the selection.
   - Ctrl+click on an edge to toggle the corresponding loop.
   - Ctrl+Shift+click on an edge to remove the corresponding loop from the selection.
   - Activate additional polygon mesh objects for edge loop selection by picking them with the left or middle mouse button.
4. When you have finished, right-click or press Esc to terminate the picking session.

Selecting Triangles, Quads, or Other Polygons

You can quickly select polygons based on their number of sides: triangles, quads, or other polygons.

To select triangles, quads, or other polygons

1. Select one or more polygon mesh objects.
2. Choose one of the following commands from the **Select** > **Select n-Sided Polygons** submenu:
   - **Triangles:** Selects all three-sided polygons on the active objects.
   - **Quads:** Selects all four-sided polygons (quadrilaterals) on the active objects.
   - **Five Sides or More:** Selects all the non-triangle, non-quad polygons.

You can also use the **Triangle**, **Quad**, and **N-gon** filters to select these types of polygons interactively using one of the selection tools. These filters are available on the Filters menu (the triangle above the Selection boxes on the Select panel).

Showing Normals on Polygon Meshes

There are some specific options for displaying normals that apply only to polygon meshes.

- Instead of (or in addition to) displaying shading normals at the points, you can display the geometric normals in the centers of polygons.
- You can control whether the normals are displayed for the entire selected object, or only for selected components.

All these options can be set on the Attributes tab of the Camera Visibility property editor:

- To display the Camera Visibility property editor for a single 3D view, press Shift+s while the pointer is over that view. Alternatively, choose **Visibility Options** from the **Show** menu (eye icon).
- To display the Camera Visibility property editor for all open 3D views, choose **View** > **Visibility Options (All Cameras)** from the main menu.

In addition, some of these options can be toggled directly from the Show menu (eye icon) for a single 3D view, or from the View menu for all open views.

Press Shift to keep the Show menu (eye icon) open when you want to toggle multiple options.

Displaying Shading and Geometric Normals

There are two independent options for displaying normals on polygon meshes:

- **Normals** displays the shading normals at vertex locations. This option also displays normals on NURBS surfaces.
- **Polygon Normals** displays the geometric normals in the center of polygons. This option affects only polygon meshes.

These options can be set on the Attributes tab of the Camera Visibility property editor, or directly from the Show menu (eye icon) or View menu.

Displaying Normals on Selected or Selected Components

When displaying normals on selected polygon meshes, you can decide whether the normals are shown for the entire object or only for selected components. Open the Camera Visibility property editor and display the Attributes tab, then set F**or Selected Objects** in the **Show Normal**s box to the desired value:

- **Show On Whole Object** always displays normals on the entire object.
- **Show On Selected Components** displays normals only on the selected points or polygons, or the points and polygons adjacent to the selected edges.
- **Combo (depends on selection mode)** displays normals on the entire object when an object selection filter is active, and on selected components when a component selection filter is active.

Viewing and Rendering Polygon Meshes

You can control the smoothness and other viewing properties of polygon meshes using geometry approximation. Geometry approximation also allows you to optimize playback and interaction of polygon meshes.

Applying and Editing Geometry Approximation

By default, the scene root node has a Geometry Approximation property which is inherited by every object in the scene. You can apply a local Geometry Approximation property to selected objects, hierarchies (in branch mode), groups, layers, and partitions. This allows you to control the display and rendering of different objects in different ways.

To apply a local Geometry Approximation property

1. Select an element: object, hierarchy (in branch mode), group, layer, or partition.
2. Choose **Get** > **Property** > **Geometry Approximation** from any toolbar.

To edit an existing Geometry Approximation property

1. Select an element.
2. Display its viewing or rendering properties, then click the Geometry Approximation tab.

Geometry Approximation on Polygon Meshes

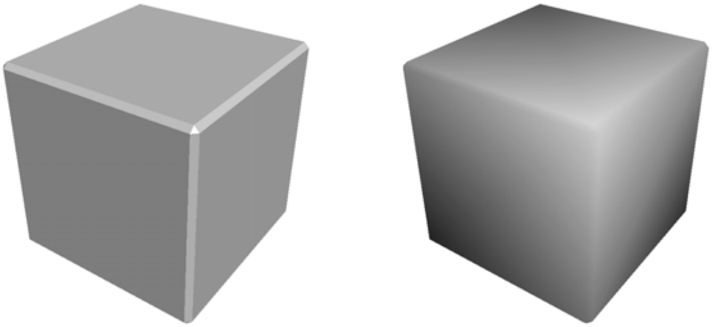
Geometry Approximation is a multipurpose property that controls the display and rendering of polygon meshes, NURBS curves and surfaces, hair, and objects with displacement maps.

Only the settings on the Polygon Mesh property page are described in detail in this section—for more information see *Setting an Object's Geometry Approximation* in Chapter 3 of the *Rendering & Compositing* guide.

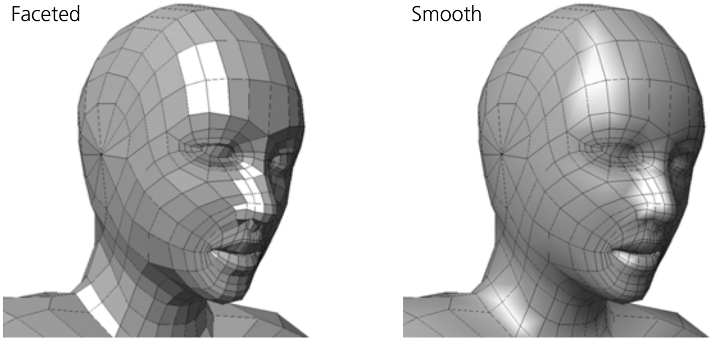
Discontinuity—Faceted and Smooth Polygons

The **Discontinuity** parameters on the Polygon Mesh page of the Geometry Approximation property editor control whether the objects are faceted or smooth at the edges.

Faceted polygons are appropriate for geometric shapes like dice.

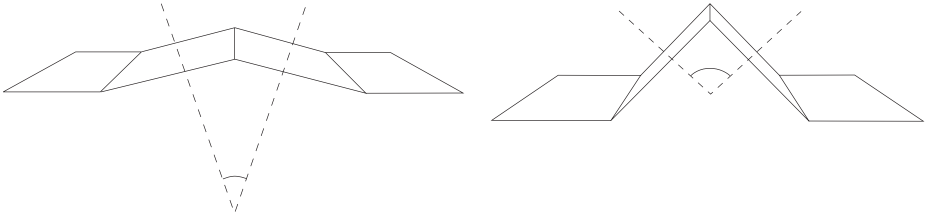


Smooth polygons are appropriate for organic shapes like faces.



The illusion of smoothness is created by averaging the normals of adjacent polygons. When normals are averaged in this way, the shading is a smooth gradient along the surface of a polygon. When normals are not averaged, there is an abrupt change of shading at the polygon edges.

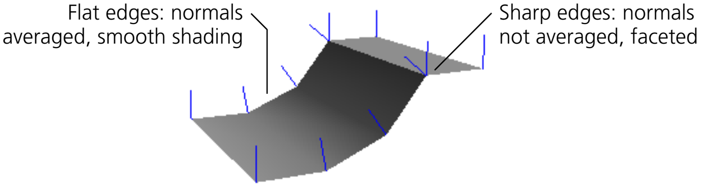
Automatic discontinuity lets you turn off the averaging of normals for sharper edges and the discontinuity **Angle** lets you specify how sharp edges must be before they appear faceted. If the dihedral angle (angle between normals) of two adjacent polygons is less than the Discontinuity Angle, the normals are averaged; otherwise, they are not averaged.



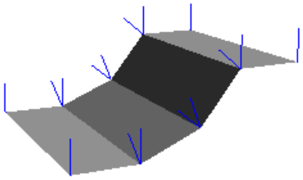
Dihedral angles: flatter edges have small angles and sharper edges large angles.

You can achieve different effects by adjusting these two parameters:

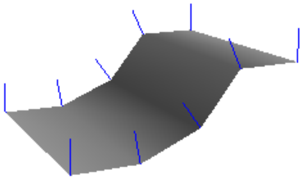
- If **Automatic** is on, then the **Angle** determines the threshold for faceted polygons.


Flat edges: normals averaged, smooth shading
Sharp edges: normals not averaged, faceted

- If **Automatic** is on and **Angle** is 0, the object is completely faceted.



- If **Automatic** is off, the object is completely smooth.



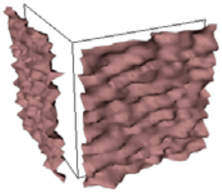
For faster interaction, turn off Automatic discontinuity. This also avoids shading artifacts that can result if an edge crosses the discontinuity Angle, thus becoming smooth or faceted as the object deforms during animation.

Instead of relying on Automatic discontinuity for faceted edges, mark the specific edges as hard—see *Discontinuity on Selected Edges* on page 199.

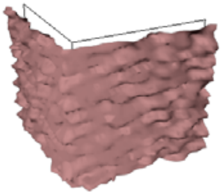
There is no option to keep exactly computed normals as there is in SOFTIMAGE|3D.

Discontinuity and Displacement

Discontinuity between polygons also affects displacement at render time. If there is a discontinuity across an edge, the displacement occurs in different directions on either side of the edge—the result is “exploded” polygons!



With displacement, discontinuous edges break apart.

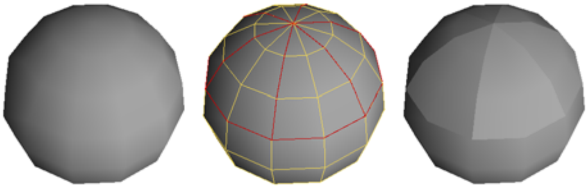


Continuous edges stay together.

If you are using displacement on polygon meshes, make sure that automatic discontinuity is off or that the angle threshold is high enough so that your meshes don’t come apart at the seams.

Discontinuity on Selected Edges

In addition to setting the geometry approximation for an entire object, you can make selected edges discontinuous by marking them as “hard.” This is the equivalent of **Edge Flag** > **Discontinuity** in SOFTIMAGE|3D.



Selected edges marked as hard.

To mark edges as hard

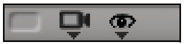
1. Select some edges.
2. Choose **Modify** > **Component** > **Mark Hard Edge/Vertex** from the Model toolbar.

The **Mark Hard Edge/Vertex** command is also used to create sharp creases in subdivision surfaces.

To toggle the display of hard edges

When you display hard edges, they are highlighted in dark blue. This may be useful if your display mode is set to Wireframe and you can't see the effect of hard edges on the shading.

Do one of the following:



- To toggle the display of hard edges in a single 3D view, click the eye icon (Show menu) and choose **Boundaries and Hard Edges**.

  or

- To toggle the display of hard edges in all open 3D views, choose **View** > **Boundaries and Hard Edges** from the main menu.

  Polygon mesh edges that have been marked as hard are highlighted in dark blue, while boundary edges are light blue. You may need to turn off **View** > **Clusters (except Sample)** to see the effect on hard edges.

Subdivisions

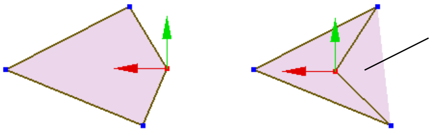
The **Rule**, **OGL Level**, and **Render Level** parameters are used with subdivision surfaces. For details, see *Chapter 9: Subdivision Surfaces* on page 325.

OGL Triangulation Update

The **OGL Triangulation Update** parameter on the Polygon Mesh page of the Geometry Approximation property editor controls the balance between speed and accuracy when interacting with polygon mesh objects as well as when playing back animation in the OpenGL display modes such as Hidden Line, Constant, Shaded, and Textured. There are four settings:

- **Recreate at each refresh** is the most accurate and always recalculates triangulation. However, playback and interaction are slowest in this mode.
- **Skip during interaction** does not recompute triangulation of polygons while you are working and waits for you to pause before retriangulating. The display updates faster as you modify objects, but you may see some artifacts in the OpenGL views, for example, when a polygon becomes concave while manipulating points. Triangulation is still recomputed when required during playback.
- **Skip during interaction and playback** does not recompute triangulation of polygons while you are working nor while animation is playing back. Again, this mode is faster but you may see some artifacts in the OpenGL views.

- **Update only when topology changes** never retriangulates unless you modify the topology, for example, by adding or removing points. This mode is the fastest but potentially creates the most artifacts in the OpenGL views.



Lack of retriangulation can produce an artifact while moving points.

Angle-based Normals

The **Angle-based normals** option controls the method used to calculate the normal at each vertex when rendering:

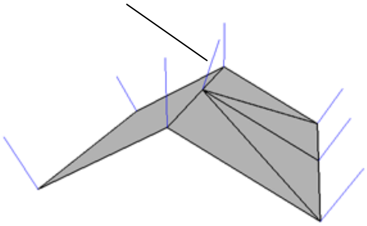
- When off, XSI uses the unweighted average of the adjacent polygon normals.

  This option is substantially faster. Although it is less accurate, the result may not be noticeable in many situations.

- When on, XSI uses the average of the adjacent polygon angles, weighted according to the angle of the corresponding polygon at the point.

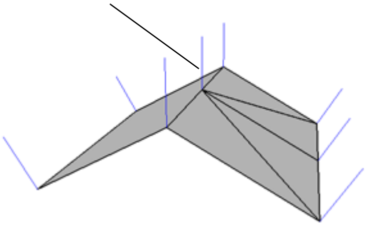
  The result is more accurate in some situations but takes longer to calculate.

Normal is skewed toward the side with more polygons because the average is not weighted.



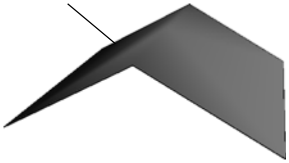
Angle-based normals = off

Normal is more accurate because the average is weighted according to the angles of the adjacent polygons.

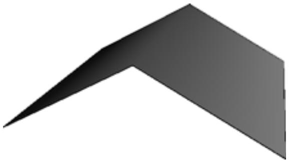


Angle-based normals = on

Skewed normal causes a shading artifact in the rendered image.



Angle-based normal results in correct shading.



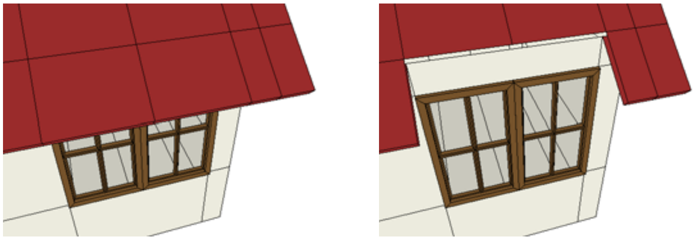
Displacement Mapping

The options on the Displacement page of the Geometry Approximation property editor are used when you apply displacement maps to polygon meshes and other objects. For more information, see *Creating a Displacement Map* in Chapter 10 of the *Shaders, Lights & Cameras* guide.

If you have set surface approximation options for a polygon mesh object in SOFTIMAGE|3D, they are translated into the corresponding options on the Displacement page when you import the object into SOFTIMAGE|XSI.

Hiding Polygons

You can hide specific polygons on a polygon mesh object. This helps you see what you're doing, particularly when working on dense, complex meshes.

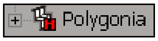


The polygons on the sloping roof make it difficult to select and work with the polygons on the window. Hiding parts of the roof solves the problem.

When they are hidden, polygons are "out of play." They normally aren't drawn or rendered and can't be selected interactively. (However, you can set a 3D view to show hidden polygons and then select them—see *Controlling the Display of Invisible Polygons* on page 207.)

There are two basic ways to hide and unhide polygons:

- Using the Hide commands on the View menu.
- Applying a Visibility property manually to a polygon cluster. This method is more powerful because you can set the view and render visibilities separately.



When a polygon mesh object contains hidden polygons, its icon in the explorer displays a red H.

Hiding and Unhiding Polygons Using Commands

You can hide and unhide polygons quickly using the View commands on the main menu.

To hide selected polygons

If certain polygons are in your way, it is easy to select them and then hide them.

1. Select some polygons or polygon clusters to hide.
2. Do one of the following:
   - Press h or choose **View** > **Hide/Unhide Selection**. This command is a toggle that lets you hide or unhide polygons.

     The polygons are still selected, so they are still drawn highlighted in the 3D views. However, they will disappear as soon as you select something else if **Extended Component Selection** is off on the Select menu.

If Extended Component Selection is on, you cannot use the middle mouse-button to deselect the hidden polygons. However, you can deselect them using other commands such as **Select** > **Deselect All** (Ctrl+Shift+a) or **Deselect All Using Filte**r (Ctrl+Shift+f).

or

- Choose **View** > **Hide and Deselect**. The polygons become hidden and are no longer selected.

To hide unselected polygons

If you want to work with specific polygons, it may be easier to select the polygons you are interested in and hide all the others.

1. Select the polygons or polygon clusters that you want to keep unhidden.
2. Press Ctrl+Shift+h or choose **View** > **Hide Unselected Objects/Polygons**.

To unhide specific polygons

1. Select the polygons to unhide. You may need to show invisible polygons first (see *Controlling the Display of Invisible Polygons* on page 207).
2. Press h or choose **View** > **Hide/Unhide Selection**. This command is a toggle that lets you hide or unhide polygons. However, if at least one of the polygons is not hidden, then all polygons become hidden.

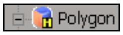
To unhide all polygons on an object

1. Select one or more objects.
2. Press Ctrl+h or choose **View** > **Unhide All Polygons**.

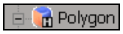
   Note that this command does not affect clusters with a Visibility property.

Applying a Visibility Property to Polygons

The Visibility property lets you control the display of polygon clusters independently for the 3D views and rendered images. You can also animate the visibility of polygons.



- If a cluster has both its view and render visibilities turned off, the cluster's icon is marked with a yellow H in the explorer.



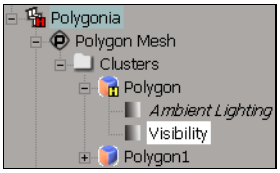
- If a cluster has one or other (but not both) visibilities turned off, the cluster's icon is marked with a gray H in the explorer.

To apply a Visibility property to polygon clusters

1. Select some polygons or polygon clusters.
2. Choose **Get** > **Property** > **Polygon Cluster Visibility** from any toolbar.
   - If you selected polygons in step 1, a cluster is automatically created.
   - If you selected multiple clusters, a separate Visibility property is applied to each.

The Visibility property editor opens, allowing you to set the view and render visibility of the cluster as described in the next section, *To set the visibility of polygons* on page 205.

To set the visibility of polygons



1. Open the Visibility property editor. You can find the Visibility property under any polygon cluster with an applied Visibility property.
2. Set or animate the parameters:
   - **View Visibility** controls whether polygons are drawn in the OpenGL display types (Wireframe, Hidden Line Removal, Constant, Shaded, Textured, Realtime Shaders, and so on).
   - **Render Visibility** controls whether polygons are drawn when rendering with mental ray, such as when rendering to file or using the render region.

To unhide all polygon clusters on objects

1. Select one or more objects that contain clusters with Visibility properties.
2. Choose **View** > **Unhide All Polygon Clusters** from the main menu.

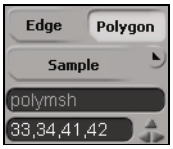
   The **View Visibility** and **Render Visibility** parameters are toggled on for all clusters with Visibility properties on selected objects.

Working with Hidden Polygons

When working with hidden polygons, there are some things to be aware of.

Selecting Hidden Polygons

Hidden polygons cannot be selected interactively in the 3D views using the various selection tools and the Polygon selection filter. However, this does not mean that hidden polygons can't be selected at all:



polymsh

33,34,41,42

Selection Name boxes

- If you know the component indices of the polygons, you can select them by typing in the Selection Name boxes (see *Selecting by Name* in Chapter 8 of the *Fundamentals* guide) or by using scripts (see the *Chapter 2: Scripts* in the *Customization* guide).
- You can use the component navigation keys to select hidden polygons (see *Navigating Component Selection* in Chapter 8 of the *Fundamentals* guide).
- You can set a 3D view to show invisible polygons (see *Controlling the Display of Invisible Polygons* on page 207); this allows you to select hidden polygons interactively in that 3D view.
- The **Grow Selection**, **Select Adjacent** > **Polygons**, and the **Select n-Sided Polygon** commands on the Select menu can be used to select hidden polygons.

  Other commands, such as **Select All Using Filter** and **Invert Using Filter,** do not select hidden polygons.

When selected, hidden polygons are always highlighted regardless of the 3D view's settings.

You can select a polygon cluster interactively if at least one of its member components is visible. Of course, you can always select any cluster using an explorer. If a cluster is selected, then **Select** > **Select Members/Components** selects all its polygons whether or not they are visible.

Transforming and Deforming Hidden Polygons

If hidden polygons are selected, they can still be transformed interactively in the 3D views.

- When Proportional modeling is on, hidden polygons may be affected when you move or transform nearby components. For more information about proportional modeling in general, see *Using Proportional Modeling* on page 112.

- Similarly when Symmetry mode is on, hidden polygons will be affected if you move or transform symmetric components. For more information about Symmetry mode, see *Manipulating Components Symmetrically* on page 117.

Hidden polygons can also be affected by volume and other deformers.

Snapping to Hidden Polygons

Normally, you cannot snap to hidden polygons. In addition, you cannot snap to edges or vertices that are completely surrounded by hidden polygons (and therefore also not visible).

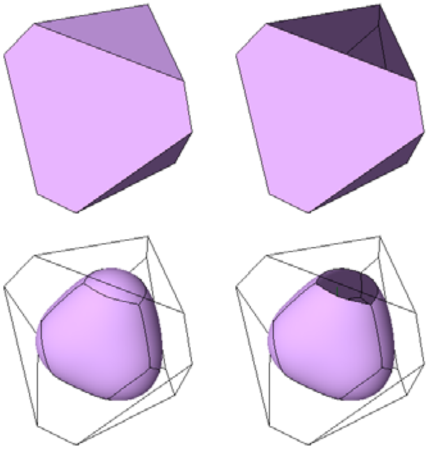
However, you can snap to these components if they are visible in a 3D view. For information about displaying invisible polygons, see *Controlling the Display of Invisible Polygons* on page 207.

Merging Meshes with Hidden Polygons

If you merge meshes with hidden polygons or clusters with Visibility properties, the invisibility is not propagated to the merged result. This also holds true for other operations that generate new mesh objects, like Booleans.

Hiding Polygons on Subdivision Surfaces

When you create a subdivision surface using **Get** > **Property** > **Geometry Approximation**, the hidden polygons are propagated from the polygon mesh hull to the subdivision surface.



Hidden polygons are propagated from the polygon mesh hull to the subdivision surface.

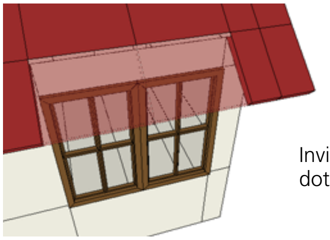
However, hidden polygons are not propagated when you subdivide using **Create** > **Poly. Mesh** > **Subdivision** on the Model toolbar, nor when you duplicate using **Edit** > **Duplicate/Instantiate** > **Duplicate Using Geometry Approx**.

For more information about subdivision surfaces in general, see *Chapter 9: Subdivision Surfaces* on page 325.

Controlling the Display of Invisible Polygons

Sometimes it's necessary to temporarily show hidden polygons. For example you may want to unhide specific polygons, but you have to see the polygons to select them.

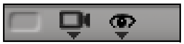
You can set a 3D view to show invisible polygons. Use the command on the eye icon menu (Show menu) to quickly set it for all objects or use the Camera Visibility property editor to set it independently for selected and unselected objects.



Invisible polygons are drawn with dotted-line edges and X-ray shading.

To control the display of invisible polygons on all objects

Do one of the following:



- To control display of invisible polygons in a single 3D view, click the eye icon (Show menu) and toggle the **Invisible Components** option.

  or

- To control display of invisible polygons in all open 3D views, choose **View** > **Invisible Components** from the main menu.

To control the display of invisible polygons on selected and unselected objects independently

1. Do one of the following:

   - To control a single 3D view, move the mouse pointer over that view and press Shift+s or click the eye icon (Show menu) and choose **Visibility Options**.

     or

   - To control all open 3D views, choose **View** > **Visibility Options** (**All Camera**) from the main menu.

   The Camera Visibility property editor opens.

2. On the Components tab, toggle the **Invisible Components** options. The left box controls display of selected objects and the right box controls display of unselected objects.

Polygon Mesh Tips & Tricks

This section includes some miscellaneous tips and techniques that you might find useful when working with polygon meshes.

Accessing Commands Quickly Using Context Menus

Many polygon modeling commands are available on the context menu that appears when you Alt+right-click a polygon mesh object in a 3D view (Ctrl+Alt+right-click on Linux).

Similar, more specific context menus are available when you right-click on an object while polygons, edges, or vertices are tagged.

Displaying Information on Polygon Mesh Objects

Press Shift+Enter or choose **Edit** > **Info Selection** to get useful information about the selected polygon mesh object.

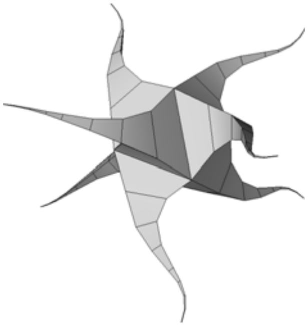
- **Components** shows the number of points, edges, and polygons.
- **Polygons** classifies the polygons into triangles, quads, and n-gons.
- **Triangles** shows the number of triangles after tessellating for rendering.

Importing Polygon Meshes from SOFTIMAGE|3D

Before importing a polygon mesh object into SOFTIMAGE|XSI, use **Effect** > **GC_ValidateMesh** in SOFTIMAGE|3D first. This helps eliminate problems that may be caused by invalid meshes.

Chapter 8 Modeling Polygons & Polygon Meshes

Modeling Polygons and Polygon Meshes



The polygon modeling tools in XSI let you create and modify polygon mesh objects in a variety of ways. You can:

- Create new polygon mesh objects in a variety of ways.
- Add and edit components by drawing, splitting, welding, or subdividing.
- Remove components by deleting, collapsing, or dissolving.
- Disconnect components to separate them from their neighbors.
- Duplicate, extrude, and inset components to create protuberances and indentations.
- Offset polygons to grow or shrink their contours.
- Knife or slice polygon meshes to subdivide them along a plane.
- Dice polygons to subdivide them along multiple axes.
- Symmetrize polygons to mirror them across a plane.
- Bevel components to round off corners.
- Invert polygons to change the direction of their normals.
- Bridge polygons to connect two areas of the same mesh.
- Combine polygon meshes by blending or merging.
- Perform Boolean operations to obtain the difference, intersection, or union of two polygon meshes.
- Filter components to clean up polygon meshes by removing or welding edges, points, and polygons that match specific criteria.
- Reduce the number of polygons in an object by collapsing edges.
- Quadrangulate polygon meshes by dissolving pairs of triangles, or triangulate them by subdividing quads and n-gons.
- Convert curves to polygon meshes.
- Convert NURBS surfaces to polygon meshes.
- Create new mesh objects by extracting specific polygons from existing meshes.

In addition to the polygon mesh-specific actions described in this chapter, you can manipulate polygons, edges, and vertices like any other component as described in *Chapter 5: Manipulating Components* on page 89.

You can also deform polygon meshes as described in *Chapter 6: Deformations* on page 131.

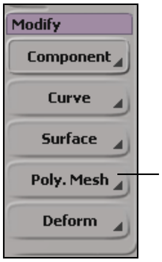
Creating Polygon Meshes

In polygonal modeling, you generally start with a polygon mesh object and then modify it, for example, by adding edges, welding points, subdividing polygons, and so on.

There are several different ways to get the polygon mesh object to start with:

- You can get primitive polygon mesh objects or create text as described in *Chapter 2: Starting From Scratch: Primitives and Text* on page 35.
- You can build polygon meshes from curves by performing operations such as extruding, lofting, and revolving, as described in*Chapter 4: Generating Objects from Curves* on page 73.
- You can draw a new polygon mesh object from scratch using the Add/Edit Polygon tool. Simply make sure that no other polygon mesh object is selected before choosing **Modify** > **Poly. Mesh** > **Add/Edit Polygon Tool** from the Model toolbar or pressing **n**, and then click in the 3D views to place vertices. For complete details about using the Add/Edit Polygon tool to start new polygons, connect to edges or vertices, move vertices, and remove vertices, see *Adding and Editing Polygons (Drawing)* on page 214.
- You can build polygon meshes by combining other ones using Blend, Merge, or Boolean operations—see *Blending and Merging Polygon Meshes* on page 278 and *Performing Boolean Operations on Polygon Meshes* on page 283.
- You can create a planar or solid polygon mesh by tessellating curves as described in *Converting Curves to Polygon Meshes* on page 306.
- You can convert a NURBS surface mesh to a polygon mesh as described in *Converting NURBS Surfaces to Polygon Meshes* on page 321.
- You can extract or detach polygons from an existing polygon mesh as described in *Extracting Polygons* on page 322.
- You can create a subdivided version of a polygon mesh as described in *Chapter 9: Subdivision Surfaces* on page 325.
- You can import models and scenes with polygon mesh objects from SOFTIMAGE|3D or other programs as described in *Chapter 24: Importing and Exporting* of the *Fundamentals* guide.

Adding and Editing Polygon Mesh Components



Add and edit polygon mesh components

There are a variety of tools and operators for adding points, edges, and polygons to polygon mesh objects. They are all available from the **Modify** > **Poly. Mesh** menu on the Model toolbar, as well as from the context menu that appears when you Alt+right-click (Ctrl+Alt+right-click on Linux) on a polygon mesh in a 3D view or right-click when polygon mesh components are selected.

You can:

- Draw and modify polygons interactively with the **Add/Edit Polygon Tool**.
- Draw edges interactively with the **Add Edge Tool**.
- Split edges and move points interactively with the **Add Vertex Tool**.
- Weld vertices interactively while moving them or weld selected pairs of vertices using **Weld Two Points**.
- Zip opposite boundaries or weld selected boundary edges using **Weld Boundary Points/Edges**.
- Split edges interactively with **Split Edge Tool**.
- Split edges with numerical accuracy using **Split Edge** (**with split control**).
- Split polygons interactively with **Split Polygon Tool**.
- Subdivide polygons and edges evenly with **Subdivide Polygons/Edges**.

Many of the tools described in this section use raycast mode when you are working in a shaded or textured display. If a tool appears not to be working, make sure that you are not trying to work on the back faces of polygons. You may need to orbit the camera, use a different 3D view, or switch to wireframe display.

Adding and Editing Polygons (Drawing)

The Add/Edit Polygon tool is a multi-purpose tool that lets you draw polygons interactively by placing vertices. You can use it to add polygons to an existing mesh, add or remove points on existing polygons, or to create a new polygon mesh object. There are two preferences that control how the Add/Edit Polygon tool works: **Preserve Planarity While Creating Polygons** and **Action When Connecting CW vs CCW Polygons**.

To draw polygons

1. Do one of the following:

   - To create a new polygon mesh object, first make sure that no polygon meshes are currently selected.

     or

   - To add polygons to an existing polygon mesh object, select the mesh first.

     or

- To add or remove points on an existing polygon in a existing polygon mesh object, select that polygon.

2. Choose **Modify** > **Poly. Mesh** > **Add/Edit Polygon Tool** from the Model toolbar or press **n**.

3. Do one of the following:

   - Click in a 3D view to add a point. If necessary, you can adjust the position by moving the mouse pointer before releasing the button.

   or

   - Click an existing point on another polygon in the same mesh to attach the current polygon to it.

   or

   - Click an existing edge of another polygon in the same mesh to attach the current polygon to it.

   or

   - Left-click and drag on a vertex of the current polygon to move it.

   or

   - Middle-click a vertex of the current polygon to remove it.

   As you move the mouse pointer, the edges that would be created are outlined in red. To insert the new point between a different pair of vertices of the current polygon, first move the mouse across the edge connecting them.

   The direction of the normals is determined by the direction in which you draw the vertices. If the vertices are drawn in a counterclockwise direction, the normals face toward the camera and if drawn clockwise, they face away from the camera. As you draw, red arrows indicate the order of the vertices.

   When you connect two polygons whose normals point in opposite directions, the behavior is determined by the **Action When Connecting CW vs CCW Polygons** preference as described in the next section.

   If desired, you can activate snapping while using this tool. For more about snapping, see *Chapter 18: Snapping* in the *Fundamentals* guide.

4. When you have finished drawing a polygon, do one of the following:

   - To start a new polygon and automatically share an edge with the current one, first move the mouse pointer across the desired edge and then click the middle mouse button. Repeat step 3 as necessary.

   or

   - To start a new polygon without sharing automatically sharing an edge, click the right mouse button. Repeat step 3 as necessary.

   or

- When you are finished drawing polygons, exit the Add/Edit Polygon tool by clicking the right mouse button twice in a row, by choosing a different tool, or by pressing Esc.

Preferences that Affect Drawing Polygons

There are two preferences you can set to control how the Add/Edit Polygon tool works: **Preserve Planarity While Creating Polygons** and **Action When Connecting CW vs CCW Polygons**.

To access these options, choose **File** > **Preferences** from the main menu, then expand **Tools** and click **Polygon**.

Preserve Planarity While Creating Polygons

The **Preserve Planarity While Creating Polygons** preference automatically adjusts the position to preserve the planarity of the polygon when you click to add a new vertex using the Add/Edit Polygon tool. As a result, new vertices may not be added exactly where you click—even if snapping is on.

If this option is off, then it is possible to draw non-planar polygons, for example, by drawing three vertices in the Front view and then a fourth in the Side view.

You can temporarily change the current behavior by pressing Shift while adding points. Pressing Shift allows you to force planarity as you draw a specific polygon when the preference is off, or allow a non-planar polygon when the preference is on.

For more information about polygons and planarity, see *Planar and Non-planar Polygons* on page 187.

Action When Connecting CW vs CCW Polygons

When you draw polygons and connect them to others, the normals might be facing in opposite directions. This occurs if the vertices on two polygons are ordered differently, that is, clockwise versus counterclockwise. When two such polygons are about to be connected, the **Action When Connecting CW vs CCW Polygons** preference controls what happens:

- **Prompt user before proceeding:** You are prompted to invert polygons or not in each case.
- **Invert conflicting polygon(s) (and all their adjacent ones):** The other polygons are inverted to fit with the current polygon you are drawing. The inversion ripples through adjacent polygons (if appropriate).
- **Accept CW-CCW connection (with double-edge side-effect):** No changes are made to the normal directions. Adjacent polygons with opposite normals are connected by a double edge to preserve the integrity of the mesh structure.

For more information about polygon normals, see *Polygon Normals and Double Edges* on page 275.

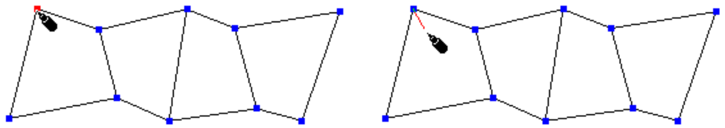
Adding Edges

You can use the Add Edge tool to split or cut polygons interactively by drawing new edges that connect existing edges or vertices.

To add edges

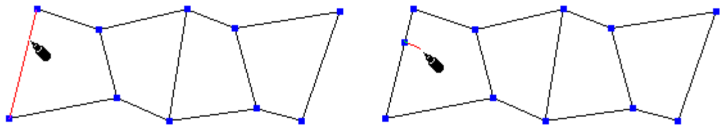
1. Select a polygon mesh object.
2. Choose **Modify** > **Poly. Mesh** > **Add Edge Tool** from the Model toolbar or press \ .
3. To specify the starting point for a new edge, do one of the following:

   - To start drawing an edge from an existing point, click on it.



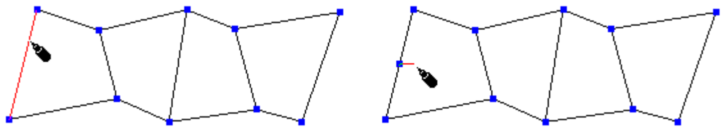
   or

   - To start drawing from an existing edge, click on the desired location along the edge. A new point is created.



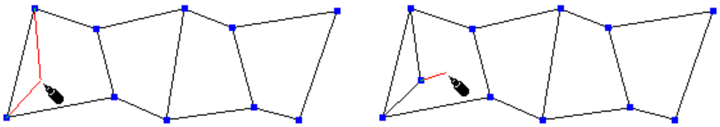
   or

   - To start drawing from the midpoint of an edge, press Ctrl while clicking anywhere on that edge. A point is created, evenly bisecting the edge.



   or

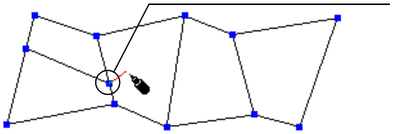
- To start drawing from a point in the interior of a polygon, press Alt while clicking inside the polygon. A point is added and a triangle is created by automatically adding edges connecting that point to the endpoints of the nearest edge.



On Linux, Alt+click may not work if that combination is used by your window manager for another purpose. In these cases, you can modify your window manager to use a different combination. For more information, see the documentation for your window manager.

4. To specify the endpoint of the new edge, do one of the following:

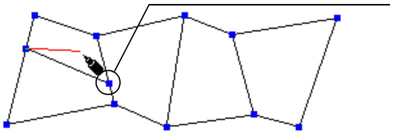
   - If you want the next edge to start from the endpoint of the edge you are currently drawing, click on another point or edge. A new edge is added connecting the two locations. You can use the Ctrl, Shift, or Ctrl+Shift keys while clicking to modify how the new edge is attached to an existing edge as described later in this step. You can repeat this step to draw a strip of edges.



Click to continue drawing edges from the last point.

or

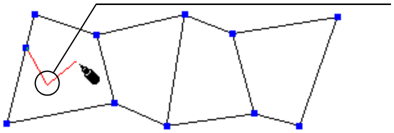
   - If you want the next edge to share the same starting point as the edge you are currently drawing, middle-click on a point or edge instead of left-clicking. Again, you can use the Ctrl, Shift, or Ctrl+Shift keys while middle-clicking to modify how the new edge is attached to an existing edge as described later in this step. You can repeat this step to quickly draw fans of polygons.



Middle-click to continue drawing edges from the previous point.

or

- Click in the interior of a polygon to add a point. You can repeat this step to add as many interior points as you like, creating a polyline, before terminating it. You terminate the polyline by attaching the endpoint to an existing edge or point of the polygon, using the left or middle mouse button as described above.



Click inside a polygon to add an interior point.

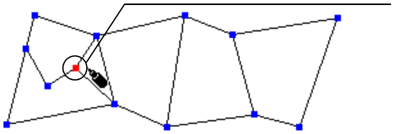
If you exit the Add Edge tool or start over at a new location (as described in step 5) before you attach the end of the polyline, the interior points are discarded because they do not form a valid mesh structure.

If you use snapping to align the interior points with the grid or other elements, the plane in which the interior points are created depends on the **Preserve Planarity During Creation** setting in your Polygon preferences. When that preference is on, then the new points snap in screen space but are still created in the plane of the existing polygon.

When that preference is off, the interior points are created at the exact position of the snapping target in 3D space. If you don't use snapping, interior points are always created in the plane of the existing polygon.

For more information about snapping in general, see *Chapter 18: Snapping* of the *Fundamentals* guide. For more information about the Polygon preferences, see *Preferences that Affect Drawing Polygons* on page 216.

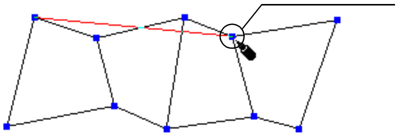
When adding interior points, you can press Alt while clicking to automatically create a triangle by connecting the new point to the endpoints of the closest edge, just as when specifying the starting point in the previous step.



Alt+click inside a polygon to add a point and connect it to the nearest edge by a triangle.

No matter which options you use to draw edges, you can press Ctrl+z one or more times to undo the last actions without exiting the Add Edge tool. You can also press Ctrl+y to redo any action you undid.

If you are trying to attach a new edge to an existing edge or vertex, and the target does not become highlighted when you move the pointer over it, it means that you cannot attach the new edge at that location because it would create an invalid mesh.

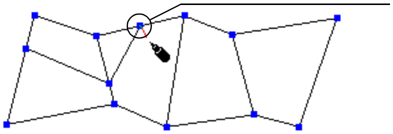


You cannot attach the edge to this point.

If you draw an edge between two points that already share an edge, no new edge is created—the second point simply becomes the starting point for the next edge you draw. If you click on an edge that is adjacent to the current point, a new vertex is created.

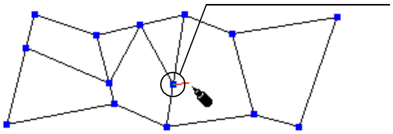
When you are attaching a new edge to an existing edge, you can use the Ctrl, Shift, and Ctrl+Shift keys as follows:

- Press Ctrl while clicking or middle-clicking an edge to bisect it evenly.



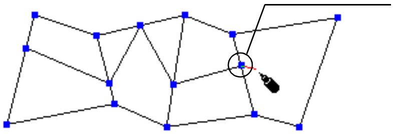
Press Ctrl to bisect an edge evenly.

- Press Shift while clicking or middle-clicking an edge to ensure that the angle between the new edge and the target edge snaps to multiples of the **Snap Increments - Rotate** value set in your Transform preferences. For example, if **Snap Increments - Rotate** is 15, then the new edge will snap at 15 degrees, 30 degrees, 45 degrees, and so on. Angles are calculated in screen space. For more information about snapping increments, see *Incremental Snapping* in Chapter 18 of the *Fundamentals* guide.



Press Shift to attach at multiples of the rotation increment. In this case, the increment is 15 and the edge is attached at 45 degrees.

- Press Ctrl+Shift while clicking or middle-clicking an edge to attach the new edge at a right angle to the target edge. The angle is calculated in object space.



Press Ctrl+Shift to attach at a right angle.

5. When you have finished drawing a strip or fan of edges, do one of the following:

   - To continue adding edges starting at a new location, right-click and then repeat steps 2 to 4.

     or

   - To exit the Add Edge tool, press Esc or choose a different tool.

Adding Vertices

The Add Vertex tool combines some of the capabilities of Split Edge and Move Point.

To add vertices

1. Select a polygon mesh object.
2. Choose **Modify** > **Poly. Mesh** > **Add Vertex Tool** from the Model toolbar or press Insert.
3. Do one of the following:

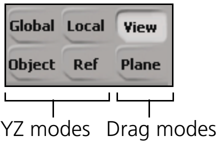
   - To add and move a vertex, click at the desired location along an edge and then drag the resulting point.

     or

   - To move a point, click and drag it.

   The three mouse buttons let you drag in different ways, depending on the current Transform mode.

   Transform Modes

   If desired, you can activate snapping while using this tool. For more information, see *Chapter 18: Snapping* in the *Fundamentals* guide.

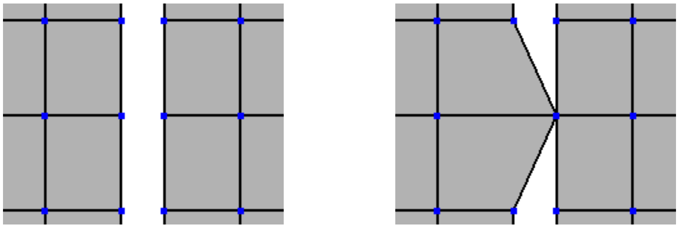
4. Repeat step 3 as desired.
5. To exit the Add Vertex tool, press Esc or choose another tool.

Welding Vertices

Welding merges two or more vertices on a polygon mesh into a single vertex.

- You can weld a single pair of points interactively while using the Move Point tool.
- You can also weld selected points either to targets or to their average position using a menu command.
- As an alternative to welding large numbers of points, you can filter points by distance (see *Filtering Points* on page 286).

You can weld points only if they belong to the same polygon mesh object.



To weld two points interactively

1. Activate the Move Point tool (press **m** or choose **Modify** > **Component** > **Move Point Tool** from the Model toolbar).
2. Press Alt while moving a point. The point snaps to other points as you move it.
3. When the point you are moving snaps to the desired target point, release the mouse button. The welded vertex is created at the target point's position.

   A WeldPoints operator is applied but its property editor is not opened automatically.

   Note that interactive welding uses the same snapping region size as the Snap tool. You can modify the region size using the Snap menu or Ctrl+ mouse-wheel.

To weld selected points to targets or to their average position

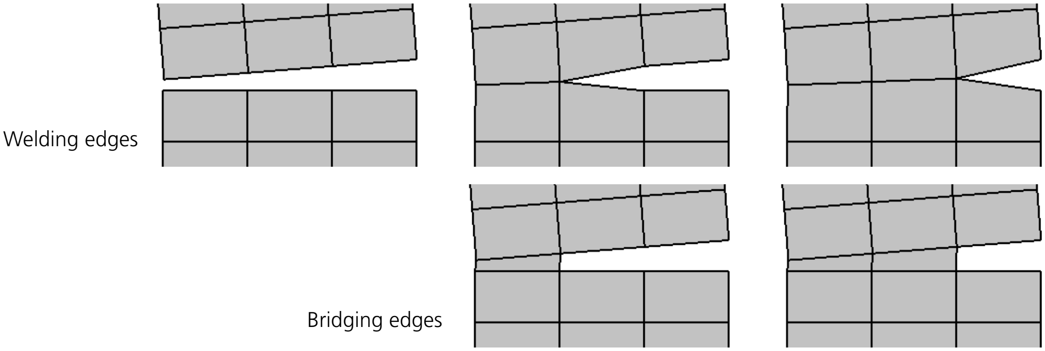
1. Select the points to weld.
2. Choose **Modify** > **Poly. Mesh** > **Weld Points to Target/Average** from the Model toolbar.
3. Pick a target vertex or vertex cluster:
   - If you pick a single vertex, all selected points are welded to it.
   - If you pick multiple vertices or a vertex cluster, each selected vertex is welded individually to the nearest target vertex.
   - If you right-click to end picking without picking a target, the selected points are welded together at their average position.

   A WeldPoints operator is applied but its property editor is not opened automatically.

Welding & Bridging Edges

Welding edges merges one or more pairs of facing boundaries on a single polygon mesh object. Bridging adds polygons between pairs of facing boundaries on a single polygon mesh object.

You can use these features to “zip” up opposite boundaries (if two edges are selected) or to restrict the effect to the edges you specify (if more than two edges are selected).



You can use **Weld Boundary Points/Edges** to clean up double edges that are not required to maintain the integrity of the mesh. Normally you should not encounter such edges, but they may occur in particular circumstances or when working with objects imported from other applications.

To weld or bridge selected edges

If you select more than two edges first, the welding or bridging operation is restricted to the selected edges. You can still use the Distance parameter to specify a tolerance for welding or bridging specific pairs of edges.

1. Select the boundary edges that you want to weld or bridge. You can specify how the selection is interpreted by setting **Target** in step 3.

   You can also specify edges by selecting all their connected points. Non-boundary edges are ignored. If you select a polygon mesh object, all boundaries are merged.

2. Choose **Modify** > **Poly. Mesh** > **Weld Boundary Points/Edges** or **Modify** > **Poly. Mesh** > **Bridge Boundary Points/Edges** from the Model toolbar.

3. The WeldEdges Op property editor opens and the **Grow Selection** option is not available. Adjust the options as desired:

   - **Distance** controls the tolerance between facing edges. As you increase this value, more pairs of edges get affected.

- **Target** specifies which edges to weld, when an appropriate pair of candidates is found: **Input Edges** welds pairs of selected edges with each other; **Other Edges** welds each selected edge with an unselected edge; and **All Edges** welds each selected edge with any other selected or unselected edge.

- **Bridge** controls the behavior for connecting edges.

  When this option is off, pairs of facing edges are merged. This option is off by default if you chose **Weld Boundary Points/Edges**,

  When this option is on, pairs of edges get connected by new polygons. This option is on by default if you chose **Bridge Boundary Edges**.

To zip opposite boundaries

You can use the **Grow Selection** option to "zip" up opposite boundaries only if you select no more than two edges first.

1. Select one edge on one of the boundaries you want to weld or bridge.

2. If desired, add the corresponding edge on the opposite boundary to the selection.

   If you do not specify the second edge, the closest opposite boundary will be used automatically.

3. Choose **Modify** > **Poly. Mesh** > **Weld Boundary Points/Edges** or **Modify** > **Poly. Mesh** > **Bridge Boundary Edges** from the Model toolbar.

4. The WeldEdges Op property editor opens and the **Grow Selection** option is available. Adjust the options as desired:

   - **Distance** controls the tolerance between facing edges. As you increase this value, more pairs of edges are affected.

   - **Bridge** controls the behavior for connecting edges.

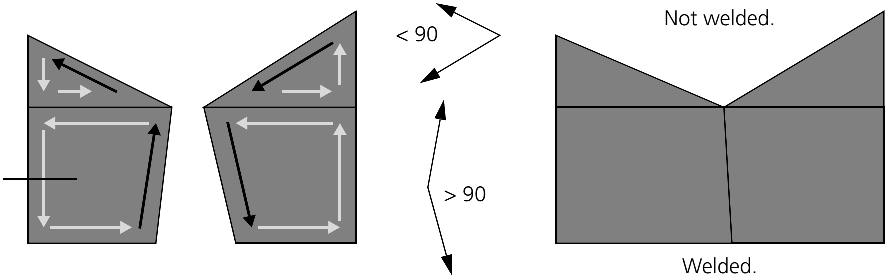
     When this option is off, pairs of facing edges are merged. This option is off by default if you chose **Weld Boundary Points/Edges**.

     When on, pairs of edges get connected by new polygons. This option is on by default if you chose **Bridge Boundary Edges**.

   - **Grow Selection** allows boundaries adjacent to the ones you picked to be welded as well. Turn this option off to weld only the two original boundaries.

- **Angle** > **90** welds two boundaries only if the difference between their orientations is greater than 90 degrees. This prevents the zipping effect from welding pairs of edges that are not facing each other.

Angle > 90



Edges are ordered counterclockwise when normals face you.

Splitting Edges

You can split edges interactively with the Split Edge tool or with numerical accuracy using the Split Edge (with split control) command. Both methods also allow you to automatically split parallel edges and join the new vertices. In addition, both methods add a SplitEdge operator to the object's stack.

To split edges interactively

The Split Edge tool lets you interactively add vertices along existing edges. You can also automatically split parallel edges and join the new vertices.

1. Select a polygon mesh object.
2. Choose **Modify** > **Poly. Mesh** > **Split Edge Tool** from the Model toolbar or press] .
3. Move the pointer over an existing edge. The targeted edge turns red.
4. Do one of the following:
   - To add a new vertex, click at the desired position along the targeted edge.

     or

   - To automatically add and join new vertices along parallel edges, click the middle mouse button. Note that this works best with a row of quadrilateral polygons.

   You can bisect the edge evenly by pressing Ctrl while clicking.

   You can adjust the position of the new vertex along the edge by moving the mouse before releasing the mouse button. However, unlike the Add Vertex tool described on page 221, you can move the new vertex only along the edge.

5. Repeat steps 3 and 4 as necessary.

6. When you have finished splitting edges, exit the Split Edge tool by right-clicking.

To split edges with numerical accuracy

The Split Edge command splits edges with numerical accuracy. You can also automatically split parallel edges and join the new vertices.

1. Select some edges on a polygon mesh object.
2. Choose **Modify** > **Poly. Mesh** > **Split Edge** (**with split control**) from the Model toolbar. The SplitEdge Op property editor opens.

To see the vertices created by the edge subdivision, press **t** to switch to Point mode.

3. Set the options as desired:

   - **Ratio Edge** determines where the new vertex is placed along an edge.
   - **Parallel Edge Loop** additionally splits edges parallel to those selected. Note that this works best with a row of quadrilateral polygons.
   - **Connect** adds edges connecting the newly created vertices. This has an effect only when Parallel Edge Loop is on.

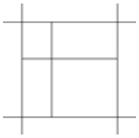
Splitting Polygons

The Split Polygon tool lets you interactively subdivide *n*-sided polygons into *n* new polygons. For example, a triangle splits into three new polygons, a quad splits into four new polygons, and so on.

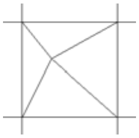
To split polygons

1. Select a polygon mesh object.
2. Choose **Modify** > **Poly. Mesh** > **Split Polygon Tool** from the Model toolbar.
3. Move the pointer over an existing polygon. The targeted polygon turns red.
4. Do one of the following:

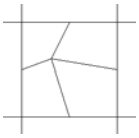
   - Left-click to split the targeted polygon by drawing edges from the point you clicked to each edge. In the case of quadrilateral polygons, two colinear pairs of edges are created, resulting in a cross shape.
   - Middle-click to draw new edges from the point you clicked to each vertex of the targeted polygon.
   - Right-click to draw new edges from the point you clicked to the midpoint of each edge of the targeted polygon.



Left-click



Middle-click



Right-click

   If desired, you can activate snapping while using this tool. For more about snapping see *Chapter 18: Snapping* in the *Fundamentals* guide.

5. Repeat steps 3 and 4 as necessary.

6. When you have finished splitting polygons, exit the Split Polygon tool by pressing Esc or choosing another tool.

Subdividing Polygons and Edges

You can subdivide polygons and edges evenly using the Subdivide Polygons/ Edges command. The options are different for each component type.

As an alternative to using this command to subdivide polygons, you can also apply local subdivision as described in *Subdividing Polygons Locally* on page 339.

To subdivide polygons evenly

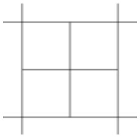
1. Select some polygons, polygon clusters, or polygon mesh objects.
2. Choose **Modify** > **Poly. Mesh** > **Subdivide Polygons/Edges** from the Model toolbar or press Shift+d.
3. In the SubdividePolygon Op property editor, set the options as desired:
   - **Subdivision Type** determines the shape of the new polygons:

     Plus adds edges from the geometric center of a polygon to the midpoint of each edge.

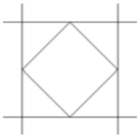
     Diamond draws edges connecting the midpoints of a polygon's edges.

     X draws edges from the midpoint of a polygon to each of its vertices.

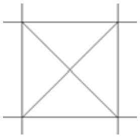
     Triangles triangulates all quad and n-sided polygons.



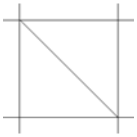
Plus



Diamond

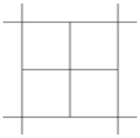


X

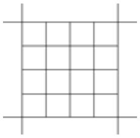


Triangles

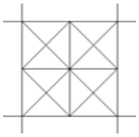
   - **Iterations** controls how many times the process is repeated on the polygons that are created at each step. This option has no effect when Subdivision Type is Triangles.



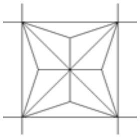
Iterations = 1



Iterations = 2



Type X, Tile on



Type X, Tile off

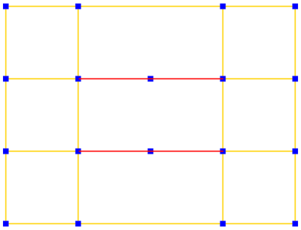
- **Tile** has a visible effect only when both Subdivision Type is Diamond or X and Iterations is greater than 1.

  When it is on, all but the last iteration is of the type Plus and the final subdivision is of the type specified. This gives a tile-like pattern when working with quadrilateral polygons.

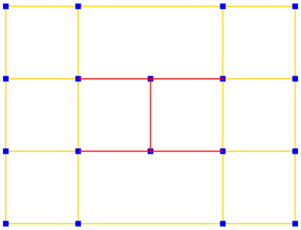
To triangulate a polygon mesh object while keeping its textures and other component attributes, subdivide all its polygons with Subdivision Type set to Triangles. See also *Triangulating Polygons* on page 305.

To subdivide edges evenly

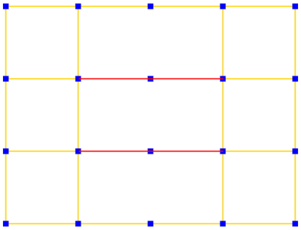
1. Select some edges on a polygon mesh object.
2. Choose **Modify** > **Poly. Mesh** > **Subdivide Polygons/Edges** from the Model toolbar.
3. In the SubdivideEdge Op property editor, set the options as desired:
   - **Subdivisions** determines how many new edges are created.
   - **Parallel Edge Loop** additionally splits edges parallel to those selected. Note that this works only with a row of quadrilateral polygons.
   - **Connect** adds edges connecting the newly created vertices.



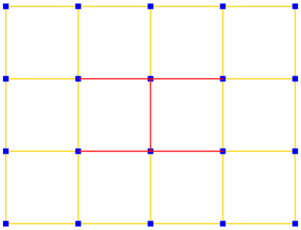
Parallel Edge Loop and Connect both off.



Connect on.



Parallel Edge Loop on.



Parallel Edge Loop and Connect both on.

To see the vertices created by the edge subdivision, press **t** to switch to Point mode.

Removing Polygon Mesh Components

There are several different ways to remove polygon mesh components:

- **Delete Components** removes the selected components and any other components that share them, leaving nothing in their place. See the next section, *Deleting Polygon Mesh Components*.
- **Collapse Components** deletes the selected components, then reconnects any other components that shared them. See *Collapsing Polygon Mesh Components* on page 231.
- **Dissolve Components** deletes the selected components and fills the resulting gap with new polygons. See *Dissolving Components* on page 232.
- **Dissolve & Clean Adjacent Vertices** dissolves the selected components (as described above) and also removes any points that are shared by only two edges as a result of dissolving components. See *Dissolving Components and Cleaning Vertices* on page 233.

All of these options are available from the **Modify** > **Poly. Mesh** menu on the Model toolbar, as well as from the context menu that appears when you right-click on an object in a 3D view while components are selected.

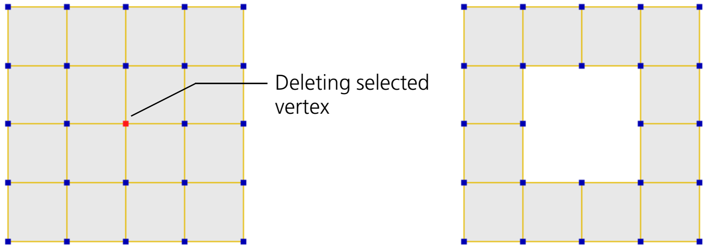
When components are selected, pressing Delete performs different actions:

- Points and edges are dissolved and adjacent vertices are cleaned.
- Polygons are deleted.

Deleting Polygon Mesh Components

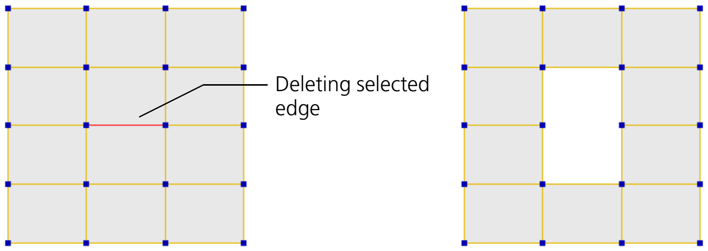
You can delete polygons, edges, and vertices using Alt+Delete. Deleting removes polygons, leaving empty holes.

- When vertices are deleted, any edges and polygons that share those vertices are also deleted (below).

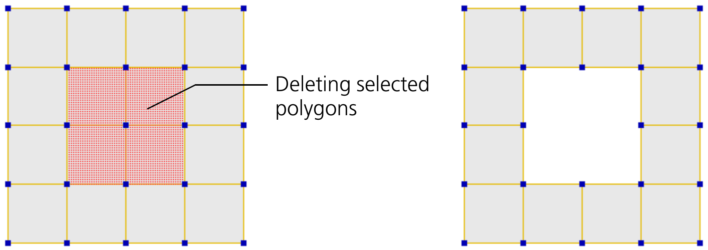


Deleting selected vertex

- When edges are deleted, any polygons that share those edges are also deleted (below).



- When polygons are deleted, any edges and vertices that are no longer attached to a polygon are also deleted (below).



To delete polygon mesh components

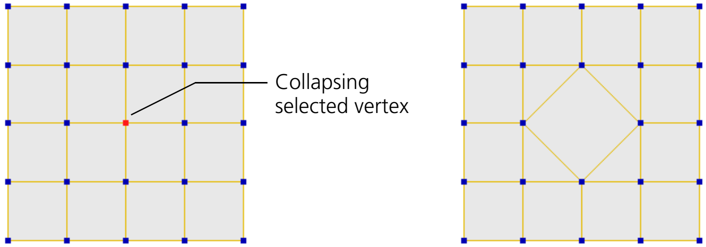
1. Select some vertices, edges, or polygons on a polygon mesh object.
2. Press Alt+Delete or choose **Modify** > **Poly. Mesh** > **Delete Components** from the Model toolbar.

   For polygons, you can also press Delete.

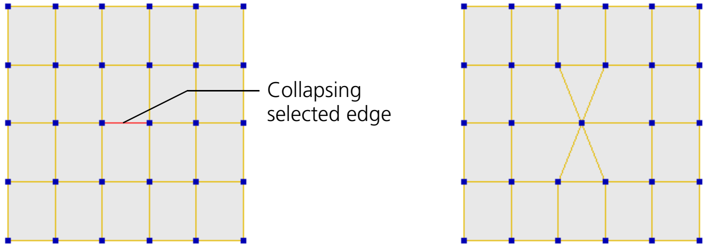
Collapsing Polygon Mesh Components

You can remove polygons, edges, and vertices by collapsing them using Ctrl+Delete. The components are removed and remaining components are reconnected, creating no new holes.

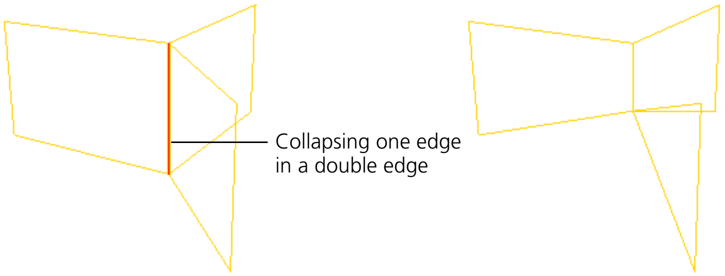
- When a vertex is collapsed, it is removed along with any edges that share it. The other vertices that were connected to those edges are joined in a loop by new edges (below).



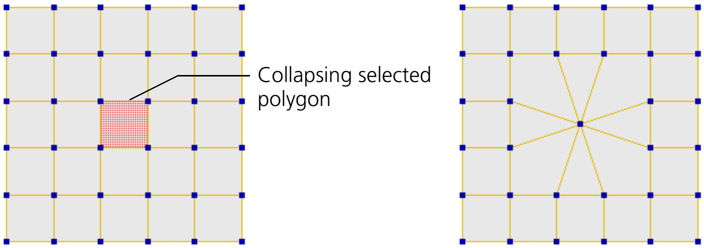
- When an edge is collapsed, it is removed along with its two endpoints. The other vertices that were connected to those edges are all joined to a new vertex at the midpoint of the collapsed edge. The effect is as if the original edge was scaled to zero (below).



If you collapse just one edge of a double edge, the selected edge is removed. The vertex at the end of one of the "dangling" edges is removed and that edge is reconnected to the other "dangling" vertex, which is moved to the midpoint. This affects other polygons adjacent to the double edge.



- When a polygon is collapsed, it is removed along with its edges and vertices. The other vertices that were connected to those edges are all joined to a new vertex at the midpoint of the collapsed polygon. Again, the effect is as if the original polygon was scaled to zero (below).



When multiply selected polygons are collapsed, connected polygons are collapsed together as if they were a single polygon. Disconnected islands of selected polygons are collapsed separately.

To collapse polygon mesh components

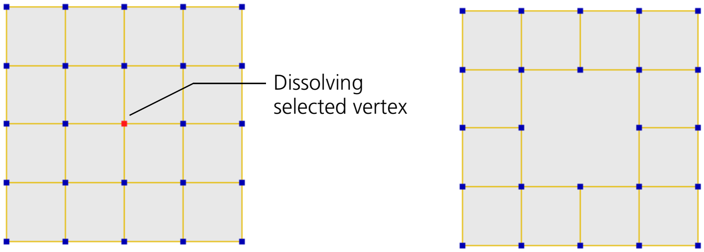
1. Select some vertices, edges, or polygons on a polygon mesh object.
2. Press Ctrl+Delete or choose **Modify** > **Poly. Mesh** > **Collapse Components** from the Model toolbar.

Dissolving Components

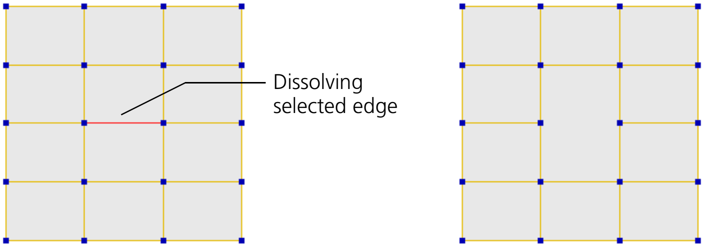
You can dissolve components using Ctrl+;. When you dissolve components, they are deleted and the resulting holes are filled with new polygons.

You have the option of automatically cleaning vertices when dissolving. See *Dissolving Components and Cleaning Vertices* on page 233.

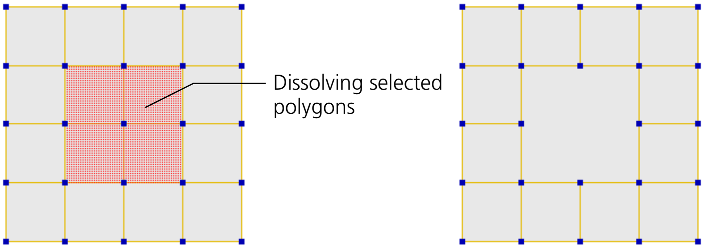
- When vertices are dissolved, any edges and polygons that share those vertices are deleted and the resulting holes are filled (below).



- When edges are dissolved, any polygons that share those edges are deleted and the resulting holes are filled (below).



- When polygons are dissolved, any edges and vertices that are no longer attached to a polygon are deleted and the resulting holes are filled (below).



- If you dissolve a single polygon, it has no visible effect.
- You cannot dissolve boundary edges.
- In versions previous to XSI 3.0, colinear points were also removed after dissolving.

  Since you can now explicitly choose to clean vertices when dissolving (see *Dissolving Components and Cleaning Vertices* on page 233), colinear points are no longer removed automatically.

To dissolve polygon mesh components

1. Select some vertices, edges, or polygons on a polygon mesh object.
2. Choose **Modify** > **Poly. Mesh** > **Dissolve Components** from the Model toolbar.

Dissolving Components and Cleaning Vertices

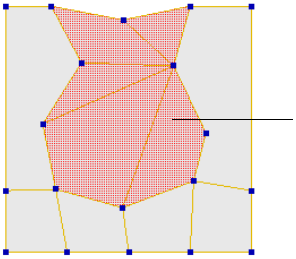
You can dissolve components and clean their adjacent vertices using the **;** (semicolon) key. In addition to removing the selected components and filling the holes with new polygons, this also performs the following actions:

- If a vertex was shared by three or more edges before dissolving, but shared by only two edges after dissolving, the vertex is then collapsed.

- Vertices that were already shared by only two edges before dissolving are not collapsed.
- Vertices that are still shared by three or more edges after dissolving are not collapsed either.

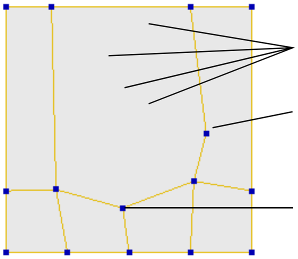
The following shows the effect of dissolving polygons and cleaning vertices:

Before



Selected polygons will be dissolved.

After
Dissolving and cleaning vertices

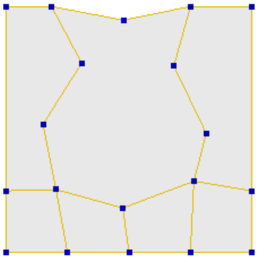


Vertices shared by two edges after dissolving are collapsed.

Vertices already shared by two edges are not collapsed.

Vertices shared by three or more edges are not collapsed.

After
Dissolving only



If you don't want to clean vertices, you can simply dissolve components. See *Dissolving Components* on page 232.

To dissolve polygon mesh components and clean vertices

1. Select some vertices, edges, or polygons on a polygon mesh object.
2. Press **;** (semicolon) or choose **Modify** > **Poly. Mesh** > **Dissolve & Clean Adjacent Vertices** from the Model toolbar.

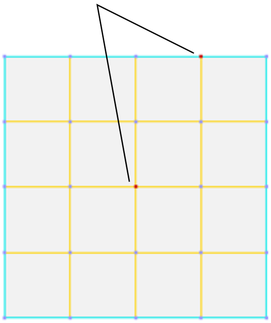
   For edges and vertices, you can also press Delete.

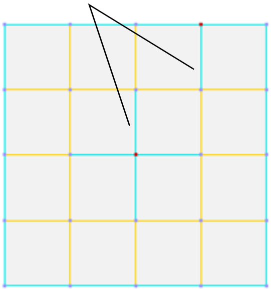
Disconnecting Polygon Mesh Components

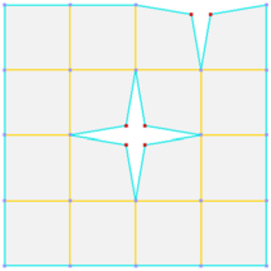
Disconnecting components is the opposite of welding. Components are separated from their neighbors, and new boundary edges and points are added as necessary. For example, you can select a set of polygons on a box and disconnect them to create a lid that is a separate “island” of polygons but still part of the box object.

You can disconnect points, edges, or polygons:

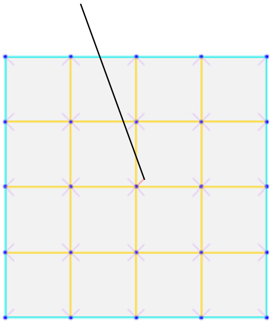
- When you disconnect a point, a new point is created for each polygon connected to it, as well as two new boundaries for each internal edge connected to it.

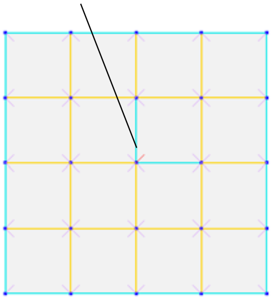

Points selected before disconnecting.

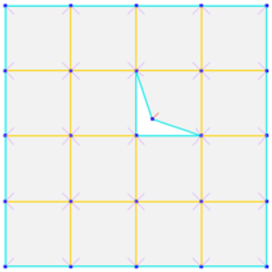

Disconnecting creates new boundaries and points. Initially, the new components are superimposed.


Here, the superimposed components have been moved slightly to see effect better.

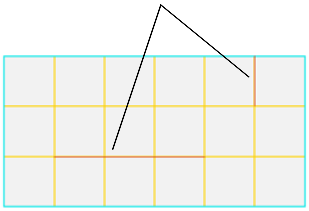
If you select samples instead of points, you can control how the point is disconnected.

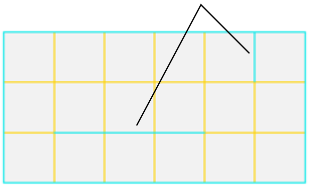

Only one of the four samples is selected.

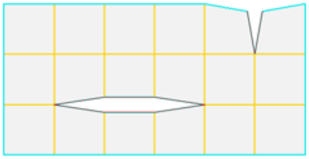

Disconnecting creates new boundaries and points. Again, the new components are superimposed.


Here, the superimposed components have been moved slightly to see effect better.

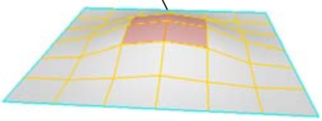
- When you disconnect an edge, two new boundaries are created, together with any corresponding points.

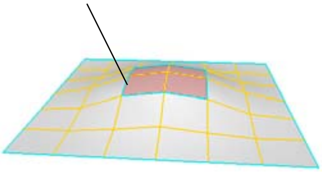

Edges selected before disconnecting.

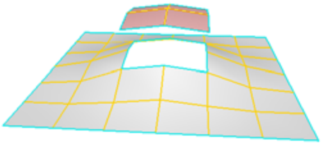

Disconnecting creates new boundaries and points. Initially, the new components are superimposed.


Here again, the superimposed components have been moved slightly to see effect better.

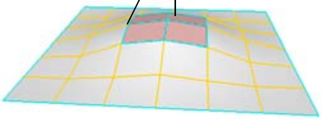
- When you disconnect a selection of polygons, boundaries are created along the outline of the selection.

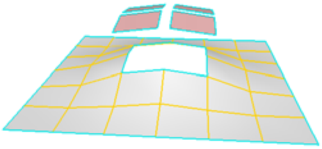

Polygons selected before disconnecting.


Disconnecting creates boundaries around the outline of the selection.

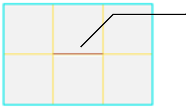

Again, the disconnected polygons have been moved slightly to see effect better.

With polygons, you also have the option of exploding them: this disconnects each polygon individually.


Exploding creates boundaries around each individual polygon.


Again, the exploded polygons have been moved slightly to see effect better.

- If you want the separate polygons to be part of a new object, you should use **Create** > **Poly. Mesh** > **Extract Polygons (delete)** instead of disconnecting them. See *Extracting Polygons* on page 322.
- Disconnecting a single interior edge has no effect.


You cannot disconnect this edge by itself.

To disconnect polygon mesh components

1. Select one or more points, edges, polygons, or clusters on a polygon mesh object.

2. Choose **Modify** > **Poly. Mesh** > **Disconnect Components** on the Model toolbar.

   If you did not select any components in step 1, you are prompted to pick them now.

   The picking session defaults to Rectangle Point selection, but you can change the filter using the buttons on the Select panel and you can change the tool using the commands on the **Select** > **Tools** menu or the keyboard equivalent (for example, F10 for the Raycast tool).

   The selected components are disconnected.

3. In the case of polygons, the Disconnect Component Op property editor opens.

   If desired, turn **Explode Polygons** on.

Extruding, Duplicating, and Insetting Polygon Mesh Components

Duplicating, extruding, and insetting are operations that copy selected polygons, edges, or points. This lets you create local details, such as indentations or protuberances like limbs and tentacles.

Two versions of the Extrude operator are responsible for all the various extruding, duplicating, and insetting effects:

- One version of the operator copies components along an axis. For more information about applying this operator, see *Insetting, Duplicating, and Extruding Components Along an Axis* on page 239.
- The other version of the Extrude operator copies components along a curve. For more information about applying this operator, see *Duplicating and Extruding Components Along a Curve* on page 242.

Different commands, on the **Modify** > **Poly. Mesh** menu of the Model toolbar as well as the **Edit** menu of the Edit panel, apply different presets of these operators to achieve different effects. (Of course, you can always adjust the parameters to modify the effects further or even change them completely.)

- For a description of the parameters that are specific to copying components along an axis, see *Parameters Specific to Copying Along an Axis* on page 241.
- For a description of the parameters that are specific to copying components along a curve, see *Parameters Specific to Copying Along a Curve* on page 243.
- For a description of the parameters that are common to both versions of the operator, see *Common Parameters* on page 244.

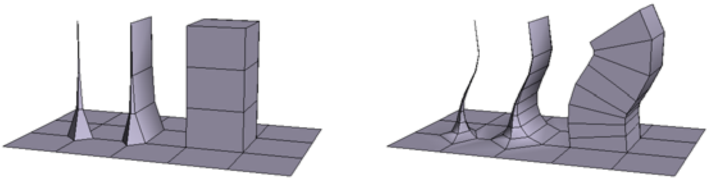
What's the Difference Between Extruding, Duplicating, and Insetting?

Extruding, duplicating, and insetting all involve copying components. They differ in how the components are copied, as well as how they are modified after copying.

Extruding Components

Extruding copies the selected components and transforms the new geometry. The new components are connected to the original selection.

Extruding a point, an edge, and a polygon...



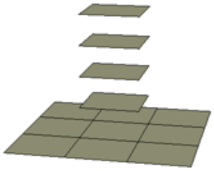
along an axis...

and along a curve.

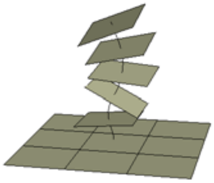
Duplicating Polygons

Like extruding, duplicating copies the selected components and optionally transforms the new geometry. However, the new components are not connected to the original selection. In addition, duplication applies only to polygons, and not to points or edges—you can apply the command to points and edges but the effect is the same as extruding.

Duplicating a polygon...



along an axis...



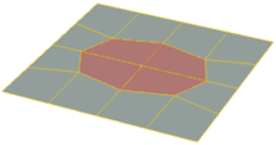
and along a curve.

Insetting Polygons

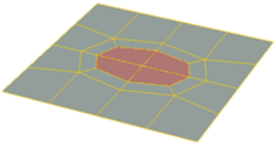
Insetting copies the selected polygons and then shrinks the contours of the new geometry. The boundary edges of the resulting polygons always remain parallel to the originals. The effect is as if you extruded the polygons by a length of 0, and then offset the result by a negative value (see *Offsetting Polygon Contours* on page 249).

Like duplication, insetting applies only to polygons, and not to points or edges—you can apply the command to points and edges but the effect is the same as extruding (by a length of 0).

Insetting polygons



Original selected polygons.



Results after insetting.

Insetting, Duplicating, and Extruding Components Along an Axis

The Extrude (Along Axis) operator is used to extrude, duplicate, and inset polygon mesh components. There are several ways to apply this operator with different presets as default values.

Copying Components Quickly Using a Hotkey

You can quickly copy polygons, edges, and points on polygon mesh objects using Ctrl+d (no subdivisions) or Ctrl+Shift+d (multiple subdivisions). While these key combinations are used for duplication when objects are selected, they apply the Extrude (Along Axis) operator when polygon mesh components are selected. In the case of selected polygons, you can choose whether to extrude or duplicate by setting a preference in Duplicate Options.

To copy components with no subdivisions

1. Select the polygons, edges, points, or clusters to copy.
2. Press Ctrl+d. Alternatively, this command is also available from **Edit** > **Duplicate/Instantiate** > **Duplicate Single**.

   The components are immediately extruded or duplicated, but the new components are initially superimposed on the originals. The new components are automatically selected so that you can scale, rotate, and translate them interactively using the transformation tools.

   Although the Extrude Op property editor does not open automatically, you can still display it and change values through the polygon mesh's operator stack as described in *Operator Stack* on page 51.

To copy components with multiple subdivisions

1. Select the polygons, edges, points, or clusters to copy.
2. Press Ctrl+Shift+d. Alternatively, this command is also available by choosing **Edit** > **Duplicate/Instantiate** > **Duplicate Multiple**.

   The components are immediately extruded or duplicated with 5 subdivisions along their local Y axis with a length of 1.

   Although the Extrude Op property editor does not open automatically, you can still display it and change values through the polygon mesh's operator stack as described in *Operator Stack* on page 51.

To set the behavior when copying polygons

1. Open the Duplicate Options property editor by choosing **Edit** > **Duplicate/Instantiate Options**.

   This property editor is also available under Commands in your preferences. For more information about preferences in general, see *Chapter 26: User Preferences* in the *Fundamentals* guide.

2. Set the desired value for **Duplicated Items - Polygons**. You can choose either **Duplicate** or **Extrude**.

Duplicating Polygons

1. Select the polygons, polygon clusters, or polygon mesh objects to duplicate.
2. Choose **Modify** > **Poly. Mesh** > **Duplicate Polygons** from the Model toolbar.

   The polygons are immediately duplicated, but the new components are initially superimposed on the originals. The new polygons are automatically selected so that you can scale, rotate, and translate them interactively using the transformation tools.

Although the Extrude Op property editor does not open automatically, you can still display it and change values through the polygon mesh's operator stack as described in *Operator Stack* on page 51.

Extruding Components Along an Axis

1. Select the polygons, edges, points, clusters, or polygon mesh objects to extrude.
2. Choose **Modify** > **Poly. Mesh** > **Extrude Along Axis** from the Model toolbar.

   The components are extruded with the default values and the Extrude Op property editor opens.
3. Set the values as desired:
   - For a description of the options that are specific to Extrude Op (Along Axis), including **Length**, **Frame**, and **Maintain Orthogonality**, see *Parameters Specific to Copying Along an Axis* on page 241.
   - For a description of the options that are common to both Extrude Op (Along Axis) and Extrude Op (Along Curve), including **Duplicate**, **Inset**, **Merge**, **Skirting Ratio**, **Subdivs**, and **Transform**, see *Common Parameters* on page 244.

Insetting Polygons

1. Select the polygons or polygon clusters to inset.
2. Choose **Modify** > **Poly. Mesh** > **Inset Polygons** from the Model toolbar.

   The polygons are inset with the default values and the Extrude Op property editor opens.
3. Set the values as desired:
   - For a description of the options that are specific to Extrude Op (Along Axis), including **Length**, **Frame**, and **Maintain Orthogonality**, see *Parameters Specific to Copying Along an Axis* on page 241.
   - For a description of the options that are common to both Extrude Op (Along Axis) and Extrude Op (Along Curve), including **Duplicate**, **Inset**, **Merge**, **Skirting Ratio**, **Subdivs**, and **Transform**, see *Common Parameters* on page 244.

Parameters Specific to Copying Along an Axis

There are several parameters that are specific to Extrude Op (Along Axis) and are not found in Extrude Op (Along Curve):

- **Length** is the length of the extrusion in Softimage units.

  If you are defining shapes in local shape mode, it is better to adjust this parameter than to translate the components interactively. Otherwise, you might not get the expected results as the local reference frame changes.

- The **X**, **Y**, and **Z** options specify the axis of extrusion. You can activate multiple axes to extrude along the combined unit vector.
- **Frame** specifies the coordinate system for the axis of extrusion, as well as for the parameters on the Transform tab:
  - **Object** uses the center of the parent polygon mesh object.
  - **Component** uses the local system of the selected components.
- **Maintain Orthogonality** ensures that the Y axis of the local reference frame is orthogonal to the selected components even when the object has been scaled non-uniformly. Only the existing transformations when the extrusion is applied are considered; if the object is animated or transformed later, the new values are not considered. This option is available only when **Frame** is set to **Component**.

Duplicating and Extruding Components Along a Curve

The Extrude (Along Curve) operator is used to extrude and duplicate inset polygon mesh components using a curve as the path. There are a couple of ways to apply this operator with different presets as default values.

To duplicate or extrude polygon mesh components along a curve

1. Select the polygons, edges, points, clusters, or a polygon mesh objects to extrude.
2. Choose one of the following commands from the Model toolbar:
   - **Modify** > **Poly. Mesh** > **Duplicate Polygons Along Curve**. This works with polygons, polygon clusters, and polygon mesh objects only. Points and edges are extruded instead of duplicated.

     or

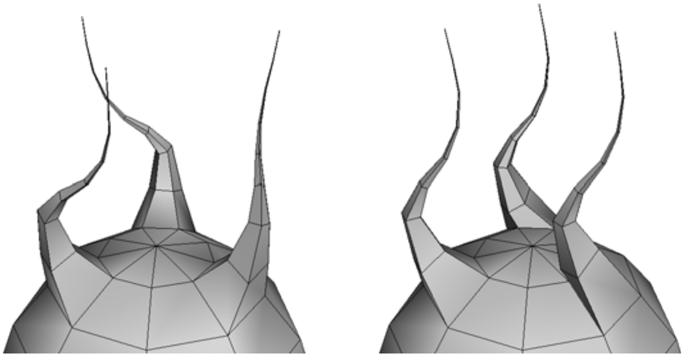
   - **Modify** > **Poly. Mesh** > **Extrude Along Curve**. This works with any type of component.
3. Pick a curve object.

   The components are duplicated or extruded with the default values and the Extrude Op property editor opens.
4. Set the values as desired:
   - For a description of the options that are specific to Extrude Op (Along Axis), including **Start**, **End**, **Autorotate**, and **Perpendicular**, see *Parameters Specific to Copying Along a Curve* on page 243.
   - For a description of the options that are common to both Extrude Op (Along Axis) and Extrude Op (Along Curve), including **Duplicate**, **Inset**, **Merge**, **Skirting Ratio**, **Subdivs**, and **Transform**, see *Common Parameters* on page 244.

Parameters Specific to Copying Along a Curve

There are several parameters that are specific to Extrude Op (Along Curve) and are not found in Extrude Op (Along Axis):

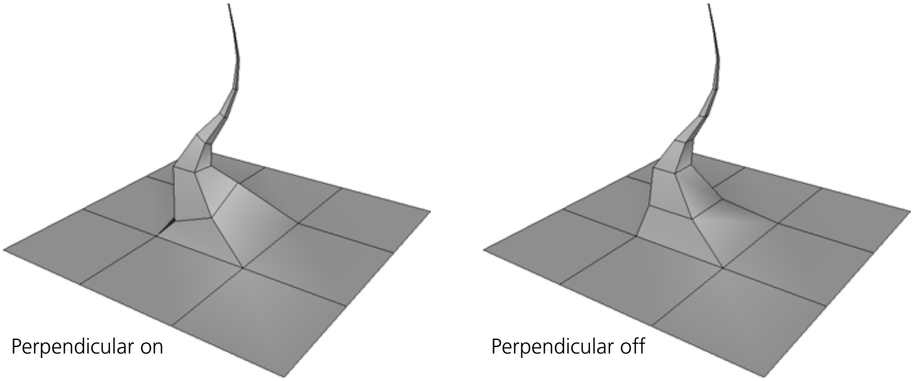
- **Start** and **End** control the offset of the extrusion from either end of the curve, as a percentage of curve length.
- **Autorotate** rotates the curve of extrusion so that the tangent at its start is aligned with the normal of each selected component.



Autorotate on

Autorotate off

- **Perpendicular** rotates the extruded components so that they are perpendicular to the curve of extrusion at each subdivision.



Perpendicular on

Perpendicular off

Common Parameters

The Extrude Op (Along Axis) and Extrude Op (Along Curve) property editors have several parameters in common: **Duplicate Polygons**, **Inset**, **Merge**, **Skirting Ratio**, **Subdivs**, and the **Transform** settings.

Duplicate Polygons

Duplicate Polygons is a toggle that controls the behavior of the Extrude operator with polygons. Different commands apply presets with different default values for this option: when it is off, the new polygons are connected (extruded) and when it is on, the new polygons are disconnected (duplicated).

See *What's the Difference Between Extruding, Duplicating, and Insetting?* on page 238 for more details and illustrations.

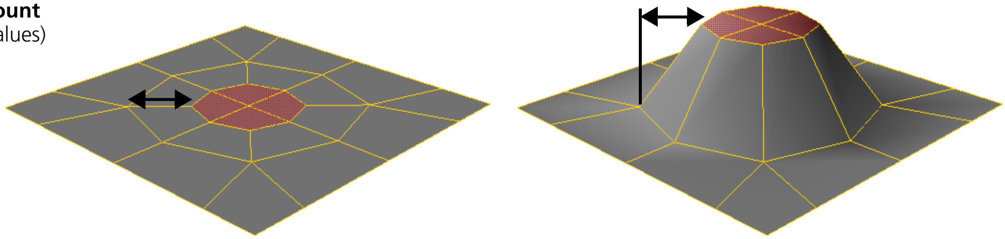
Inset

The Inset options allow you to shrink the boundary of the new polygons. You can achieve similar effects by scaling using the options on the Transform tab, but with Inset the new edges are always parallel to the originals.

In addition, scaling does not always work well with concave polygons whose edges might self-intersect. Another difference is that the inset always applies to the entire extrusion, rather than being repeated per subdivision.

To control the inset amount

- Adjust **Inset Amount** in the Extrude Op property editor. Values are in Softimage units.

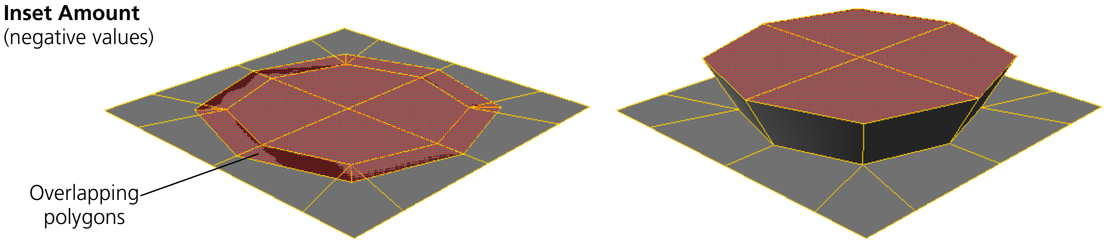


Inset Amount
(positive values)

Length = 0

Length > 0

Negative values expand the boundary instead of shrinking it. If the extrusion length is 0, the resulting polygons will overlap. If you do not want overlapping polygons, you should offset the polygons (see *Offsetting Polygon Contours* on page 249) before insetting them.



Length = 0

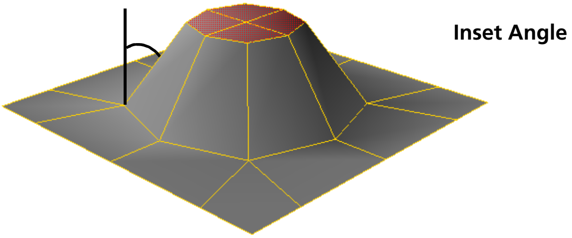
Length > 0

When polygons have extreme concavities, setting high negative values will produce spikes.

To control the inset angle (along axis only)

1. Make sure that **Length** is not 0.
2. Activate **Inset Locked**.

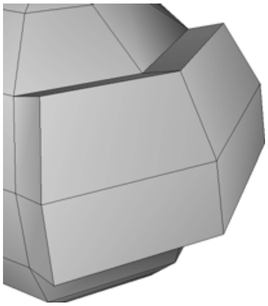
   Inset Amount becomes inactive and **Angle** becomes available.
3. Adjust **Angle**. Values are in degrees.



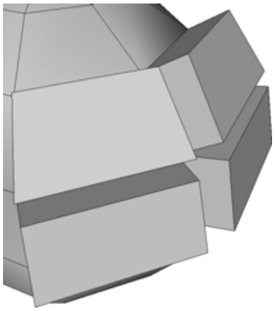
Merge

When you extrude adjacent components, the **Merge** option in the Extrude Op property editor controls whether they are extruded as one or separately.

When you are extruding or duplicating along an axis, there are only two options: on or off.



Merge on
Connected polygons extruded along average component axis.



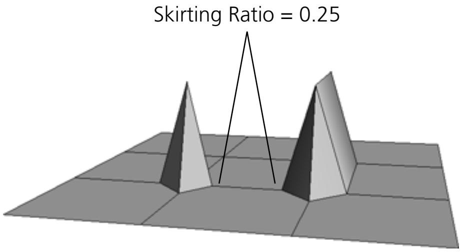
Merge off
Polygons extruded separately along individual component axes.

When you are extruding or duplicating along a curve, there are three options:

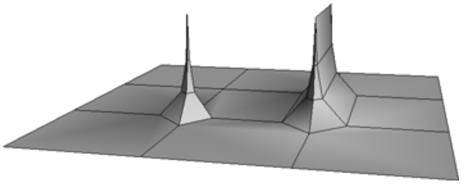
- **None** (**per polygon**) treats each polygon individually. It is the same as turning Merge off as shown above.
- **Adjacent Polygons** treats each group of connected polygons in the selection together, and treats disconnected groups separately.
- **All** treats all polygons together, whether or not they are connected. This is useful when you want to duplicate an entire object that is composed of disconnected islands of polygons. For example, you can duplicate text along a curve and keep the whole text aligned at each subdivision, rather than have each separate letter oriented differently.

Skirting Ratio

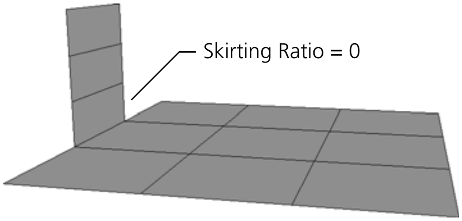
When you extrude edges and points, the **Skirting Ratio** parameter in the Extrude Op property editor controls the relative size of the "base."



When there are multiple subdivisions, the Skirting Ratio applies at each subdivision.



If the Skirting Ratio is 0.05 or less, then the components are extruded without a base. For example, this lets you extend boundary edges.



Subdivs

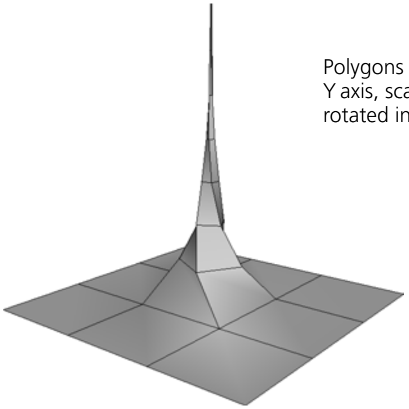
The **Subdivs** parameter in the Extrude Op property editor controls the number of subdivisions along the axis or curve of extrusion.

When you extrude along a curve, the **Per Knot** option controls whether the number of subdivisions is applied for every knot segment in the curve of extrusion or along the curve as a whole.

When this option is on, you can control the distribution of subdivisions in the extrusion by adding, removing, or moving points in the curve.

Transform

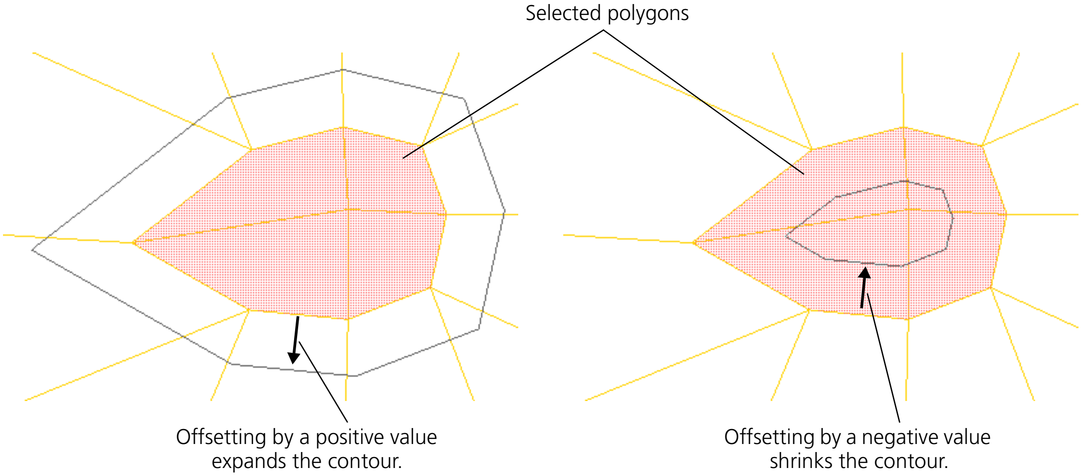
The options on the Transform tab of the Extrude Op property editor let you specify scaling, rotation, and translation values to be applied to the extrusion. The **Transform Per Subdivision** option applies the transformation values at each subdivision of the extrusion; if this option is off, the transformations apply to the extrusion as a whole.



Polygons extruded along the Y axis, scaled in X and Z and rotated in Y per subdivision.

Offsetting Polygon Contours

Offsetting contours expands or shrinks the outline of selected polygons. Each contour edge is shifted by a specified amount, and remains parallel to its original direction in the same plane as its polygon.


Selected polygons
Offsetting by a positive value expands the contour.
Offsetting by a negative value shrinks the contour.

Unlike **Inset Polygons** (which is a special case of duplicating or extruding polygons), **Offset Polygon Contours** does not modify the topology. In this respect, it is like a deformation that applies only to polygons.

To offset polygon contours

1. Select one or more polygons or polygon clusters.
2. Choose **Modify** > **Poly. Mesh** > **Offset Polygons Contour** from the Model toolbar.

   If you did not select any polygons or clusters in step 1, you are prompted to pick them now.

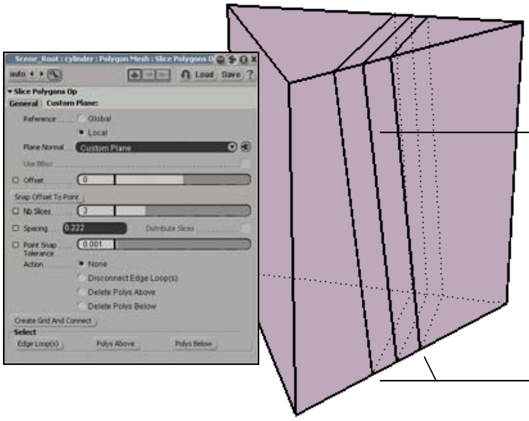
   The picking session defaults to Rectangle Polygon selection, but you can change the selection tool using the commands on the **Select** > **Tools** menu or the keyboard equivalent (for example, F10 for the Raycast tool).

   The outline of the selection is offset and the Offset Polygons Op property editor opens.
3. Adjust the **Offset** as desired. Values are in Softimage units. Positive values expand the outline and negative values shrink the outline. A value of 0 has no effect.

Knifing and Slicing Polygons

Knifing and slicing allow you to split the polygons of a mesh object along any plane. Both techniques add a Slice Polygons Op node to the construction history, but the workflow is different:

- **Modify** > **Poly. Mesh** > **Slice Polygons** applies the operator and opens its property editor so that you can modify values. This command is useful if you need to specify many settings: plane, offset, number of slices, spacing, and so on. See *Knifing Polygons* on page 251.



Scene_Root : cylinder : Polygon Mesh : Slice Polygons O

Slice Polygons Op

General | Custom Plane:

Reference ○ Global ● Local

Plane Normal Custom Plane

Use Bbox

□ Offset 0

□ Nb Slices 3

□ Spacing 0.222 Distribute Slices

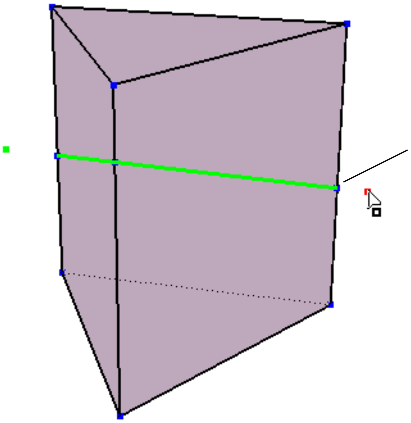
□ Point Snap Tolerance 0.001

Action ● None ○ Disconnect Edge Loop(s) ○ Delete Polys Above ○ Delete Polys Below

Slice lets you cut polygons along any plane.

You can specify the number of slices, spacing, offset, and so on.

- **Modify** > **Poly. Mesh** > **Knife Tool** is an interactive tool that lets you slice polygons by drawing a line in a 3D view. The operator is applied but its property editor does not open automatically. This tool is useful when you want to quickly and simply cut polygons along a custom plane. See *Slicing Polygons* on page 252.



Use the Knife tool to draw a line that slices polygons perpendicularly to the viewing plane.

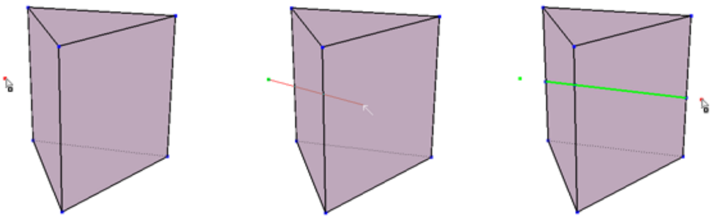
If you want to subdivide polygons along multiple orthogonal planes, you can dice them instead. See *Dicing Polygons* on page 258.

You can slice polygons to achieve the same results as the **Effect > Plane Clipping** command in the Model module of SOFTIMAGE|3D. In addition, slicing gives you extra options and control.

Knifing Polygons

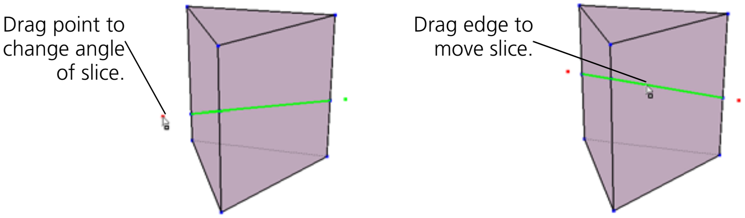
To knife polygons

1. Do one of the following:
   - To cut an entire object, select the object.
   - To cut specific polygons or polygon clusters on the object, select them first.
2. Choose **Modify** > **Poly. Mesh** > **Knife Tool** from the Model toolbar, or press [.
3. Click in a 3D view to define one endpoint of a line, and then click in a second location to define the other endpoint.



   The polygons are split along an infinite plane that contains the line and is perpendicular to the viewing plane.

4. Before you start another cut or exit the Knife tool, you can adjust the cutting plane if desired:



   - Move the mouse pointer over either one of the two endpoints you drew. When the mouse pointer changes from Knife (an arrow) to Move Point (an arrowhead and box as shown above) and the point is highlighted red, you can click and drag the point to a new location. This lets you adjust the angle of the slice. You can do this only in the same view you drew the endpoints, and only if the cutting plane is still perpendicular to the viewing plane (for example, if you have not orbited in supra mode).

- Move the pointer over any of the newly created edges (highlighted in green). When the mouse pointer changes from Knife to Move Point and the both points are highlighted red, you can click and drag the edges to a new location. This lets you adjust the position of the slice. You can do this in any 3D view.

Clicking anywhere else will start a new cut.

You can also display a context menu by pressing and holding the right mouse button (right-clicking quickly terminates the Knife tool). The context menu contains the following items:

- **Create Grid and Connect** creates an implicit grid that you can transform to manipulate the cutting plane. See *Specifying the Cutting Plane* on page 253.
- **Inspect Operator** displays the property editor for the Slice Polygons operator. See *Slicing Options* on page 253.

5. The Knife tool remains active, so you can repeat steps 3 and 4 to make more cuts if desired.

   To exit the Knife tool, right-click or press Esc. Alternatively, simply activate another tool.

   To make adjustments to the operation after you have exited the Knife tool, open the Slice Polygons Op property editor in the object's modeling stack. For more details, see *Slicing Options* on page 253.

> If you already used Knife on a particular object and reactivate the tool, it remembers the last cut you drew if it is the last topological change and the cutting plane is still perpendicular to the viewing plane. You can adjust the last cut as described in step 4 before performing a new cut.

Slicing Polygons

To slice polygons (apply Slice operator manually)

1. Do one of the following:
   - To cut an entire object, select the object.
   - To cut specific polygons or polygon clusters on the object, select them first.
2. Choose **Modify** > **Poly. Mesh** > **Slice Polygons** from the Model toolbar.

   The polygons are sliced with the default options and the Slice Polygons Op property editor opens. For information about setting the various options, such as specifying the cutting plane, see the sections that follow.

Slicing Options

The Slice Polygons Op property editor allows you to control various options, such as the position of the cutting plane and the number of slices.

Displaying Slice Options

The Slice Polygons Op property editor opens automatically when you apply the operator manually using **Modify** > **Poly. Mesh** > **Slice Polygons** from the Model toolbar.

If you are slicing interactively and the Knife tool is still active, you can open the property editor for the most recent operation by pressing and holding the right mouse button, then choosing **Inspect Operator** from the context menu.

Of course as with any modeling operator, you can always redisplay the property editor later by clicking on its icon in an explorer. See *Viewing and Modifying Operators* on page 51.

Specifying the Cutting Plane

The cutting plane determines the location and angle of the slice. You can:

- Use the XY, YZ, or XZ plane of the object's local or global reference frame.
- Specify a custom plane by entering position and rotation values numerically.
- Specify a custom plane interactively by connecting the operator to a 3D object and then translating and rotating it.

To specify a local or global plane

1. On the General tab of the Slice Polygons Op property editor, specify a **Reference** frame:

   - **Global** is relative to the scene origin.
   - **Local** is relative to the object's own center.

2. Specify the **Plane Normal**:

   - **X Axis** cuts along the YZ plane (X = 0).
   - **Y Axis** cuts along the XZ plane (Y = 0).
   - **Z Axis** cuts along the XY plane (Z = 0).

To specify a custom plane manually

1. On the General tab of the Slice Polygons Op property editor, set Plane Normal to **Custom Plane**.
2. On the **Custom Plane** tab, set the **Position** and **Rotation** values of the custom plane you want to use.

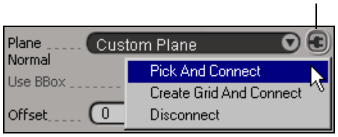
   The values entered here are interpreted as either local or global coordinates, depending on the value of **Reference** set on the General tab.

3. Specify the **Plane Normal**:

   - **X Axis** cuts along the YZ plane (X = 0).
   - **Y Axis** cuts along the XZ plane (Y = 0).
   - **Z Axis** cuts along the XY plane (Z = 0).

To specify a custom plane interactively

Connection icon



1. Connect the cutting plane to an object by doing one of the following:

   - To use an existing object in the scene to define a custom plane, right-click on the connection icon next to the **Plane Normal** on the General tab of the Slice Polygons Op property editor.

     Choose **Pick and Connect**, then click on the desired object in any view. The cutting plane immediately aligns with the center of the object you picked.

   - To create a new implicit grid and use it to define a custom plane, press **Create Grid and Connect** (near the bottom of the General tab). An implicit grid is created at the position of the current cutting plane.

     Alternatively, you can invoke this command by right-clicking on the connection icon and choosing **Create Grid and Connect**. If the Knife tool is active, you can also press and hold the right mouse button, then choose this command from the context menu.

   The connection icon turns red to show that an object is connected.

2. Translate and rotate the object into the desired position. As you move the object, the cutting plane updates automatically. The **Position** and **Rotation** values on the Custom Plane tab of the Slice Polygons Op property editor show the current position of the object, but cannot be modified directly.

3. When you are done, you can disconnect the object (if desired) by right-clicking on the connection icon and choosing **Disconnect**. The cutting plane is no longer affected by the transformation of the object.

   Alternatively, if you no longer need the object in your scene, you can simply delete it and it is disconnected automatically.

   You can also leave the object connected, for example, if you want to animate the position and angle of the slice by animating the transformations of the object. In

   If you leave the object connected, you can activate **Update With Transfo**. When this option is on, the slice is updated every time the sliced object is moved (not just when the slicing object is moved). This allows you to make the slicing object a child of the sliced object and transform the whole hierarchy without affecting the position of the slice. However, it is best to leave this option off unless necessary because it forces the slice to be recalculated every time the object is transformed.

Offsetting the Cut

You can offset the slice along the normal direction of the cutting plane. This can be useful, for example, if you want to animate the position but not the angle of the slice.

You can also force the slice to pass through a point's position by offsetting it the required amount. This is useful if you missed slicing through a desired vertex when using the interactive Knife tool.

To set the offset manually

- Adjust the **Offset** slider on the General tab of the Slice Polygons Op property editor. The slice slides along the normal of the cutting plane.

  The units for the offset value depend on whether **Use BBox** is on or off. If it is off, values are in Softimage units. If it is on, the slider's name changes to **Offset %** and values are percentages of the object's bounding box. The bounding box is calculated with respect to the local or global frame, depending on the value of **Reference**. If you are using custom planes, **Use BBox** is not available and the offset value is always in Softimage units.

To snap the offset to the location of a point

1. Click **Snap Offset to Point** on the General tab of the Slice Polygons Op property editor. A picking session begins.

   Note that **Snap Offset to Point** has no effect if **Distribute Slices** is on.

2. Pick the desired vertex in a 3D view. The vertex does not need to be on the same object as the slice—simply click on the desired object and then click on the desired point.

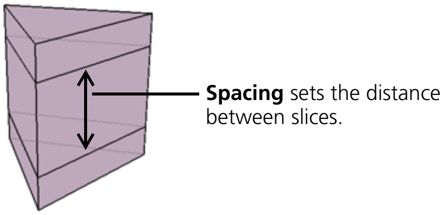
   The correct **Offset** value is automatically calculated and the slice passes through the picked point.

Making Multiple Slices

To create multiple parallel slices, set **Nb Slices** on the General tab of the Slice Polygons Op property editor to the number of slices you want.

There are two ways to control the spacing between the slices:

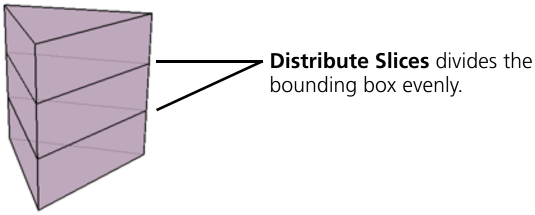
- To use a specific distance between adjacent slices, set **Spacing** to the desired value. If **Use BBox** is off, values are in Softimage units. If it is on, values represent percentages of the extent of the object's bounding box in the appropriate axis.



Spacing sets the distance between slices.

- To distribute the slices evenly along the extent of the object's bounding box in the direction of the cutting plane's normal, turn **Distribute Slices** on.

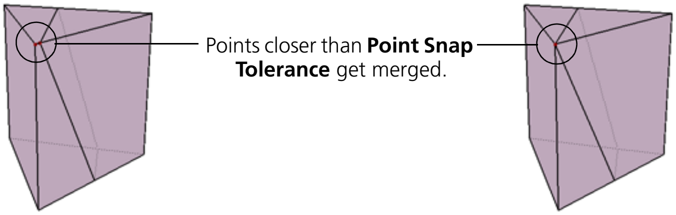
  Note that this option is available only when **Use BBox** is on, so you cannot use it with custom planes. In addition, the bounding box is calculated with respect to the local or global frame, depending on the value of **Reference**.



Distribute Slices divides the bounding box evenly.

Removing Small Edges and Polygons

Slicing sometimes creates very small edges and polygons when a slice passes close to an existing vertex. You can eliminate these small components using **Point Snap Tolerance** on the General tab of the Slice Polygons Op property editor—if a new point created by the slice operator is closer to an existing point than the tolerance value, the new point will get snapped to the existing point and welded. As a result, some of the new edges are no longer exactly parallel to the cutting plane, but no existing points get moved.



Points closer than **Point Snap Tolerance** get merged.

Disconnecting Edges and Deleting Polygons

On the General tab of the Slice Polygons Op property editor, **Action** specifies the components created or removed by the slice operation:

- **None** simply creates edges at the slice position.
- **Disconnect Edge Loop(s)** breaks the object apart into separate, disconnected islands of polygons along the new edges by creating pairs of opposing boundary edges instead of simple edges. This results in disconnected islands of polygons. The effect is the same as if you selected all new edges created by the splice operation and then chose **Modify** > **Poly. Mesh** > **Disconnect Edges**.
- **Delete Polys Above** deletes the polygons in the direction of the cutting plane's normal from the slice's position (or the last slice's position, in the case of multiple slices).
- **Delete Polys Below** deletes the polygons in the inverse direction of the cutting plane's normal from the slice position (or the first slice's position, in the case of multiple slices).

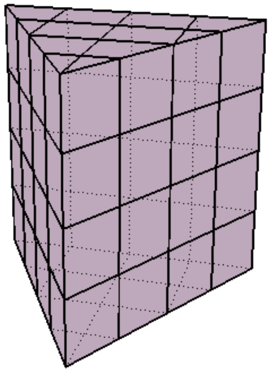
Selecting Specific Components

The three buttons in the **Select** box on the General tab of the Slice Polygons Op property editor are convenient for quickly selecting specific components:

- **Edge Loop(s)** selects all edges created by the slice operator.
- **Polys Above** selects the polygons in the direction of the cutting plane's normal from the slice position (or the last slice's position, in the case of multiple slices).
- **Polys Below** selects the polygons in the inverse direction of the cutting plane's normal from the slice position (or the first slice's position, in the case of multiple slices).

Dicing Polygons

Dicing subdivides polygons by adding edges at regular intervals along one or more perpendicular planes.



You can dice polygons to achieve the same results as the **Effect** > **Subdivision** command in the Model module of SOFTIMAGE|3D.

To dice polygons

1. Do one of the following:

   - To dice an entire object, select the object.

   - To dice specific polygons or polygon clusters on the object, select them first.

2. Choose **Modify** > **Poly. Mesh** > **Dice Polygons** from the Model toolbar.

   The polygons are diced and the Dice Polygons Op property editor opens.

Dice Options

Specifying the Cutting Planes

The cutting planes used for dicing are specified by a combination of the **Style**, **Frame**, and **Merge** parameters in the Dice Polygons Op property editor.

Style selects the axes to cut along. The result is a crisscross pattern across the Z = 0 plane (**XY**), X = 0 plane (**YZ**), Y = 0 plane (**XZ**), or all planes (**XYZ**).

Frame determines which set of axes to use:

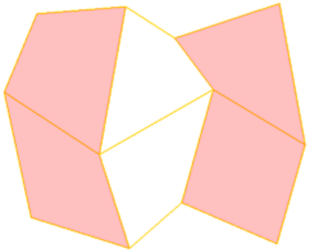
- **Global** uses the scene origin as reference.

- **Local Object** uses the object's own center as reference.

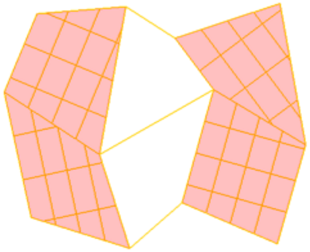
- **Local Component** uses the local reference frame of the affected polygons. The local reference frame can be based on each individual polygon, each group of adjacent polygons, or all selected polygons, depending the **Merge** parameter. **Local Component** has no effect if you diced an object instead of components.

Merge determines how the components are diced:

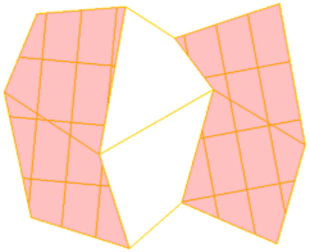
- **None** dices each polygon separately.
- **Adjacent Polygons** performs one dice operation on each group of connected polygons.
- **All** performs a single dice operation on all polygons together, whether or not they are connected.



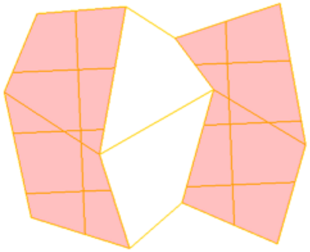
Two groups of adjacent polygons are selected.



Merge = None
Each polygon is diced individually. In this example, there are 3 slices in both X and Z, and **Frame** is set to **Local Component**.



Merge = Adjacent
Connected polygons are diced as one piece by a 3x3 pattern. Disconnected polygons are diced separately.



Merge = All
All polygons are diced together as one piece by a 3x3 pattern. Notice that one of the vertical slices misses the selected polygons.

Specifying the Number of Slices

You can specify the number of slices directly using the **X**, **Y**, and **Z** sliders, or specify a uniform grid:

- To set the number of slices directly, set **Number of Slices X**, **Y**, and **Z** to the desired number in each direction. The slices are spaced so that they evenly divide the bounding box with respect to the reference set by the **Frame** parameter.
- To use a uniform grid for slicing, activate **Grid Like** and set the desired **Grid Size**. XSI automatically calculates the required number of slices in each direction.

Removing Small Edges and Polygons

Dicing sometimes creates very small edges and polygons when a slice passes close to an existing vertex. You can eliminate these small components using **Point Snap Tolerance**—if a new point created by the dice operator is closer to an existing point than the tolerance value, the new point gets snapped to the existing point and welded. As a result, some of the new edges are no longer exactly parallel to the cutting plane, but no existing points get moved.

Disconnecting Edges

To break the object apart into separate, disconnected polygons and islands of polygons along the new edges, activate **Disconnect Edges**. The dice operator creates pairs of boundary edges instead of simple edges. The effect is the same as if you selected all new edges created by the dice operation and then chose **Modify** > **Poly. Mesh** > **Disconnect Edges**.

Selecting Edge Loops

To select all edges or boundaries created by the dice operator, click **Select Edge Loops**.

Symmetrizing Polygons

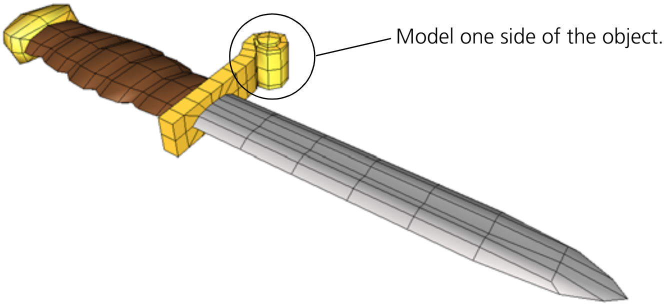
You can model one half of a polygon mesh object and then symmetrize it. This creates new polygons that mirror the geometry on the original side. Unlike Duplicate by Symmetry, the new components belong to the same object and share the same clusters: no new objects or clusters are created. In addition, materials, textures, and other cluster properties are propagated symmetrically.

If you are modeling a perfectly symmetrical object, you can build half of it and then symmetrize the whole object to create the other half. If the object is not symmetrical, you can symmetrize selected polygons. For example, you can mirror just the ear of a character who has a scar on one cheek.

Overview of Symmetrizing Polygons

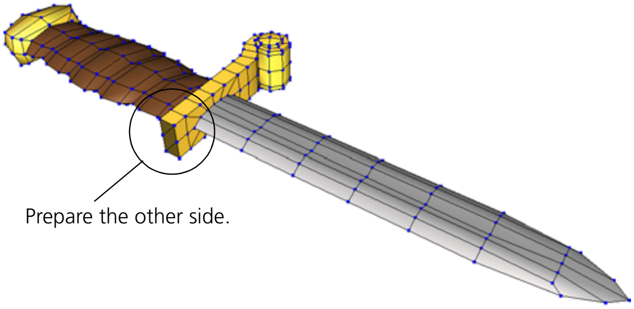
1. Model the polygons on one side of the object. You can use any combination of deformations (operations that move components) and topology modifications (operations that add or delete components).

   In the example below, an ornamental curlicue was added to the hilt of the dagger.


Model one side of the object.

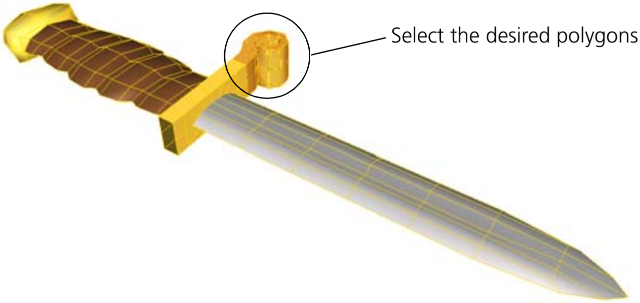
2. Prepare the other side of the object for symmetrization.

   For example, if you intend to merge the symmetrized portions using the Weld or Bridge options, then you may need to create holes for the new polygons to fit and add vertices to aid the merge.


Prepare the other side.

3. Select the polygons to be symmetrized. You can select one or more of the following:

   - Individual polygons.
   - Polygon clusters.
   - Polygon mesh objects. Selecting an object is equivalent to selecting all its polygons.

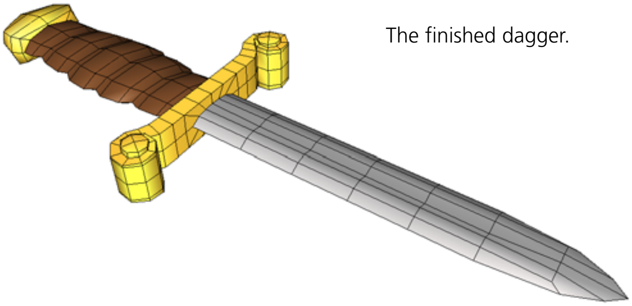


Select the desired polygons.

4. Choose **Modify** > **Poly. Mesh** > **Symmetrize Polygons** from the Model toolbar.
5. In the Symmetrize Polygon Op property editor, set the parameters as desired:

   - To control the plane of symmetry, see *Specifying the Plane of Symmetry* on page 263.
   - To connect the new polygons with pre-existing polygons, see *Welding or Bridging Edges* on page 265.

   See also *Notes on Symmetrizing Polygons* on page 266 for other information.



The finished dagger.

Specifying the Plane of Symmetry

You can specify the plane of symmetry used for mirroring the polygons with the **Reference** and **Plane Normal** parameters on the General tab of the Symmetrize Polygon Op property editor. If **Plane Normal** is set to **Custom Plane**, then the parameters on the Custom Plane tab are also used.

Reference

Reference specifies the coordinate system used for the plane of symmetry:

- **Global** mirrors the polygons about the scene origin.
- **Local** mirrors the polygons about the object's center.

Plane Normal

Plane Normal specifies the plane across which the polygons are mirrored:

- **X Axis** mirrors the polygons across the YZ plane.
- **Y Axis** mirrors the polygons across the XZ plane.
- **Z Axis** mirrors the polygons across the XY plane.
- **Custom Plane** allows you to specify any other plane. See the next section, *Custom Planes*, for more information.

Custom Planes

When **Plane Normal** is set to **Custom Plane** on the General tab of the Symmetrize Polygon Op property editor, the parameters on the Custom Plane tab specify the plane of symmetry. You can adjust the position and orientation of the plane manually, or you can set these values automatically based on the current reference plane used for manipulation. You can also update the reference plane used for manipulation based on these values.

For more information about using reference planes for manipulation in general, see *Chapter 17: Reference Planes* in the *Fundamentals* guide.

To activate a custom plane

1. On the General tab of the Symmetrize Polygon Op property editor, set **Plane Normal** is set to **Custom Plane**,
2. On the Custom Plane tab, define a plane manually or automatically using one of the procedures that follow.

To set the position and orientation manually

- Set the values on the Custom Plane tab of the Symmetrize Polygon Op property editor:
  - The **Position** and **Rotation** parameters define the center of the plane of symmetry.

If **Reference** is set to **Global** on the General tab of the Symmetrize Polygon Op property editor, these values are relative to the scene origin. If **Reference** is set to **Local**, these values are relative to the object's own center.

- **Plane Normal** specifies the plane to mirror across: **X Axis** (YZ plane), **Y Axis** (XZ plane), or **Z Axis** (XY plane).

To help visualize the custom plane when you adjust values manually, click **Set As Ref Plane**. A cyan grid appears, showing the current reference plane. Click **Set As Ref Plane** again to update the displayed grid after you change another value.

To set the position and orientation automatically

1. Do one of the following:

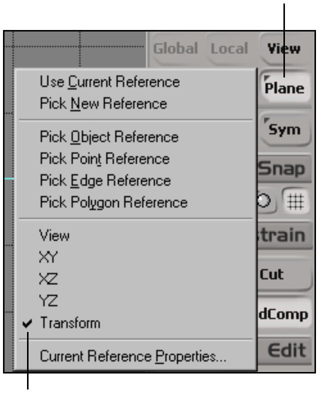
   - If necessary, create a new reference plane. For more information, see *Creating Reference Planes* in Chapter 17 of the *Fundamentals* guide.

   - If you have already defined the reference plane you want, make sure that it is the active one. For more information, see *Activating Reference Planes* in Chapter 17 of the *Fundamentals* guide.

2. Click **Update from Ref Plane** on the Custom Plane tab of the Symmetrize Polygon Op property editor.

   The **Position**, **Rotation**, and **Plane Normal** parameters are updated with the values from the active reference plane. In addition, **Reference** is set to **Global** on the General tab.

Plane manipulation mode



Transform active reference frame

To update the reference plane for manipulation

You can update the reference plane used for manipulation with the values from the Custom Plane tab of the Symmetrize Polygon Op property editor. This is useful if you want to move elements relative to the same plane you used for symmetrizing polygons.

- Click **Set As Ref Plane** on the Custom Plane tab of the Symmetrize Polygon Op property editor.

  A transient reference plane is defined using the values of the Position, Rotation, and Plane Normal parameters. If **Reference** is set to **Local** on the General tab, these values are converted to global coordinates first.

  In addition, the manipulation mode is automatically set to **Plane** and the active reference plane is set to **Transform** (as shown on left).

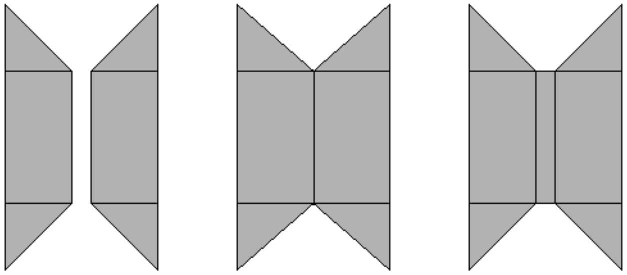
Offsetting the Result

Offset shifts the created polygons from the plane of symmetry by the specified amount. Values are in Softimage units. Note that when **Reference** is set to **Local**, the size of units are affected by the object's scaling.

Welding or Bridging Edges

The **Weld Options** parameters on the General tab of the Symmetrize Polygon Op property editor allow you to connect the boundary edges of the mirrored polygons with the original geometry by welding (merging edges) or bridging (adding polygons between edges).

- **Weld** connects polygons on the mirrored section and the original geometry by welding or bridging boundary edges, and activates the other Weld Options.
  - If Weld is on and Bridge is off, boundaries are merged (welded).
  - If both Weld and Bridge are on, boundaries are connected by new polygons.
  - If Weld is off, polygons are not connected and the other Weld Options are deactivated. Pairs of existing boundaries on the original geometry are never welded or bridged.
- **Bridge** controls the behavior for connecting edges.
  - When this option is off, pairs of facing edges are merged (welded).
  - When on, pairs of edges get connected by new polygons.



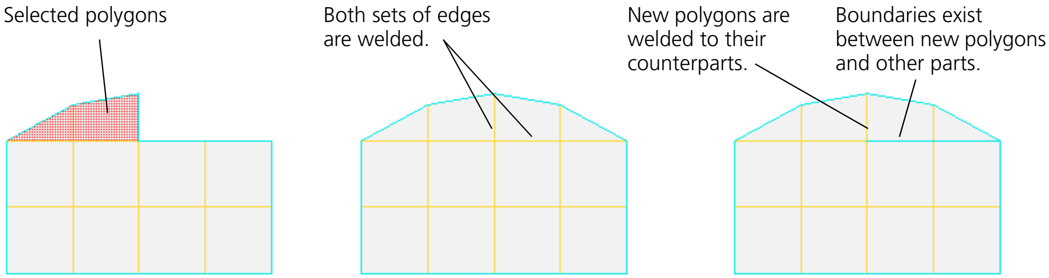
Weld **off**

Weld **on**
Bridge **off**

Weld **on**
Bridge **on**

- **Distance** controls the tolerance between facing edges. As you increase this value, more pairs of polygons become connected.

- **Symmetric pairs only** restricts the welding or bridging operation to the new polygons created by the Symmetrize Polygon Op and their counterparts on the original geometry.


Selected polygons
Both sets of edges are welded.
New polygons are welded to their counterparts.
Boundaries exist between new polygons and other parts.

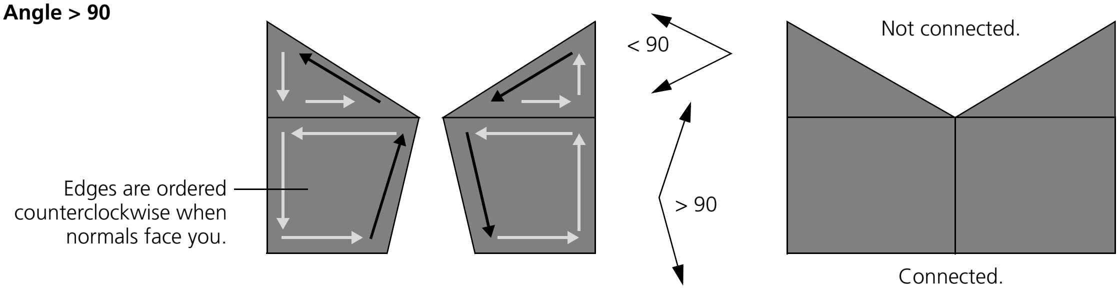
Original geometry

Symmetric pairs only **off**

Symmetric pairs only **on**

- **Angle > 90** connects two boundaries only if the difference between their orientations is greater than 90 degrees. This prevents the symmetry operator from connecting pairs of edges that are not facing each other.

Angle > 90



Notes on Symmetrizing Polygons

When symmetrizing polygons, there are some things to be aware of.

Materials, Textures, and Other Cluster Properties

When you symmetrize polygons, the new components share the same clusters as their counterparts on the original geometry. Local materials, textures, and other cluster properties (such as hard edges, polygon visibility, and the like) are extended symmetrically. In addition, weight maps and texture UV coordinates are also propagated symmetrically.

Envelopes

If the object is already used as an envelope when you symmetrize polygons, the envelopes weights are propagated symmetrically. However, the newly created points are assigned to the same deformers as their symmetric counterparts. In most cases, you probably want the new points to be assigned to the corresponding deformers on the other side. To fix this, you can mirror the envelope weights manually—see *Mirroring Envelope Weights Symmetrically* in Chapter 3 of the *Character Animation* guide.

Shapes

Existing shapes are extended when you symmetrize polygons. If a point on the original is included in one or more shapes, its new symmetric counterpart is added to the same shapes (new shapes are not created). For more information about shape animation in general, see *Chapter 4: Shape Animation* in the *Nonlinear Animation* guide.

Transforming Symmetrized Objects While the Stack is Disabled

When you symmetrize polygons, the kinematic state of the object is stored. This is for performance reasons so that the operator does not need to be re-evaluated whenever you move the object. However, it also means that under certain circumstances, you will not get the expected results when you perform certain operations in the stack.

If **Reference** is set to **Global**, you should avoid performing any operation that modifies the kinematic state of the object at the point in the stack where the Symmetrize Polygon Op node is evaluated. Such operations include the following things:

- Transforming the object while the Symmetrize Polygon Op node is disabled in the operator stack.
- Moving the object's center while the Symmetrize Polygon Op node is disabled.
- Freezing transformations while the Symmetrize Polygon Op node is disabled.
- Moving the object's center or freezing transformations, and then moving the Center operator across the Symmetrize Polygon Op node in the operator stack.

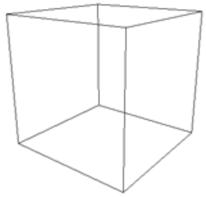
Maintaining Symmetry After Symmetrizing

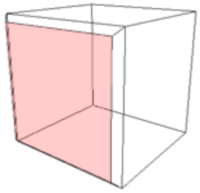
When you symmetrize polygons, the effect is immediate. If you make further modifications to the object, symmetry is not maintained automatically. However, there are a few ways to maintain symmetry:

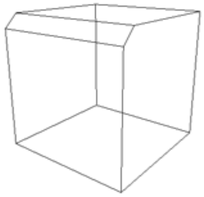
- Apply a symmetry map and then manipulate components in **Sym** mode. For more information, see *Manipulating Components Symmetrically* on page 117.
- Disable the operator stack from the Symmetrize Polygon Op node up before performing other modeling operations, and then re-enable it when you are done. For more information, see *Disabling the Top of the Stack* on page 62.
- Perform other modeling operations, and then move the Symmetrize Polygon Op node above the other operators in the stack. This works best if you move the Symmetrize Polygon Op node across only deformation nodes. In certain cases, moving the node may not be allowed or may give unexpected results. For more information, see *Changing the Order of Operators* on page 60.

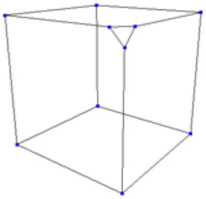
Beveling Polygon Mesh Components

The **Modify** > **Poly. Mesh** > **Bevel Components** command on the Model toolbar smooths corners by adding polygons. You can bevel polygons, points, and edges. In addition to a simple flat bevel, you can use multiple subdivisions to achieve a rounded effect.


Original geometry


Beveling a polygon


Beveling an edge


Beveling a point

To bevel polygon mesh components

1. Select one or more polygons, edges, points, or clusters on a polygon mesh object.

   Note that if you select polygons, only the contour of the selection is beveled. The interior edges are not beveled. This means that you cannot bevel a closed object like a cube by beveling all its polygons because the selection has no contour—instead, select all its edges.

   Also note that if the object has been scaled non-uniformly along different axes, then the beveling will also be non-uniform. If you want uniform beveling, freeze the object's scaling before applying the bevel. For more information about freezing transformations, see *Freezing Transformations* in Chapter 16 of the *Fundamentals* guide.

2. Choose **Modify** > **Poly. Mesh** > **Bevel Component**.

   The components are beveled and the Bevel Op property editor opens.

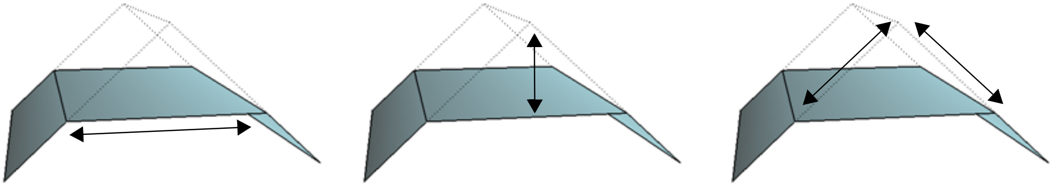
3. Adjust the parameters as described in the sections that follow.

   If you are beveling selected points, you can adjust only the **Distance**. None of the other parameters are available for points.

Controlling the Size and Position of the Bevel

Distance, **Type**, and **Units** on the General tab of the Bevel Op property editor control the size of the bevel:

- **Distance** is the amount of beveling. The values depend on the **Type** and **Units** settings.
- **Type** controls the direction in which **Distance** is measured:
  - **Bevel width** is the width of the new polygons created by the Bevel operator.
  - **Along bisectors** is the distance between the original, unbeveled edge and the new polygon created by the Bevel operator.
  - **Along adjacent polygons** is the distance between the original, unbeveled edges and the new edges created by the Bevel operator.



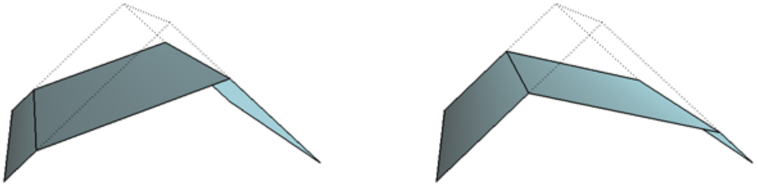
Type = Bevel width

Type = Along bisectors

Type = Along adjacent polygons

- **Units** determines the values used by the **Distance** slider:
  - **Relative to selection** uses relative values based on the size of the edges to be beveled.
  - **SI Units** allows you to enter exact values in Softimage units.

In addition, **Center** controls the position of the bevel, allowing you to create lopsided bevels.



Center < 0

Center > 0

Controlling the Behavior at Joints

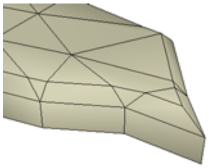
Joint options on the General tab of the Bevel Op property editor control how beveling behaves at joints between edges.

Managing Collisions

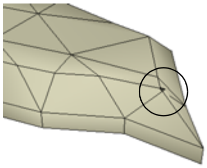
As you increase the bevel distance, the beveled edges may collide with other components on the object, resulting in overlapping polygons. This problem can be especially acute near sharp corners.

When **Manage collisions** (available on the General tab of the Bevel Op property editor) is on, the Bevel operator automatically collapses points that collide with each other so that no overlapping occurs. Note that only collisions between adjacent components are detected.

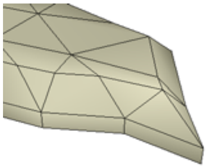
You can turn this option off if you need to ensure that the number of vertices does not change when the bevel distance is animated, or if you prefer to manually clean up any overlapping polygons.



At low bevel distances, there are no collision problems.



As the distance increases, collisions cause overlapping polygons.

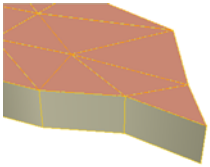


When **Manage collisions** is on, colliding vertices are automatically collapsed.

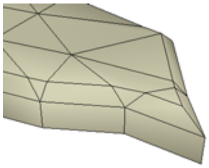
Mitering Uniformly

The **Uniform miter** option on the General tab of the Bevel Op property editor controls the tesselation at junctions between two beveled edges:

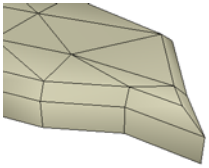
- When this option is off, 2-edge junctions with three or more polygons on the same side are tesselated by polygons in order to preserve the surrounding geometry.
- When this option is on, all 2-edge junctions are single edges. This may result in cleaner geometry in the beveled region but may also change the shape somewhat in the surrounding areas.



Polygons selected before beveling.



Uniform mitering = off

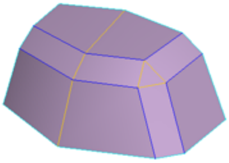


Uniform mitering = on

Marking Hard Edges

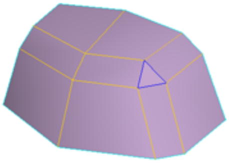
You can automatically mark the edges created by the Bevel operator as hard. This is useful for creating discontinuities in the shading to achieve a faceted appearance. There are three independent options for **Hard edges** on the General tab of the Bevel Op property editor:

- **On contour** marks the outer edges created by the Bevel operator as hard.



Contour edges marked as hard.

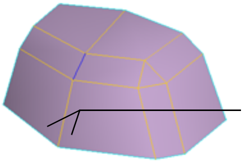
- **On 3+ edge junctions** marks the edges at junctions between three or more beveled edges as hard.



Edges on 3+ junctions marked as hard
marked as hard.

- **Min angle for 2 edge junction** controls the minimum angle for marking 2-edge junctions as hard. If the dihedral angle between two adjacent polygons on the same side of the beveled edges is greater than this value in degrees, the junction is marked as hard. A value of 0 marks all 2-edge junctions as hard, and a value of 180 marks none as hard.

  If you set this angle to the same value as the Angle for automatic discontinuity in the object's Geometry Approximation property, then discontinuities on the object propagate to the beveled region.



Edge on 2-edge junction marked as hard.

The slider controls the minimum angle between any two polygons on the same side for the corresponding edge to be marked hard.

Rounding

The options on the Rounding tab of the Bevel Op property editor allow you to create rounded bevels with multiple subdivisions.

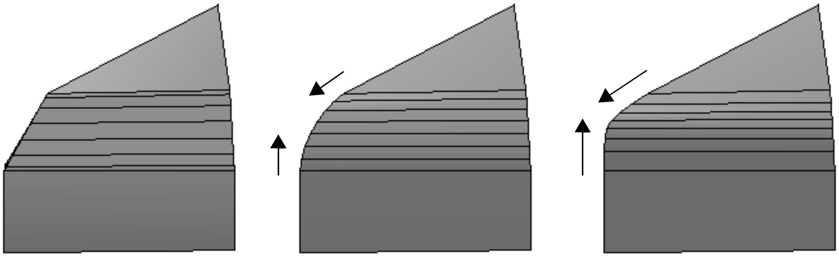
To make rounded bevels

- Set **Nb subdivs** to the desired number of subdivisions.

  As long as this value is nonzero, the other options on the Rounding tab are available and can be used to control the rounding as described in the sections that follow.

Controlling the Sharpness of the Profile

Sharpness on the Rounding tab of the Bevel Op property editor controls the shape of the profile. A value of 1 uses an arc-like segment (the default). Higher and lower values increase or decrease the tension, or the influence of the tangency of the adjacent polygons, similar to adjusting the length of Bézier control handles. A value of zero gives a flat profile.



Sharpness = 0 **Sharpness = 1** **Sharpness > 1**

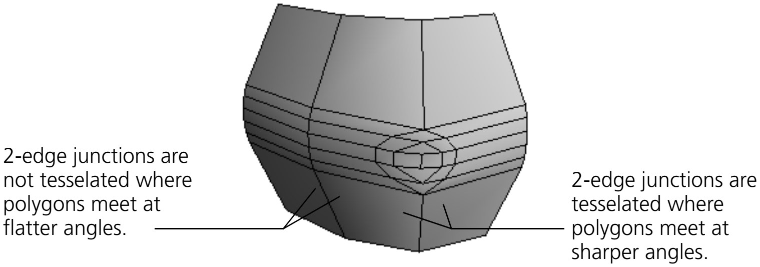
You can also control the shape of the profile using the Extrapolation slider as described in *Extrapolating the Rounding Displacement* on page 274.

Tesselating Junctions

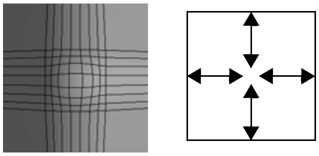
The **Junctions** options control the tesselation where beveled edges meet. This lets you smooth the shape and shading on the insides of junctions.

- **Tesselate** is the “master control” that turns tesselation on or off. When this option is off, no junctions are tesselated and the other **Junctions** options are unavailable. When this option is on, all junctions of three or more beveled edges are tesselated and you can use the other options to tesselate 2-edge junctions and control other aspects of the tesselation.

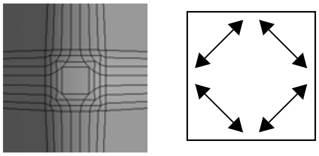
- **2 edge junction min angle** controls the minimum angle for tesselating 2-edge junctions. If the dihedral angle between two adjacent polygons on the same side of the beveled edges is greater than this value in degrees, the junction is tesselated. A value of 0 tesselates all 2-edge junctions, and a value of 180 tesselates none.


2-edge junctions are not tesselated where polygons meet at flatter angles.
2-edge junctions are tesselated where polygons meet at sharper angles.

- **3+ edge junctions parameterization** controls the direction of the smoothness and the flow lines at junctions where three or more edges meet.
  - **Plus** maintains continuity between the junction's center and the middle of each side of the junction's contour.

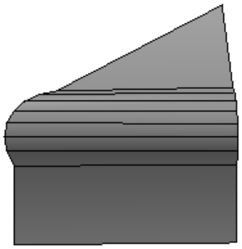


  - **Diamond** maintains continuity between adjacent sides of the junction's contour.

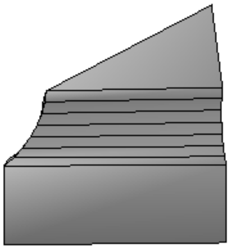


Extrapolating the Rounding Displacement

Extrapolation Amount on the Rounding tab of the Bevel Op property editor multiplies the displacement of the rounding effect above the flat bevel. Positive values produce exaggerated bulges, and negative values produce incised bevels. A value of zero produces no extra displacement, and a value of –1 produces an exactly inverted displacement.

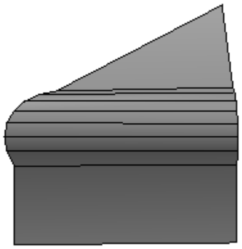


Amount > 0

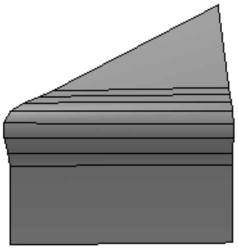


Amount < 0

When **Amount** is nonzero, use **Keep tangency** to ensure that the borders of the beveled region remain continuous with the adjacent polygons.



Keep tangency = off

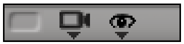


Keep tangency = on

Inverting Polygons and Polygon Meshes

The **Modify** > **Poly. Mesh** > **Invert Polygons** command on the Model toolbar changes the direction of the normals of a polygon mesh object. It can also invert the normals of selected polygons.

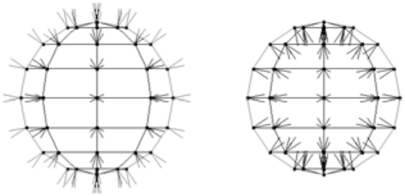
Polygon Normals and Double Edges



Normals are vectors that are perpendicular to the surface of polygons. They indicate the “outside” of an object for the purpose of shading and textures.

Normals are represented by thin blue lines. To display or hide them, click the eye icon (Show menu) of a 3D view’s menu bar and choose **Normals**.

Normals should point toward the camera.


Correct

Incorrect

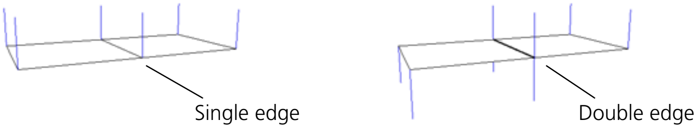
The direction of the normals is determined by the direction in which you draw the vertices of a polygon.

- If the vertices are drawn in a counterclockwise direction, the normals face toward the camera and
- If the vertices are drawn in a clockwise direction, they face away from the camera.

If the normals of two adjacent polygons point in opposite directions, then a double edge is drawn between the polygons to preserve the integrity of the mesh.

Normals point the same direction.

Normals point in opposite directions.



The double edges are considered boundaries and are displayed in light blue if **Boundaries and Hard Edges** are visible in a 3D view.

Inverting Polygon Normals

You can invert all the polygons of an object by inverting the object itself or you can invert only the selected polygons.

To invert all polygons of a mesh object

1. Select a polygon mesh object.
2. Choose **Modify** > **Poly. Mesh** > **Invert Polygons**. The normals are reversed and the Inverse Normals property editor opens.

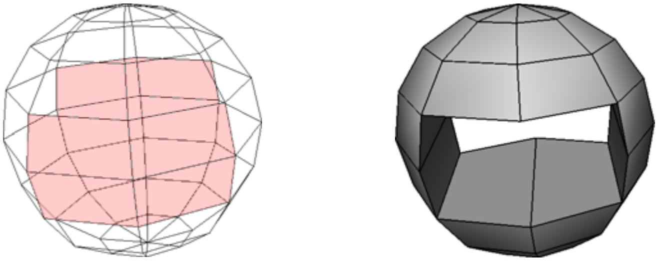
   If desired, you can mute the inversion by turning the **Inverse** option off.

To invert selected polygons on a mesh object

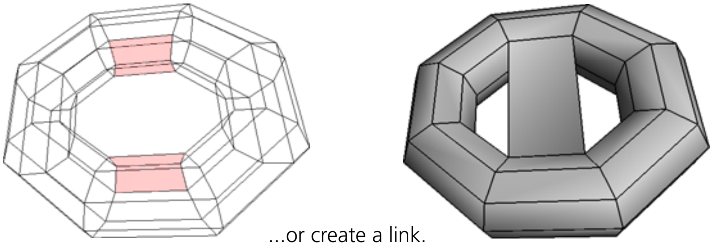
1. Select a polygon mesh object.
2. Select some polygons on the mesh object.
3. Choose **Modify** > **Poly. Mesh** > **Invert Polygons**. The normals of the selected polygons are reversed.

Bridging Polygons

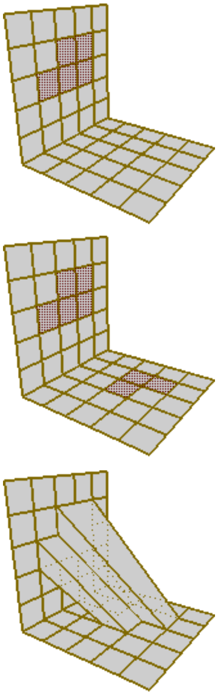
The **Modify** > **Poly. Mesh** > **Bridge Polygons** command on the Model toolbar creates a bridge or tunnel between two sets of polygons on the same object.



Bridging polygons can create a hole...



...or create a link.



To bridge polygons

1. Select one or more adjacent polygons on a polygon mesh object.

2. Add one or more other polygons on the same object to the selection. The polygons should be adjacent to each other, but not adjacent to the first group of polygons.

   For best results, make sure that the two selected areas have a similar outline when colinear edges are considered as one. For multiple bridge operations on the same object, perform them one pair at a time to make sure the correct pairs of areas are bridged.

3. Choose **Modify** > **Poly. Mesh** > **Bridge Polygons**.

   The selected polygons are deleted and the resulting holes are joined by new polygons.

To bridge between two different polygon meshes, you can merge or blend them into one object first as described in *Blending and Merging Polygon Meshes* on page 278.

Blending and Merging Polygon Meshes

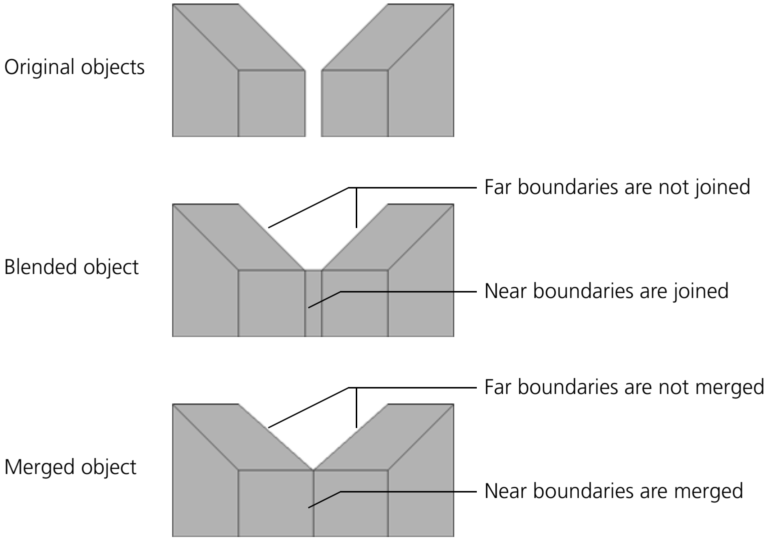
You can combine two or more polygon mesh objects using the **Blend** and **Merge** commands on the **Create** > **Poly. Mesh** menu of the Model toolbar. Although these commands can be used in many situations, they are particularly useful for combining two halves of a symmetric object.

Both commands create a new polygon mesh object. The new object inherits materials and textures—clusters are automatically created on the new mesh for these properties.

The two commands differ in how they treat boundary edges on different objects when the boundaries are close to each other:

- With Blend, nearby boundaries on different objects are joined by new polygons.
- With Merge, nearby boundaries on different objects are merged into a single edge at the average position.

There is a **Tolerance** parameter for determining the maximum distance in Softimage units between boundaries for them to be considered "nearby".


Original objects
Far boundaries are not joined
Blended object
Near boundaries are joined
Far boundaries are not merged
Merged object
Near boundaries are merged

To blend or merge polygon mesh objects

1. If desired, you can select some or all of the polygon mesh objects to be blended before choosing the command.
2. Do one of the following:

   - Choose **Create** > **Poly. Mesh** > **Blend**.

     or

   - Choose **Create** > **Poly. Mesh** > **Merge**.
3. Click to pick any additional polygon mesh objects.

   If you make a mistake, Ctrl+click to undo the last pick. You can also press Esc to cancel the operation.
4. When you have finished picking, right-click in a 3D view.

   A new polygon mesh object is created and the Merge Meshes property editor opens with the **Blend** option on or off as appropriate.
5. Adjust the **Tolerance** as desired. Boundary edges that are closer than this value in Softimage units are blended or merged.

Transferring and Merging Attributes

You can transfer and merge clusters with properties from the input objects to the generated object. The cluster properties that you can transfer in this way include materials, texture UVs, color at vertices maps, property weight maps (but not envelope weights), and shapes.

These procedures apply not only when merging polygon meshes, but also when creating an object by subdividing as described in *Generating a New Subdivision Surface Object* on page 337.

The transfer of attributes is controlled by the buttons under **Merge Attributes** or **Transfer Attributes** in the property editor:

- **All** transfers all materials, texture UVs, color at vertices maps, property weight maps, and shapes from the input objects. In the case of merge, clusters and properties from different objects are merged if XSI considers them to match.
- **Materials** transfers clusters with materials only. Again in the case of merge, clusters and properties from different objects are merged if XSI considers them to match.
- **Edit** opens the Transfer/Merge Attributes window, where you can control the transfer and merging manually. Use this option if you want to transfer only certain attributes, or if XSI did not merge attributes the way you want. See the next section, *Transferring and Merging Attributes Manually*.

Transferring and Merging Attributes Manually

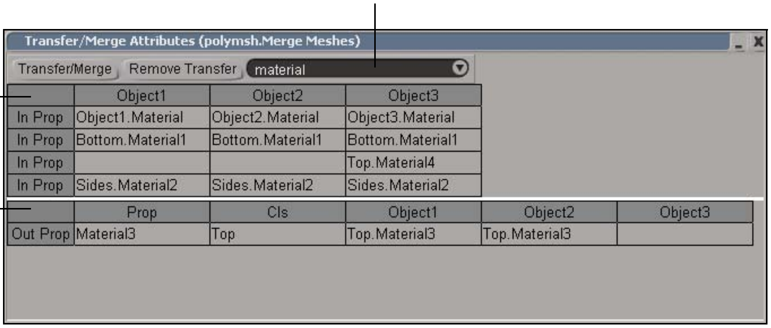
Use the Transfer/Merge Attributes window to control how clusters and cluster properties are transferred from input objects to generated objects.

To display this window, click **Edit** under **Merge Attributes** or **Transfer Attributes** in the property editor.

Select the type of property.

Clusters and properties on the input objects are shown in the top grid.

Clusters and properties that have been transferred are shown in the bottom grid.



	Object1	Object2	Object3
In Prop	Object1.Material	Object2.Material	Object3.Material
In Prop	Bottom.Material1	Bottom.Material1	Bottom.Material1
In Prop			Top.Material4
In Prop	Sides.Material2	Sides.Material2	Sides.Material2

	Prop	Cls	Object1	Object2	Object3
Out Prop	Material3	Top	Top.Material3	Top.Material3	

The top grid contains a column for each input object. The cells show the ClusterName.PropertyName for each untransferred cluster property of the type selected from the drop-down box. The cells are arranged into rows based on XSI's best guess about which clusters and properties can be merged. Items in the same row would be merged automatically if you click **All** in the generator operator's property editor.

The bottom grid contains a row for each cluster property of the selected type that has already been transferred to the generated object. The first column shows the property name and the second column shows the cluster name. There is an additional column for each input object; these show the ClusterName.PropertyName of the input cluster properties that have been merged into the generated cluster property.

You can resize the columns and rows by clicking and dragging their borders.

To select cells in the grids

In order to transfer or remove attributes, you must first select the appropriate cells:

- To select a single cell, click on it.
- To add a cell to the selection, Ctrl+click on it.
- To select a horizontal, vertical, or rectangular range of cells, you can drag across them from one end (or corner) to the other end (or diagonally opposite corner). Alternatively, you can select one end (or corner) and then Shift+click on the other end (or diagonally opposite corner).

Selected cells are highlighted in lavender.

To transfer and merge attributes

1. Choose the type of attribute from the drop-down list at the top of the editor. For example, to transfer shapes, choose **clskey** (cluster key).
2. In the top grid, select the cluster properties to transfer.

   Clusters and properties are merged according to the following rules:

   - If no more than one cell is selected per column, then all selected cluster properties are merged.
   - If multiple cells are selected in any column, selected cluster properties in the same row are merged together. Each row with selected cells generates a separate cluster property.
3. Click **Transfer/Merge** at the top of the editor.

   New rows appear in the bottom grid showing the generated cluster properties. The items that were transferred are removed from the top grid.

To remove attributes

If cluster properties were incorrectly merged, you can “untransfer” them before retransferring and merging them correctly.

1. In the bottom grid, select any cell(s) in the row(s) corresponding to the cluster properties you want to remove.
2. Click **Remove Transfer** at the top of the editor.

   The selected rows are removed from the bottom grid, and the untransferred items appear in the top grid.

Shapes and Merged/ Blended Objects

If the original input objects have shape animation, the shape keys (sources) can be copied to the resulting merged or blended object, but there are special considerations. To make sure you get good results, follow this procedure.

To transfer shape keys to merged or blended objects

1. If the centers of the objects to be merged are not aligned, first do the following:

    - Convert the shapes to local reference mode. See *Changing the Shape Reference Mode* in Chapter 4 of the *Nonlinear Animation* guide.

    - Align the object's centers by choosing **Transform** > **Freeze Rotation**.

2. Convert all shapes on the objects to be merged to Object Relative reference mode.

3. Make sure that all objects to be merged are in their base pose (no shape applied).

4. Merge or blend the objects.

5. Transfer the shapes as described in *Transferring and Merging Attributes* on page 279. You should do this manually, otherwise XSI may merge shapes that you want to keep separate.

6. To animate the shape of the merged object, add shape tracks and instantiate shape clips using **Insert Source** as described in *Chapter 4: Shape Animation* of the *Nonlinear Animation* guide.

    If you want to convert the shape reference mode to Local Relative, you must freeze the Merge Mesh operator in the operator stack first.

If there are shape clips on both the input objects and the generated object, and the modeling relation still exists, the result is a double deformation. To animate the shape of the generated object, you should freeze its operator stack to remove the modeling relation with the input objects, and then create some shape clips.

Managing Input Objects

Whenever you generate a new object from other input objects, you can manage the inputs, for example, hide or delete them. See *Managing Generator Inputs* on page 68.

Performing Boolean Operations on Polygon Meshes

Boolean operations let you create the difference, intersection, or union of two polygon meshes. There are two ways to combine polygon mesh objects using Boolean operations:

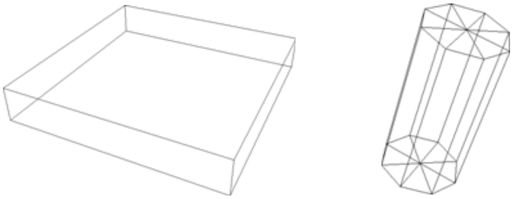
- Create a new object using **Create** > **Poly. Mesh** >**Boolean**. The new object does not inherit materials, textures, or other properties.

  or

- Modify an existing one using **Modify** > **Poly. Mesh** >**Boolean**. The modified object retains its existing materials, textures, and other properties.

Types of Boolean Operation

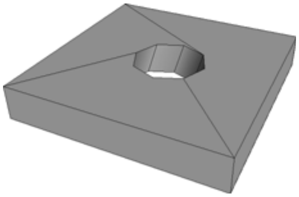
There are three possible Boolean operations: difference, intersection, and union.



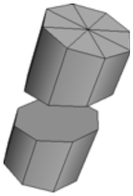
Original objects:
scaled cube and cylinder

Difference

Difference subtracts the volume of the second object from the first.



Cube minus cylinder.

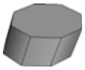


Cylinder minus cube.

You can slice a polygon mesh by subtracting it from an open one (such as a grid).

Intersection

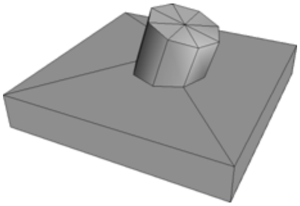
Intersection consists of the common volume contained by both objects.



Intersection of cube and cylinder.

Union

Union combines the volume of both objects.



Union of cube and cylinder.

Generating New Boolean Objects

The **Difference**, **Intersection**, and **Union** commands on the **Create** > **Poly. Mesh** >**Boolean** menu generate a new polygon mesh object. The new object does not inherit materials, textures, or other properties.

To generate a new polygon mesh using a Boolean operation

1. Select the first polygon mesh object, then add a second one to the selection. The order of selection is important only for the Difference operation.
2. Choose a Boolean operation from the **Create** > **Poly. Mesh** > **Boolean** menu.

   If you did not select both objects in step 1, you are prompted to pick the remaining ones now.

Managing Input Objects

Whenever you generate a new object from other input objects, you can manage the inputs, for example, hide or delete them. See *Managing Generator Inputs* on page 68.

Applying Boolean Operations to Existing Objects

The **Difference**, **Intersection**, and **Union** commands on the **Create** > **Poly. Mesh** >**Boolean** menu apply a Boolean operator to the first selected polygon mesh object. The modified object retains its existing materials, textures, and other properties.

To apply a Boolean operator to an existing polygon mesh

1. Select the first polygon mesh object, then add a second one to the selection.
2. Choose a Boolean operation from the **Modify** > **Poly. Mesh** > **Boolean** menu.

   If you did not select both objects in step 1, you are prompted to pick the remaining ones now.

Filtering Polygon Mesh Components

You can filter polygon mesh objects to clean up areas by removing components that match certain criteria, for example, small components that represent insignificant detail. You can filter edges, points, and polygons.

Filtering Edges

Edge filtering removes edges by collapsing them based on either their length or angle. In both cases, you can protect boundary edges with the **Keep Borders Edges Intact** option.

Edge filtering is especially useful for reducing the triangulation on polygon meshes generated by Boolean operations.

Filtering Edges by Length

When you filter polygons by area, the smallest polygons are removed.

To filter edges by length

1. Select a polygon mesh object or multiple edges.
2. Choose **Modify** > **Poly. Mesh** > **Filter Edges**. The Edge Filter Op property editor opens.
3. Make sure that **Filter Type** is set to **Edge Length**.
4. Adjust the **Edge Length** value. This is the threshold for removing edges. Edges with a length smaller than this value are collapsed.

Filtering Edges by Angle

When you filter edges by angle, points are collapsed if they are shared by exactly two edges whose angular difference is less than a threshold you specify.

This lets you combine edges that are almost colinear. Points that are shared by three or more edges are never collapsed.

To filter edges by angle

1. Select a polygon mesh object or multiple edges.
2. Choose **Modify** > **Poly. Mesh** > **Filter Edges**. The Edge Filter Op property editor opens.
3. Make sure that **Filter Type** is set to **Edge Angle**.
4. Adjust the **Edge Angle** value. This is the threshold for removing edges. Edges with an angular difference smaller than this value are combined.

Filtering Points

Filtering points on polygon mesh objects welds together vertices that are within a specified distance from each other.

To filter points by distance

1. Select the points or point clusters that you want to filter on a polygon mesh object. Selecting an object is equivalent to selecting all its points.
2. Choose **Modify** > **Poly. Mesh** > **Filter Points** from the Model toolbar.

   The selected points are filtered with the default settings and the FilterPointsByDistance Op property editor opens.
3. Adjust the **Distance**. Points that are closer to each other than this value are welded together. Points that are farther apart than this value may get welded into separate "clumps". Values are in Softimage units.
4. If desired, specify the targets for welding:
   - **Average position** welds each clump of points in the selection together at their average position.
   - **Selected point** welds each clump of points in the selection together at the position of the point that is nearest to the average position.
   - **Unselected point** welds each selected point to an unselected point on the same object.
5. If desired, adjust the other parameters as described in the sections that follow.

Cleaning Up the Results

There are two options in the Cleanup box of the FilterPointsByDistance Op property editor that remove undesirable geometry that may result from filtering points:

- **Remove new double-sided polygons** eliminates pairs of back-to-back polygons created by the filtering process. These double-sided polygons can occur where two polygons face each other across a narrow gap. In most situations, you probably don't want such double-sided polygons in the final result, so it's best to leave this option on. However, you can turn it off if you intend to use the resulting polygons for further modeling.
- **Collapse new double edges** removes double edges created by the filtering process. You can turn this option off, for example, if you want to weld two rows of non-boundary points without collapsing them all to a single vertex.

Restricting the Filtering

A couple of options in the FilterPointsByDistance Op property editor allow you to restrict the filtering process to vertices that meet specific criteria:

- **Border points only** welds only points on boundary edges, even if non-border points were included in the selection. This option is useful, for example, when joining pieces that were modeled separately and then merged into a single object.
- **Max normal angle difference** welds two points only if the difference between their normal angles is less than the specified amount. This option can be used to prevent points that are facing in opposite directions, such as on the front and back of a character's arms, from being welded together. Activate the checkbox and then adjust the slider. Values are in degrees.
- **Connected user target cluster** lets you specify a cluster of points that act as "preferred targets" and are never collapsed to other points. Click the connection icon, then choose **Connect** and pick a cluster. If you change your mind, click the connection icon again and choose **Disconnect**.

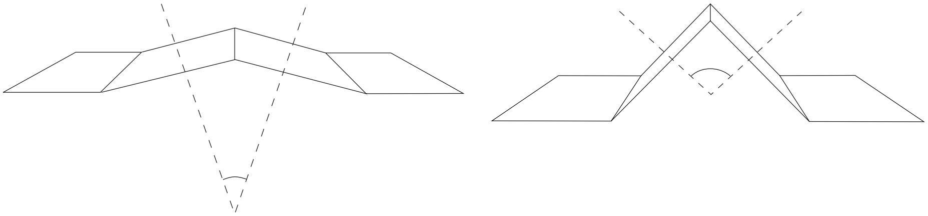
Filtering Polygons

When filtering polygons, you can remove them based on their area or their dihedral angles. In both cases, you have the option of cleaning up edges afterwards.

As an alternative to filtering polygons, you can apply polygon reduction. For more information, see *Polygon Reduction* on page 290.

Filtering Polygons by Angle

When you filter polygons by angle, adjacent polygons are merged together if their dihedral angle is less than the threshold you specify. Small angles correspond to flat areas, so this method preserves sharp detail.



Dihedral angles: flatter edges have small angles and sharper edges large angles

To filter polygons by angle

1. Select a polygon mesh object or multiple polygons.
2. Choose **Modify** > **Poly. Mesh** > **Filter Polygons**. The Polygon Filter Op property editor opens.
3. Make sure that **Filter Type** is set to **Incidence Angles**.
4. Adjust the other parameters:

   - **Polygon Incidence Angle** is the threshold for merging polygons. Adjacent polygons whose dihedral angle is less than or equal to this value are combined.
   - **Dissolve by Pair** combines polygons sequentially, one pair at a time. This ensures that, for a given polygon, the adjacent polygon being considered for co-removal is the most nearly coplanar one.

     If this option is off, polygons are dissolved all at once and the end result may consist of large, non-planar polygons.

Filtering Polygons by Area

When you filter polygons by area, the smallest polygons are removed. This eliminates small, “noisy” details.

To filter polygons by area

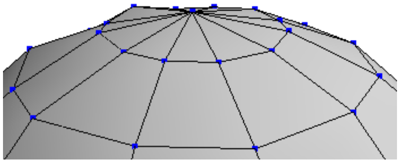
1. Select a polygon mesh object or multiple polygons.
2. Choose **Modify** > **Poly. Mesh** > **Filter Polygons**. The Polygon Filter Op property editor opens.
3. Make sure that **Filter Type** is set to **Polygon Areas**.
4. Adjust the other parameters:

   - **Polygon Area** is the threshold for removing polygons. Polygons with an area smaller than this value are removed.
   - **Action** is the method used to remove polygons: **Dissolve** or **Collapse**. For a description of the different methods of removal, see *Removing Polygon Mesh Components* on page 229.

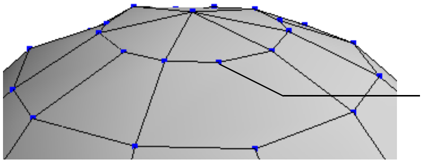
     In particular, note that Dissolve will remove areas of adjacent small polygons, but not a single small polygon surrounded by larger ones.

Cleaning Up Edges

When you filter polygons, you have the option of filtering the edges afterwards. **Filter Sub Edges** in the Polygon Filter Op property editor cleans up edges that are no longer necessary after removing polygons. If a vertex is shared by three or more polygons before filtering, but only by two polygons after filtering, then it is collapsed.

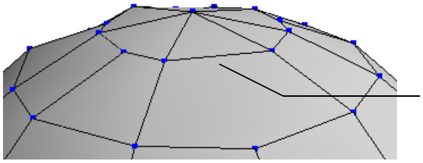


Original geometry.



Geometry after filtering polygons by angle with Filter Sub Edges off.

Point is shared by two polygons.



Geometry after filtering polygons by angle with Filter Sub Edges on.

Point is collapsed.

Filter Sub Edges does not affect points that were already shared by only two polygons before filtering, and in particular, it will not affect boundaries.

To remove those points, filter the edges as described in *Filtering Edges* on page 285.

Tips for Filtering

Here are some tips related to filtering polygon mesh components:

- Although you can filter an entire object, it's usually better to select specific areas of components. Often, there are large areas of your object that you do not need or want to be filtered, and by leaving these areas out, you can speed up the operation significantly.

  In addition, selecting components for filtering gives you more control because you can use different criteria in different areas.

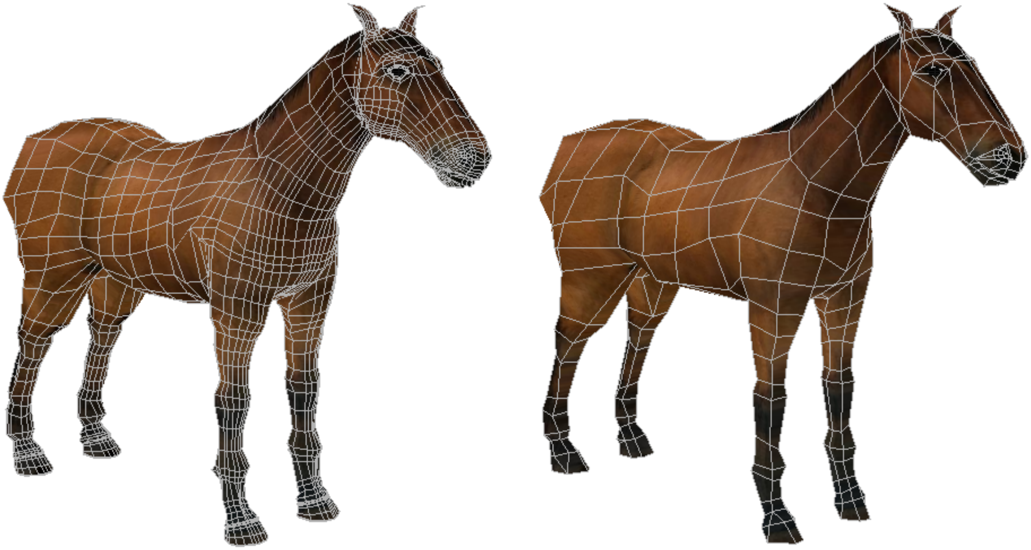
- The component filter property editors show statistics on the total number of components on the object. This information can be useful if you need to meet certain geometry requirements for games or other specific applications.

Polygon Reduction

Polygon reduction lightens a heavy object by reducing the number of polygons, while still retaining a useful fidelity to the shape of the original high-resolution version. For example, you can use polygon reduction to meet maximum polygon counts for game content, or to reduce file size and rendering times by simplifying background objects.

Polygon reduction also allows you to generate several versions of an object at different levels of detail (LODs).

Polygon reduction works by collapsing edges into points. Edges are chosen according to their “energy”, which is a metric based on their length, orientation, and other criteria. In addition, you have options to control the extent to which certain features, such as quad polygons, are preserved by the process.



To apply polygon reduction

1. Select one or more polygon mesh objects, polygon clusters, or polygons.
2. Choose **Modify** > **Poly. Mesh** > **Polygon Reduction** from the Model toolbar.

   The polygons are reduced using the default settings and the Polygon Reduction Op property editor opens. For information about adjusting the settings, see the sections that follow.

Controlling the Amount of Reduction

The options in the **Reduction Amount** box of the Polygon Reduction Op property editor specify the amount of reduction. You can specify either a percentage (ratio) or a final number of vertices or triangles. The reduction amount is animatable, so, for example, you can use an expression to control it based on the distance from the camera.

In all cases, these values are targets—the operator may not be able to achieve the exact amount of reduction for various reasons. For example, if the original object has an odd number of polygons, then it is impossible to achieve a reduction of exactly 50%.

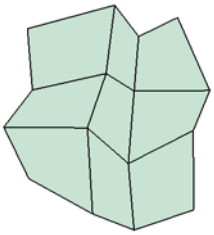
In addition, you have the option of using a weight map. This lets you paint weights to differentiate between areas where you want more or less reduction to take place. You can also use a weight map to protect parts of your object.

Although you can animate the amount of reduction, you must be careful. As with any topology operator, you may experience problems if its parameters are animated and there are other operators above it in the construction history.

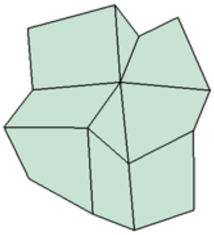
For more information about animated topology operators, see *Special Considerations for Topology Modifications* on page 59.

To set the amount of reduction

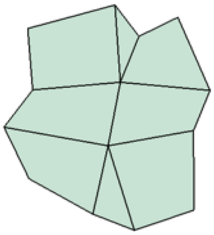
1. On the General tab of the Polygon Reduction Op property editor, set **Unit** to the desired value:
   - **Ratio** allows you to set the amount of reduction as a percentage using the **Ratio** slider.
   - **Final vertex count** lets you set the number of vertices in the result using the **Vtx Count** slider.
   - **Final triangle count** lets you set the number of triangles (polygons after tessellation) in the result using the **Tri Count** slider.
2. Use the corresponding slider to set a target for the amount of reduction.



As you increase the amount of reduction...



the edge with the lowest energy is collapsed...



followed by the next lowest, and so on.

To use a weight map to control the reduction in specific areas

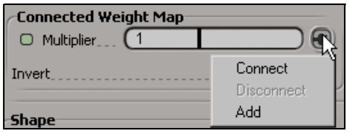
1. Before you apply polygon reduction, create a weight map as described in *Chapter 11: Weight Maps* of the *Fundamentals* guide.

   The cluster that contains the weight map must exist before you apply the Polygon Reduction operator. If you have already applied polygon reduction, then you must disable the operator before creating the weight map. This ensures that the cluster containing the weight map is created at a lower position in the construction history than the reduction operator.

   Similarly if you want to add or remove points from the cluster or paint additional weight strokes, make sure you disable the Polygon Reduction operator first.

   For more information about disabling and re-enabling operators in the stack, see *Disabling the Top of the Stack* on page 62.

2. Apply polygon reduction, or re-enable the Polygon Reduction operator if you disabled it.



3. On the General tab of the Polygon Reduction Op property editor, right-click on the connection icon to right of the **Multiplier** slider and choose **Connect**.

4. In the pop-up explorer that appears, expand the appropriate cluster and then select the desired weight map.

   By default, areas in which the weight map has a high opacity are more likely to be reduced than areas with a low opacity. Areas with a weight of 0 are reduced last—they are essentially protected from reduction unless the reduction amount is so high that the non-zero areas of the weight map cannot be reduced further.

   If you want to reverse the effect of the weight map so that areas of high opacity are protected and areas of low opacity are reduced, turn **Invert** on.

   If you want to paint additional strokes on the weight map after it is connected, make sure you disable the Polygon Reduction operator first.

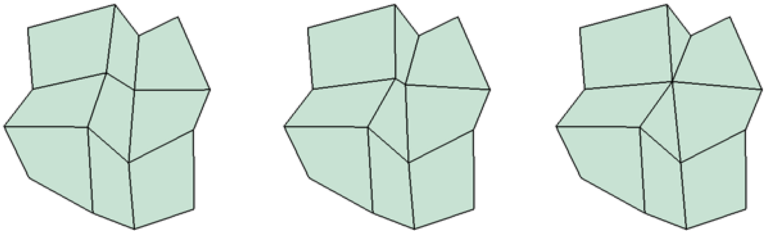
   See *Chapter 11: Weight Maps* of the *Fundamentals* guide for information about working with weight maps in general, including painting, disconnecting, and mixing (adding).

5. If desired, adjust the **Multiplier** slider to modulate how strongly the values in the weight map affect the likelihood of edges collapsing.

Using Smooth Transitions

The **Smooth Transitions** option is available on both the General tab and the LOD Controls tab of the Polygon Reduction Op property editor. When it is on, edges shrink before they collapse as the reduction amount is increased. In addition, corresponding properties, such as texture UVs and vertex colors, are interpolated. This allows the object to gradually morph when the reduction amount is animated, and avoids sudden popping in the object's shape or textures as edges collapse.

It is particularly useful when combined with **Reduce parallel edge loops** (see *Preserving Quads* on page 294) or LODs (see *Generating LODs* on page 299).

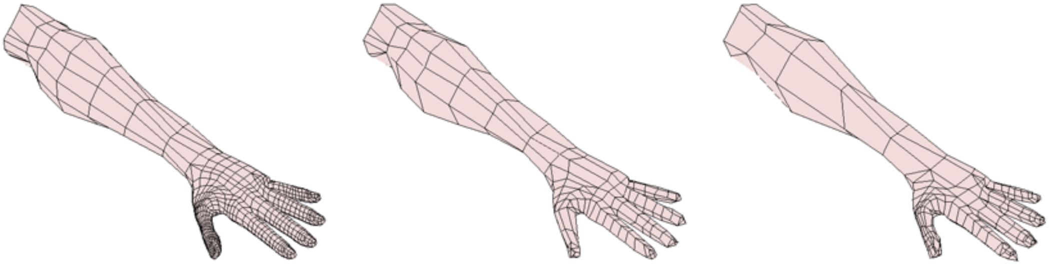


Smooth Transitions
Edges gradually shrink before they collapse.

Adjusting the Shape

A couple of options on the General tab of the Polygon Reduction Op property editor allow you to control how reduction affects the details and shape of an object:

- **Sharpness** controls the balance between preserving small, sharp details versus larger shapes. At low values, details that are small relative to the general shape of the object are more likely to be collapsed. At high values, they are more likely to be kept.

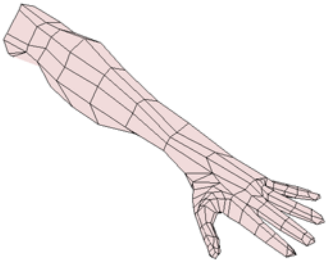


Original object.

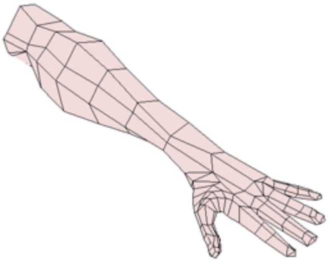
Low **Sharpness** tends to eliminate short edges first.

High **Sharpness** keeps more fine detail, as in the fingers and nails.

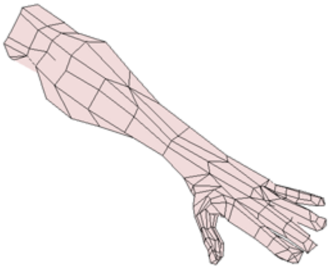
- When you reduce an object, it usually shrinks. **Preserve volume** rearranges vertices to keep the object's volume unchanged. When an edge is collapsed into a point, the difference in the dihedral volume is calculated and the point is moved along its normal to compensate. In some cases, this can help maintain an object's silhouette and prevent protuberances like fingers and limbs from disappearing. However, it can also distort an object's shape, especially at high reduction amounts, so use the slider to achieve a good balance.



Without **Preserve volume**, reduced objects will shrink.



A moderate value preserves the object's silhouette.



Setting the value too high introduces distortions.

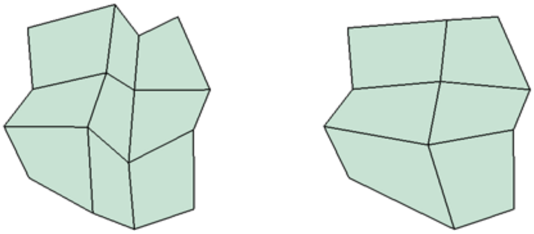
Because **Preserve volume** moves points, the lengths of edges are changed and this affects the order in which subsequent edges are collapsed.

Preserving Quads

When edges are collapsed on quadrilateral polygons, triangles are created. A couple of mutually exclusive options on the General tab of the Polygon Reduction Op property editor give you two different ways to make sure that the final result still contains as many quads as possible.

Reduce parallel edge loops will never create new non-quads, but it might distort certain objects depending on their geometry. **Preserve quad lines** has a slider so you can adjust it to minimize distortions, but at the cost of more non-quads in the result. You should try different settings to see what combination gives the best results in specific cases.

- **Reduce parallel edge loops** collapses an entire set of parallel edge loops together instead of collapsing individual edges one at a time.

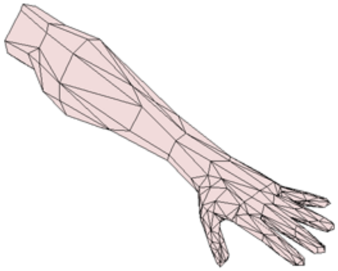


Reduce parallel edge loops
The edge loop with the least average energy is collapsed first.

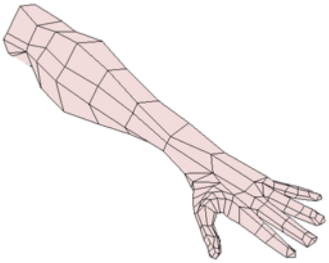
Although **Reduce parallel edge loops** is incompatible with **Enforce symmetry**, it tends to preserve symmetry on truly symmetric objects anyway.

- **Preserve quad lines** provides relative control over the amount of quads remaining in the result. After the edge with the lowest energy is collapsed, edges that were parallel to it become temporarily more likely to be collapsed next (as if their energy has been lowered).

  Use the slider to control the strength of the effect: the higher the value, the more quads in the final result. A value of 0 is equivalent to turning this option off.



Low values for **Preserve quad lines** produce many triangles.



High values for **Preserve quad lines** produce more quads.

Enforcing Symmetry

There are two ways to preserve symmetry during the polygon reduction process: using a symmetry map or not.

- If you don't use a symmetry map, the reduction operator uses its own internal method for determining which edges are symmetric. This allows it to find local symmetries, such as between the two sides of the same finger, in addition to global symmetries such as between the fingers of opposite hands.

  However, if an object has multiple planes of symmetry (like a sphere), the internal method arbitrarily pick one pair out of many possible symmetric edges. In such cases, it may be preferable to use a symmetry map.

- If you connect a symmetry map, the operator uses the map to determine which edges are symmetric. This is useful when the object is not in its base symmetric shape. It also resolves ambiguities when there are multiple planes of symmetry, and ensures that symmetrical edges are always found in spite of possible numerical imprecisions in calculations.

  The symmetry map must already exist before you apply polygon reduction.

In either case, after the operator collapses an edge, it automatically collapses the symmetric edges before proceeding to the edge with the next least energy.

To enforce symmetry without using a symmetry map

1. On the General tab of the Polygon Reduction op property editor, make sure that **Enforce symmetry** is on.

   As long as you don't connect a symmetry map, the reduction operator uses its own internal method for determining which edges are symmetric.

2. If desired, you can adjust **Tolerance level**. The tolerance is the maximum difference in energy between symmetric edges. If a symmetric edge has an energy difference greater than the tolerance, it is not collapsed automatically (although it may still be collapsed later in the process, depending on its own energy).

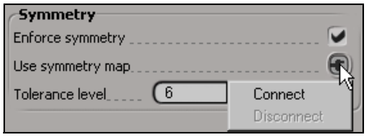
To connect a symmetry map

1. Before you apply polygon reduction, create a symmetry map by selecting the object and choosing **Get** > **Property** > **Symmetry Map** from any toolbar.

   For more information about applying symmetry maps in general, see *Applying Symmetry Maps* on page 117.

2. Apply polygon reduction.

3. Right-click on the **Use symmetry map** connection icon and choose **Connect**.



Symmetry

Enforce symmetry ☑

Use symmetry map

Tolerance level 6

4. In the pop-up explorer that opens, expand the appropriate cluster and select the desired symmetry map. The connection icon appears red to show that a map is connected.

   If you change your mind and want to disconnect the symmetry map or connect a different one, right-click on the **Use symmetry map** connection icon again. The connection icon appears gray when no map is connected.

When you are using a symmetry map, you do not need to turn **Enforce symmetry** on. As long as a symmetry map is connected, it will be used.

Preserving Features

The Feature Preservation tab of the Polygon Reduction Op property editor includes several options that help to preserve specific features during the reduction process.

To preserve edges according to certain criteria

Use the **Preserve** sliders to control the priority given to preserving edges that meet the corresponding criteria:

- **Boundaries** are edges that are shared by only one polygon.
- **Input cluster boundaries** are the boundaries of the polygons or polygon clusters on which you applied reduction (if you did not apply it on an entire object).
- **Hard edges** are edges that have been marked as hard using **Modify** > **Component** > **Mark Hard Edge/Vertex**.
- **Creases** are edges that have had crease values set using **Modify** > **Component** > **Set Edge/Vertex Crease Value**.
- **Material boundaries** are the boundaries of polygon clusters with local materials.
- **Property discontinuities** are breaks in the texture UVs, user normals, or vertex color (CAV) properties. User data is not considered.

Setting a slider to its maximum value preserves the corresponding edges completely, and may prevent the operator from reaching the amount of reduction set on the General tab. Setting a value of 0 means that the criterion is not considered at all when reducing polygons. Turning a checkbox off is equivalent to setting a value of 0.

To preserve specific vertices

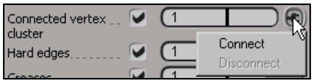
1. Before you apply polygon reduction, create a cluster containing the points you want to preserve.

   The cluster must exist before you apply the Polygon Reduction operator. If you have already applied polygon reduction, then you must disable the operator before creating the cluster. This ensures that the cluster is created at a lower position in the construction history than the reduction operator.

   Similarly if you want to add or remove points from the cluster, make sure you disable the Polygon Reduction operator first.

   For more information about disabling and re-enabling operators in the stack, see *Disabling the Top of the Stack* on page 62.

2. Apply polygon reduction, or re-enable the Polygon Reduction operator if you disabled it.



Connected vertex cluster ☑ 1

Hard edges ☑ 1

3. On the Feature Preservation tab of the Polygon Reduction Op property editor, right-click on the **Connected vertex cluster** connection icon and choose **Connect**.

4. In the pop-up explorer that opens, select the desired cluster. The connection icon appears red to show that a cluster is connected.

5. Use the **Connected vertex cluster** slider to set the priority for preserving the vertices in the connected cluster.

   As with the other properties, setting the slider to its maximum value preserves the corresponding edges completely, and may prevent the operator from reaching the amount of reduction set on the General tab. Setting a value of 0 means that the connected vertices are not considered at all when reducing polygons. Turning the checkbox off is equivalent to setting a value of 0.

   If you change your mind and want to disconnect the cluster or connect a different cluster, right-click on the **Connected vertex cluster** connection icon again. The connection icon appears gray when no cluster is connected.

To optimize the reduction process for certain attributes

Use the **Optimize for attributes** sliders to bias the reduction process by taking into account value differences in other properties such as texture UVs, vertex colors (CAVs), weight maps, and user normals. A value of 0 ignores the attribute when considering an edge for reduction, while non-zero values set a weighted priority which is considered together with the edge's geometric energy.

To use existing vertex positions

Normally when an edge is collapsed, the new vertex is created at the midpoint of the edge. However, if **Use existing vertex positions** (available on the Feature Preservation tab of the Smooth property editor) is active, then edges collapse to one of the two endpoints. This means that each point in the reduced object shares its position with a point in the unreduced object. This could be useful in some situations, for example, if you are exporting several version of the geometry for use with an engine that does not interpolate attributes.

If you are using this option, you probably also want to set **Preserve volume** to 0 on the General tab of the Smooth property editor; otherwise, points will be moved after edges are collapsed and the positions will no longer correspond.

You should also be aware that **Use existing vertex positions** may break symmetry if edges cross the plane of symmetry.

Generating LODs

The LOD Controls tab of the Polygon Reduction Op property editor allows you to generate several versions of an object at different levels of detail. The operator precalculates the geometry for each version and caches the results. This has several advantages:

- It avoids potentially expensive recomputations when the amount of reduction is animated.
- You can transition smoothly between discrete LODs, and avoid texture popping and other artifacts that might occur when edges are collapsed individually.
- You can pregenerate several LODs of the same object for use as game content or for any other purpose.

Activating LODs

To activate LODs

1. On the LOD Controls tab of the Polygon Reduction Op property editor, turn **Use LODs** on.
2. Adjust the **Settings** to control the amount of reduction:
   - **Reduction ratio per LOD** is the percentage of reduction applied successively at each LOD. For example, if this value is 50, the first LOD has a reduction of 50%, the second LOD has a total reduction of 75% (half of the remaining polygons), the third LOD has a total reduction of 87.5%, and so on.
   - **Max ratio** is a clamp that sets the maximum total amount of reduction. This overrides the amount of reduction set on the General tab, and prevents the object from becoming too distorted at high reduction amounts.

Controlling the LOD

There are two ways to control which LOD is currently in use:

- If **Control by LOD Stage - Activate** is off, adjust the **Reduction Amount** on the General tab of the Polygon Reduction Op node. The LOD which is closest to the amount of reduction is used. If **Smooth transitions** is on, reduction amounts that fall between discrete LODs morph between the levels.
- If **Control by LOD Stage - Activate** is on, specify the LOD directly using **LOD Stage**. This is especially useful if you want to export LODs for game content or other purposes.

To control the LOD using the Reduction Amount

1. Activate LODs, and make sure that **Control by LOD Stage - Activate** is off.
2. If desired, turn on **Smooth transitions** on either the General or LOD Controls tab of the Polygon Reduction Op property editor:

   - When it is on, the object morphs smoothly between LODs as the reduction amount varies between levels.
   - When it is off, the geometry abruptly snaps to the closest LOD as the reduction amount changes.

   You can also restrict the interval during which the morphing occurs to a fraction of the interval between LODs using **Transition size**.
3. Adjust the **Reduction Amount** on the General tab.

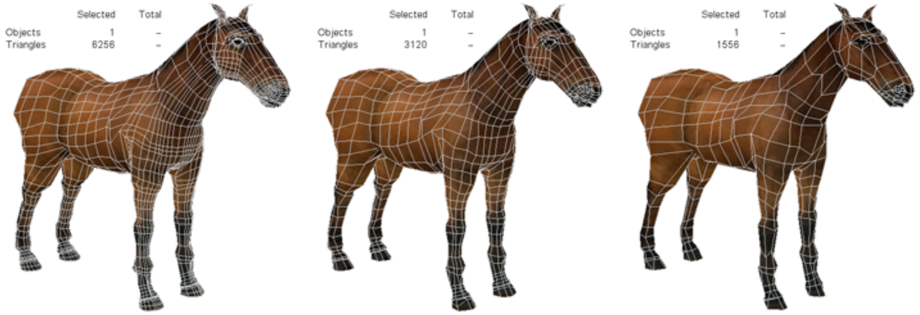
To control the LOD directly

1. Activate LODs, and make sure that **Control by LOD Stage - Activate** is on.
2. Adjust **LOD Stage** to specify the LOD. The meaning of the stages depends on whether **Smooth transitions** is on or off.

   If **Smooth transitions** is off:

   - **0** is no reduction.
   - **1** is the first level of reduction.
   - **2** is the second level of reduction, and so on.

Control by LOD Stage with **Smooth transitions** off



Stage 0
Unreduced geometry.

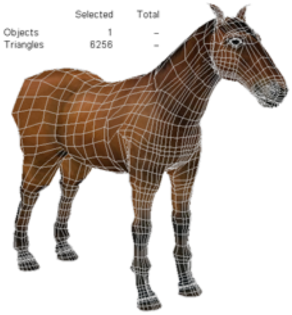
Stage 1
Reduced geometry.

Stage 2
Further reduced geometry.

If **Smooth transitions** is on:

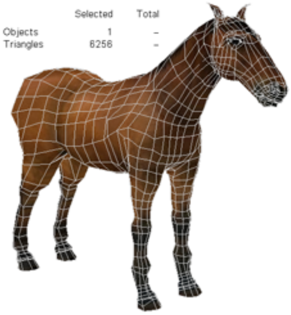
- **0** is no reduction.
- **1** is the unreduced object morphed to first level of reduction. Edges have been shrunk to zero length but coincident points have not been merged. It has the same number of vertices as level 0.
- **2** is the first level of reduction, after coincident points have been merged.
- **3** is the first level of reduction morphed to the second level of reduction, and so on.

Control by LOD Stage with **Smooth transitions** on



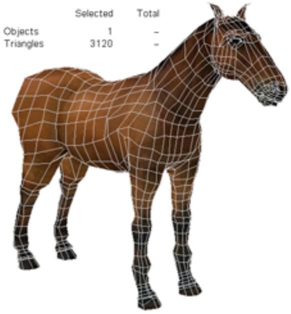
	Selected	Total
Objects	1	-
Triangles	6256	-

Stage 0
Unreduced geometry.



	Selected	Total
Objects	1	-
Triangles	6256	-

Stage 1
Same number of points as stage 0 but morphed to stage 2.



	Selected	Total
Objects	1	-
Triangles	3120	-

Stage 2
Reduced geometry.

Quadrangulating and Triangulating Polygons

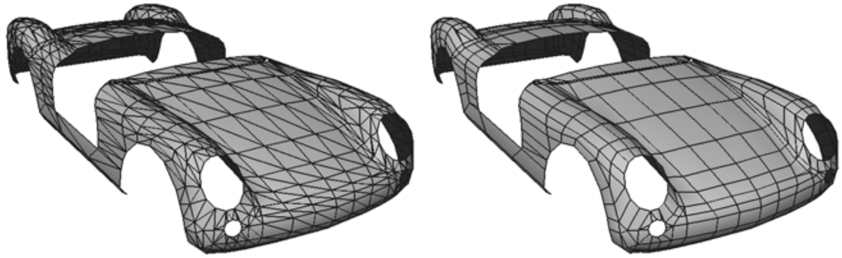
Quadrangulation and triangulation convert between polygons with different numbers of edges:

- Quadrangulation dissolves pairs of adjacent triangles to form quads.
- Triangulation subdivides quads and polygons with five or more sides to form triangles.

You can use these commands when you require polygons with a specific number of sides, for example, when converting between formats for different 3D software or for use in a game engine.

Quadrangulating Polygons

Quadrangulation dissolves pairs of adjacent triangles to form quadrilateral polygons, according to rule-based criteria. This is useful for cleaning up imported quad objects that were converted to triangles. It can also be useful as a first step when turning any arbitrary triangle mesh into quads, although the modified mesh may still contain some triangles.



To quadrangulate polygons

1. Select one or more polygon mesh objects, polygon clusters, or polygons.
2. Choose **Modify** > **Poly. Mesh** > **Quadrangulate**. The Quadrangulate Op property editor opens.
3. Set the parameters as desired:
   - **Incidence Angle** controls the sharpness of edges that can be dissolved. See *Adjusting the Incidence Angle* on page 303.
   - The **Aspect** parameters define several additional tests for eliminating pairs of triangles from consideration. See *Setting the Aspect Tests* on page 303.
   - The **Attributes Preservation** parameters control whether attributes like hard edges are preserved in the modified mesh. See *Preserving Attributes* on page 305.

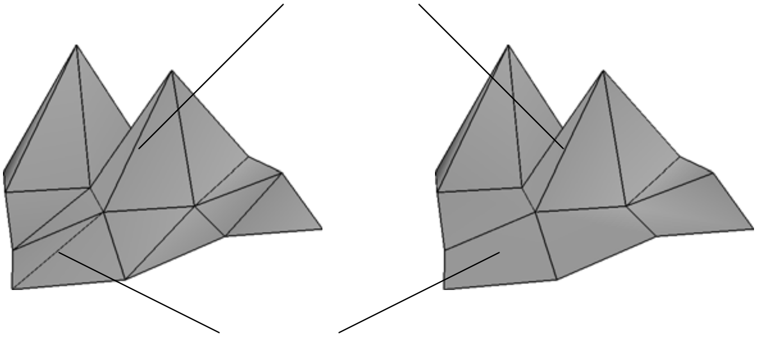
   As you adjust values, **Nb Polygons** indicates the total number of polygons in the modified mesh object. This number depends on the other settings and is displayed for information only.

If you select polygons first, you can quadrangulate different areas of the object using different values.

Adjusting the Incidence Angle

Incidence Angle controls the sharpness of edges that can be dissolved. The value is the maximum difference between the normals of two adjacent triangles at which they are considered as candidates for dissolving.

For example, if this value is 30, then only adjacent triangles with dihedral angles between 30 and -30 degrees are considered. As you increase this value, you start to lose more sharp details.



Edges sharper than **Incidence Angle** are not dissolved.

Flatter edges are dissolved.

Setting the Aspect Tests

The **Aspect** parameters define several additional tests for eliminating pairs of triangles from consideration as candidates for dissolving. First, the polygons are optionally projected onto a plane. Then, when a pair of triangles is considered, the potential resulting quad is tested to see:

- How close are its angles to 90 degrees?
- How well aligned is it with its neighbors?

You can use both of these rules in conjunction to control the quadrangulation. For example, set **Max Alignment Angle** to its maximum value while you gradually increase **Max Corner Right Angle Deviation**; more pairs of triangles get dissolved but the resulting quads may be poorly aligned. Next start decreasing **Max Alignment Angle** to make the alignment rule stricter. Continue to adjust both sliders to get the best possible result.

Setting the Projection Plane for the Aspect Tests

Projection Plane specifies the projection plane used for the tests.

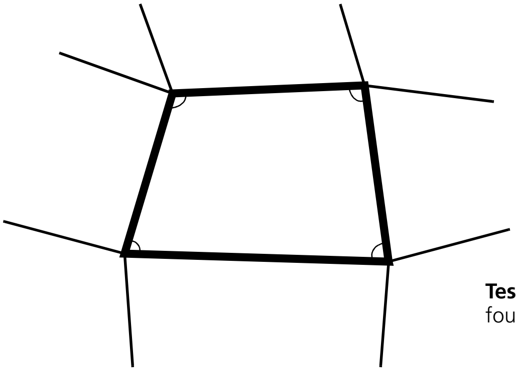
- **Don't Project:** The tests are applied relative to the triangles' own normals.
- **XoY Plane:** The object's XY plane.
- **YoZ Plane:** The object's YZ plane.
- **XoZ Plane:** The object's XZ plane.

For example, if your mesh object is a terrain model that looks like regular triangles in the Top view, set **Projection Plane** to **XoZ Plane**.

Controlling the Rectangularity

Test Corner Angle tests the corners of the resulting quad. If they differ from 90 degrees by more than **Max Corner Right Angle Deviation**, the pair of triangles is eliminated as a candidate for dissolving.

Decreasing **Max Corner Right Angle Deviation** results in more rectangular quads but more leftover triangles.

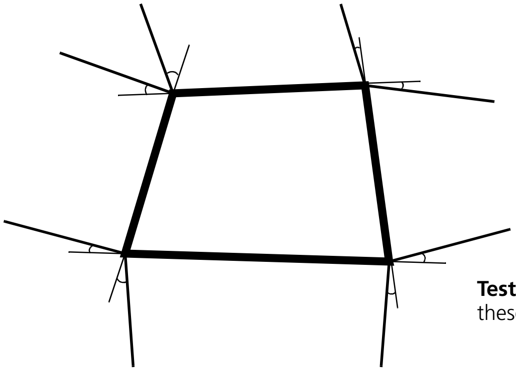


Test Corner Angle tests these four angles.

Controlling the Alignment

Test Quad Alignment tests the alignment of the quad's edges with the edges of its neighboring polygons. If the angle between edges differs from a straight line by more than **Max Alignment Angle**, the pair of triangles is eliminated as a candidate for dissolving.

Decreasing **Max Alignment Angle** results in better aligned quads but more leftover triangles.



Test Quad Alignment tests these eight angles.

Preserving Attributes

You can prevent specific edges from being removed by quadrangulation:

- **Preserve Hard Edge** prevents the removal of edges that have been marked as hard.
- **Preserve Property Seams** prevents the removal of edges that represent discontinuities in attributes such as materials, texture UVs, and so on.

Triangulating Polygons

Triangulation subdivides polygons with four or more edges into triangles. Existing triangles are not affected.

To triangulate polygons

1. Select one or more polygon mesh objects, polygon clusters, or polygons.
2. Choose **Modify** > **Poly. Mesh** > **Triangulate**.

   This command actually applies a SubdividePolygon operator with **Subdivision Type** set to **Triangles**, but it does not automatically open the property editor.

   For more information about the SubdividePolygon operator, see *Subdividing Polygons and Edges* on page 227.

Converting Curves to Polygon Meshes

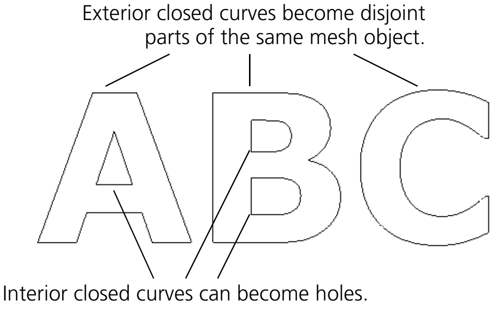
You can convert curves to create complex polygon meshes with holes. The Curve to Mesh Converter is also used to create planar and solid mesh text.

- For information about creating text curves, see *Creating Text* on page 41.
- For information about controlling how text and other curves are converted to polygon meshes, including extruding and beveling, read on.

Before You Start—Preparing the Input Curves

If you understand how the conversion treats the input curves, you'll know what to expect in the resulting polygon mesh.

- You can use multiple curve objects, curve objects with multiple subcurves, or a combination.
- You should have at least one closed exterior curve. Multiple closed exterior curves become disjoint parts of the same mesh object.


Exterior closed curves become disjoint parts of the same mesh object.
Interior closed curves can become holes.

- You can use open, interior curves to define contour (or "flow") lines inside the mesh.
- Closed interior curves can be used either to define holes or as interior contours for the tesselation.
- Curves can intersect or self-intersect. These intersections can be used to define holes or not, depending on the options you choose.
- Instead of using a single closed curve to define the exterior contour, you can use multiple curve objects that enclose an area.

  In this case, **Curve Can Intersect** must be on —see *Setting General Options* on page 307.
- Curves can be non-planar. However, for the purpose of tesselation, the curves are projected onto their average plane and then after the tesselation, depth is added to the resulting mesh based on the height of the original curves.

If the curves are extremely skewed or twisted, they may intersect when projected and the resulting tesselation may not be what you expected— it's easier to visualize the results if the curves are almost coplanar.

Creating Polygon Meshes from Curves

To convert curves to a polygon mesh object

1. Draw, import, or otherwise create one or more curves.
2. Select all the curves.
3. Choose **Create** > **Poly. Mesh** > **Curves to Mesh** from the Model toolbar.

   A polygon mesh object is created and the Curve to Mesh Converter property editor opens. The options on the various tabs of this property editor are described in more detail in the sections that follow.

Managing Input Objects

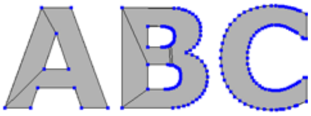
Whenever you generate a new object from other input objects, you can manage the inputs, for example, hide or delete them. See *Managing Generator Inputs* on page 68.

Setting General Options

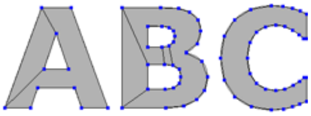
The items in the Step and Options sections of the General tab of the Curve to Mesh Converter property editor control how the input curves are treated.

Step

The **Step** options control how many vertices are created for every knot in the input curves. You can control the values for the exterior **Contours** and the interior **Holes** separately (unless **Curves Can Intersect** is on).



Steps = 3 produces smooth but dense geometry

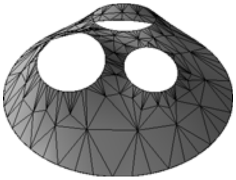


Steps = 1 produces light but blocky geometry

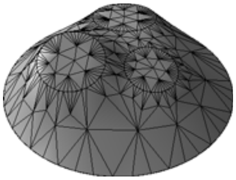
The actually number of vertices in the final geometry can also be affected by **Filter Colinear Curves**. In the illustrations above, that option is on.

Create Holes

The **Create Holes** option determine whether closed interior curves and intersections result in holes. You can turn this option off to use closed interior curves as interior edges for the tesselation; this lets you control the tesselation more directly to meet your final needs, especially if you are using non-planar curves to create a terrain-like effect.

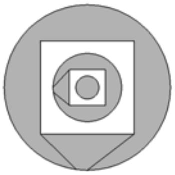


Create Holes on



Create Holes off

This option is recursive; it will create holes within polygon "islands" within other holes, and so on.



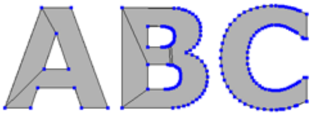
Holes in holes

Filter Colinear/Spike Points

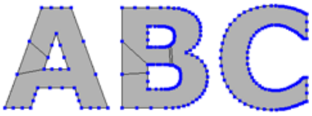
The **Filter Colinear/Spike Points** option can be used to remove points from straight edges to simplify the geometry, as well as to eliminate spiky noise in the input curves. Turn this option on to reduce the geometry. Turn it off if you need the number and distribution of vertices in the polygon mesh to reflect the actual vertices in the input curves, or if you need the extra geometry, for example, to allow the polygon mesh to deform smoothly.

Use **Min. Angle Colinear** to specify which vertices are considered colinear. Any vertex with two edges that meet at an angle greater than this value in degrees is removed.

Similarly, use **Max Angle Spike** to specify which vertices are considered spikes. Any vertex with two edges that meet at an angle less than this value in degrees is removed. This applies to both convexities and concavities.



Filter Colinear Points on

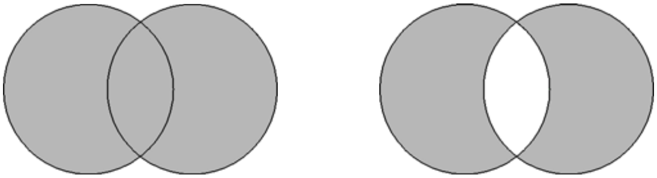


Filter Colinear Points off

Curves Can Intersect

The **Curves Can Intersect** option allows holes to be created in areas bounded by the intersection of two curves. This option may be necessary to provide the correct tesselation on some imported EPS curves. It treats all input curves as if they were one object, so if you turn this on then you cannot create partition clusters as described in *Creating Clusters* on page 319.

In addition when this is on, you cannot set the **Step** value for **Holes**, nor can you use **Proportionally Smaller for Holes** on the Bevel tab. If this option is off and the curves do intersect, you might not get the desired results.



Curves Can Intersect off

Curves Can Intersect on

Offset

Use the **Offset** slider to offset the input curves. Positive values enlarge the curves and negative values shrink them. Values are in Softimage units.

Tessellating

Tesselation is the process of tiling the curves' shapes with polygons. XSI offers three different tesselation methods:

- **Minimum Polygon Count** uses the least number of polygons possible.
- **Delaunay** generates a mesh composed entirely of triangular polygons. This method gives consistent and predictable results, and in particular, it will not give different results if the curves are rotated.
- **Medial Axis** creates concentric contour lines along the medial axes (averages between the input boundary curves), morphing from one boundary shape to the next. This method creates mainly quads with some triangles, so it is well-suited for subdivision surfaces.

Each method has its own set of options, described in the sections that follow.

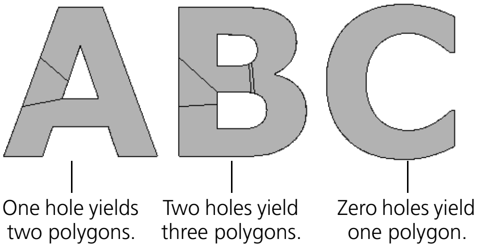
Minimum Polygon Count

The **Minimum Polygon Count** tesselation method uses the least number of polygons possible. For a single contour curve with no holes, this results in exactly one polygon. When holes are present, there are multiple polygons and their shapes are not predictable.

This is because of the fundamental rules for valid polygon meshes in XSI: points cannot be used more than once in a the same polygon and the edges of a polygon must all be connected—you cannot have a hole in a single polygon because the outer and inner boundaries would be disjointed.

In general, Minimum Polygon Count creates irregularly shaped n-sided polygons. For this reason, it is best used when you need to keep the geometry light on simple objects that will not be deformed. Of course, you can always add and remove edges manually.

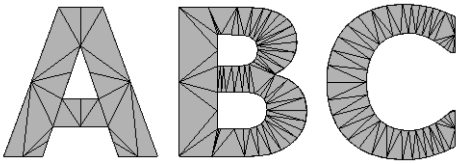
Minimum Polygon Count yields irregular polygons.


One hole yields two polygons.
Two holes yield three polygons.
Zero holes yield one polygon.

Delaunay

The **Delaunay** tesselation method (or more precisely, *constrained Delaunay tesselation*) generates a mesh composed entirely of triangular polygons. This method gives consistent and predictable results, and in particular, it does not give different results if the curves are rotated.

Delaunay yields triangles.

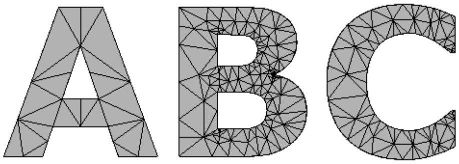


With this method, there are several options for fine-tuning the tesselation further.

To set the Minimum Angle of polygons

- Turn **Minimum Angle** on and then adjust the slider.

  If a triangle contains an angle that is smaller than this value, it gets replaced by better-shaped ones. Eliminating small-angled triangles gives a more uniform OpenGL shading and is more suited for deformations.

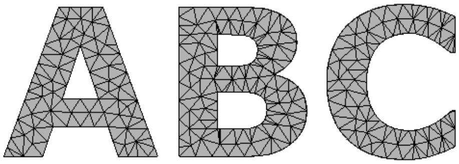


Minimum Angle = 20

To set the Maximum Area of polygons

- Turn **Maximum Area** on and then adjust the slider.

  If a triangle is larger than this value, it gets replaced by smaller ones. This allows the polygon mesh to be deformed more smoothly.



Maximum Area = 10

To limit the number of new vertices

- Turn **Enable** on and then adjust the **Added Vertices Limit** slider.

  This limits the total number of new vertices that can be added by the **Minimum Angle** and **Maximum Area** options.

  Use this option as a precaution against accidentally setting the other options to values that would create huge amounts of geometry with long processing times.

  It is not recommended that you rely on this option to control the final number of vertices because it can force the tesselation to stop before the process is completed, thereby giving an unpredictable combination of polygon shapes and sizes. Look at the **Stats** in the property editor to see whether your object is at or near the limit.

To control tesselation along boundaries

- Set **Boundary Split** to one of these options:

  - **Free**: Allows the boundary edges along the outer contour and inner holes to be split further during tesselation.

This is particularly useful for text and other shapes that may contain straight edges that need to be deformed smoothly.

- **None on Contour Only**: Allows boundary edges along inner holes to be split, but not boundary edges along the outer contour.

  Note that this may affect the uniformity of the mesh if you are using **Minimum Angle** or **Maximum Area**.

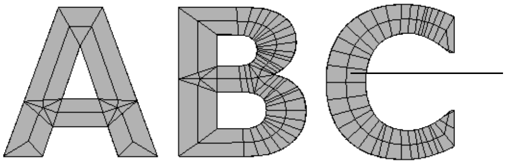
- **None (Contour and Holes)**: Does not allow any boundary to be split.

  Again, this may affect the uniformity of the mesh if you are using **Minimum Angle** or **Maximum Area**.

Medial Axis

The **Medial Axis** tesselation method creates concentric contour lines along the medial axes (averages between the input boundary curves), morphing from one boundary shape to the next. This method creates mainly quads with some triangles, so it is well-suited for subdivision surfaces.

Medial Axis yields quads and triangles.



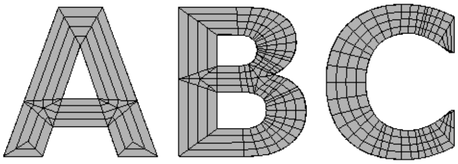
Medial axis runs between opposite boundaries.

With this method, there are several options for fine-tuning the tesselation further.

To control the number of contour lines

- Adjust **Nb. of Loops**.

  If **Adaptive** is off, this is the exact number of medial axes drawn per boundary (rounded to the nearest integer). If Adaptive is on, this is the average number of medial axes drawn and the distance between them is kept fairly constant.



Nb. of Loops = 2

To enhance the accuracy of the medial axes

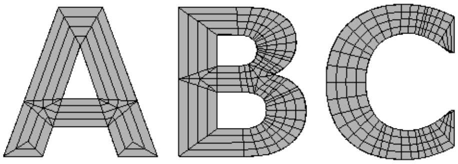
- Turn on **Split edges to enhance**.

  This option adds vertices to allow the contour lines to follow the medial axes more accurately.

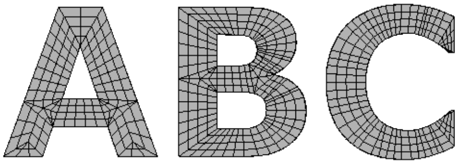
You should turn this option off if you are not creating holes (that is, if **Create Holes** is off on the **General** tab); otherwise, there may be shading artifacts along internal curves.

To control the shape of polygons

- Set **Edge Tesselation** to one of the following values:
  - **None**: Long edges are not split. This results in fewer polygons and lighter geometry, but the resulting long, thin polygons may not deform well.
  - **Equal on both sides**: This results in squarer polygons that deform better.



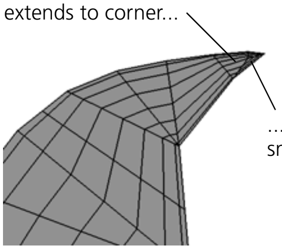
Edge Tesselation set to None.



Edge Tesselation set to Equal on Both Sides.

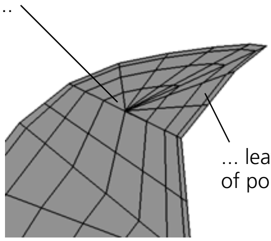
To control the tesselation at extremities

- Adjust **Backtrack Length**.
  - When this value is 0, the medial axis intersects boundaries at each point of concavity, which can often create many small triangles especially in sharp extremities.
  - When this value is positive, the medial axis does not extend completely to the boundary and the remaining area is tesselated with a fan shape.
  - When this value is negative, it creates sharper embossing effects.



Medial axis extends to corner...
... resulting in many small polygons.

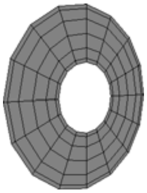
Backtrack Length = 0



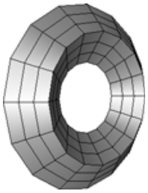
Backtrack Length > 0

To emboss

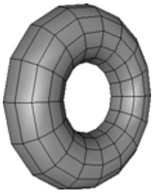
1. Set **Emboss Height**.
   - Positive values raise the central medial axes in the direction of the normals and negative values push the medial axes back.
   - A value of 0 corresponds to no embossing.
2. Set the **Emboss Type** to **Rounded** or **Linear**.



Emboss Height = 0



Emboss Type = Linear

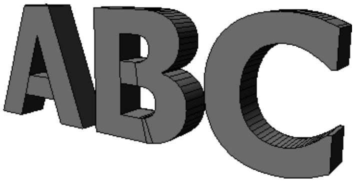


Emboss Type = Rounded

Extruding

You can extrude the tesselated mesh to add depth using the controls on the Extrude tab of the Curve to Mesh Converter property editor.

The only difference between the **Create** > **Text** > **Planar Mesh** and **Create** > **Text** > **Solid Mesh** is the default value of Extrude Length.



To set the length of the extrusion

- Adjust **Length**. For no extrusion, set this value to 0.

To set the number of subdivisions along the extrusion

- Adjust **Subdivisions.** Use lower values to keep your geometry as light as possible or higher values if you need to deform the result smoothly.

To set the direction of the extrusion

- Set **Direction**. **Forward** extrudes in the direction of the normals and **Backward** extrudes in the opposite direction.

To keep different parts of the extrusion

- Turn the **Parts** options on or off individually: **Front**, **Back**, and **Tube**.

Front and **Back**

Tube

Beveling

When converting curves to polygon meshes, beveling extends the contours using a profile.

Controlling the Size and Direction of the Bevel

The **Bevel** options on the Bevel tab of the Curve to Mesh Converter property editor control the size of the bevel:

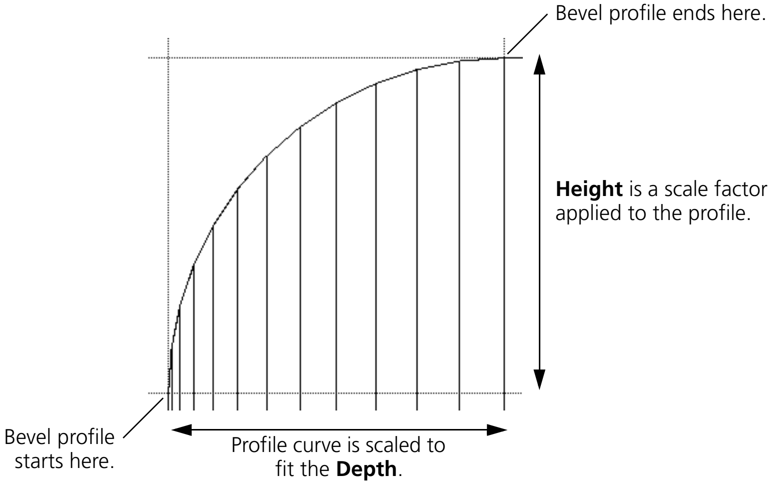
- **Depth** controls how much the bevel protrudes in front and in back of the base mesh.

  The bevel profile is automatically scaled so that the width of its bounding box equals this value. This value is in addition to any extrusion set on the Extrude tab.

- **Height** is a scale factor that modifies the profile curve in the direction orthogonal to the Depth. Positive values push the sides out, while negative values push the sides in.

  If your curves have small, thin features and sharp, closely-spaced corners, then don’t set this value too high.

Side view of a beveled corner


Bevel profile ends here.
Height is a scale factor applied to the profile.
Bevel profile starts here.
Profile curve is scaled to fit the **Depth**.

- **Proportionally Smaller for Holes** in the Options box further scales the bevel profile applied to holes.

  This is useful to avoid overlapping geometry when holes are small compared to the bevel size. This option is not available when **Curves Can Intersect** is on in the General tab.

- **Interior** controls the direction of the bevel.
  - When this option is on, the front and back are beveled outward. As a result, the silhouette matches the original curves but polygons may intersect or interpenetrate in narrow areas when the bevel depth is high.
  - When this option is off, the sides of the tube are extruded outward. In some cases, this may distort the object's outline or cause separate islands of polygons (such as characters in text) collide with each other.

- The options in the Sides box allow you to bevel the **Front** or **Back** individually.

Mitering and Rounding

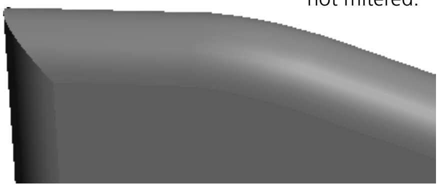
Mitering creates sharp corners, like at the corners of a picture frame.

- **Create Hard Edges** turns mitering on and off.

- **Min Miter Angle** sets the threshold for mitering. Adjacent polygons whose dihedral angle is greater than this value are mitered.

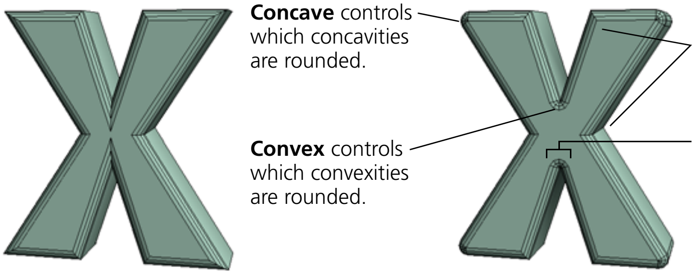
  For best results, set this to the same value as the **Discontinuity Angle** in the object's Geometry Approximation property—this ensures that the discontinuous angles of the bevel are aligned with the discontinuous edges of the extrusion and avoids shading artifacts.

For more information about Geometry Approximation in general, see *Viewing and Rendering Polygon Meshes* on page 196.



Mitering can sometimes produce ugly results with extremely sharp angles. In these situations, you can round the corners instead.

- Use the **Convex** and **Concave** sliders in the Corner Mitering box to define the minimum threshold for mitering convex and concave angles in degrees. Angles that are sharper than these values are rounded. A value of 0 is equivalent to turning rounding off.
- Use **Roundness Subdivisions** to control the smoothness of the rounding. Higher values produce smoother arcs with denser geometry. A value of 0 creates a flattened effect like a sawn-off corner.



Mitering can produce a poor effect on extremely sharp corners.
Concave controls which concavities are rounded.
Convex controls which convexities are rounded.
Flatter angles are not rounded.
Roundness Subdivisions controls the smoothness of the rounding.

Choosing the Bevel Profile

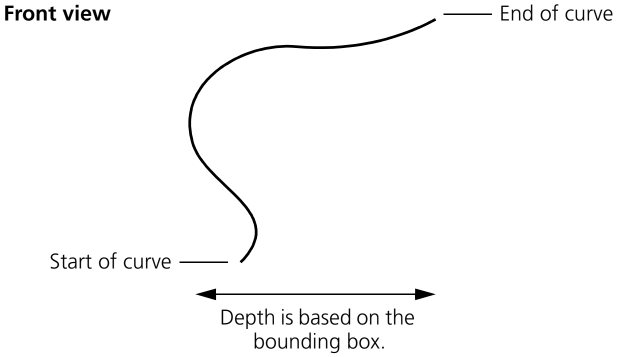
You can choose one of several preset profile curves or draw your own.

To choose a preset profile

1. Click the browse button (**...**) to the right of the **Curve** box on the Bevel tab of the Curve to Mesh Converter property editor.
2. Use the Select File browser to select a preset curve shape.

To use a custom profile

1. In the Front view, draw an open profile curve. The direction in which you draw the curve affects the direction of the bevel.



Connection icon

Unconnected

Connected

2. Click the connection icon (the little plug on the far right of the **Curve** box on the Bevel tab of the Curve to Mesh Converter property editor) and choose **Pick and Connect**.
3. Pick the profile curve. The curve is connected as an input and the connection icon changes.

   A modeling relation exists between the beveled polygon mesh and the custom profile curve. Modifying the profile curve automatically changes the bevel shape.

To reconnect to a different custom profile

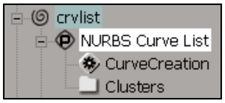
1. Right-click connection icon (on the far right of the **Curve** box on the Bevel tab of the Curve to Mesh Converter property editor) and choose **Pick and Connect**.
2. Pick a different profile curve.

To disconnect a custom profile

- Right-click connection icon (on the far right of the **Curve** box on the Bevel tab of the Curve to Mesh Converter property editor) and choose **Disconnect**.

  The last-used preset profile curve is used.

To save a custom profile curve as a preset


crvlist
NURBS Curve List
CurveCreation
Clusters

1. In an explorer, click the curve's NURBS Curve List icon. An empty property editor opens.
2. Click **Save**. A browser opens.
3. Choose a file name and path for the preset.

It is recommended that you save it in the **Data\DSPresets\2DCurves** folder of your user or workgroup application data path—you can use the **Paths** button to quickly switch to those locations.

Creating Clusters

When you convert a set of curves to polygon mesh, you can create polygon clusters automatically.

Creating Clusters per Island/Character

You can automatically create polygon clusters based on separate characters when converting text, or based on the disconnected islands created by separate outer contour curves when converting curves. This makes it easier, for example, to break up the mesh by selecting clusters and extracting new polygon mesh objects using **Create** > **Poly. Mesh** > **Extract from Polygons**.

If you change the resulting polygon mesh structure by adding, removing, or moving curves after creating island/character clusters, the clusters are not updated automatically. If the number of polygon islands is different, then you must re-create the clusters again.

Note also that you cannot create island/character clusters if **Curves Can Intersect** is on.

To create clusters per island/character

- Choose **Per Island/Character** under **Create Polygon Clusters** on the General tab of the Curve to Mesh Converter property editor.

Creating Clusters Along the Depth

You can automatically create five polygon clusters corresponding to the Front, FrontBevel, Extrusion, BackBevel, and Back. This is useful, for example, if you want to apply different materials to those section.

These clusters are automatically updated even if you change the mesh structure by changing settings, for example, setting a bevel after clicking this button.

The clusters are created even if they are empty. For example, if you do not apply a bevel then FrontBevel and BackBevel contain no polygons.

The clusters span multiple characters or islands. If you explode the mesh after creating these clusters, each mesh object in the exploded hierarchy will have these five clusters.

To create clusters along the depth

- Choose **Per Section Along Depth** under **Create Polygon Clusters** on the General tab of the Curve to Mesh Converter property editor.

Exploding into Separate Polygon Mesh Objects

When converting curves to polygon meshes, you can create a hierarchy of separate polygon mesh objects. There is one new object for each “island” of polygons (or for each character, if using text) and all objects are parented under a common null.

To explode curves into a hierarchy of polygon meshes

- Choose **Explode Mesh** on the General tab of the Curve to Mesh Converter property editor.

  Each island of polygons (or character) is automatically extracted and parented under a new null object. The null object is selected in branch mode.

  A modeling relation exists between the unexploded mesh and the extracted objects, so modifying the options on the Curve to Mesh Converter property editor affects the extracted objects as well.

  However if you change the number of polygon islands, for example, by adding or removing input curves after exploding, you should re-explode the mesh.

Adding & Removing Curves

After you have generated a polygon mesh from a set of curves, you can still add and remove curves without starting over from scratch.

To add a curve using the Curve to Mesh property editor

1. On the General tab of the Curve to Mesh property editor, click **Add Curve**. A picking session starts.
2. Pick a curve. The curve is added to the Curve to Mesh operator and the resulting polygon mesh is re-tesselated according to the current settings.

To add a curve using the Edit menu

1. Select a Curve to Mesh operator using an explorer view.
2. Choose **Edit** > **Operator** > **Add Input**. A picking session starts.
3. Pick a curve. The curve is added to the Curve to Mesh operator and the resulting polygon mesh is re-tesselated according to the current settings.

To remove a curve

1. Select a curve to remove. If you select multiple curves, only the first one will be removed.
2. Choose **Edit** > **Operator** > **Remove Input**. The curve is removed from the Curve to Mesh operator and the resulting polygon mesh is re-tesselated according to the current settings.

The **Remove Input** command does not work if the curve is an input to more than one operator.

Converting NURBS Surfaces to Polygon Meshes

You can create a polygon mesh by converting a NURBS surface or assembled surface mesh. You can choose whether the generated polygon mesh follows the interpolated surface of the NURBS or the hull (that is, whether the vertices are created at the knot locations or at the control point locations).

Trimmed NURBS surfaces are not supported. The trim is not taken into account when converting the surface to a polygon mesh.

To convert a NURBS surface to a polygon mesh

1. Select a NURBS surface or assembled surface mesh.
2. Choose **Create** > **Poly. Mesh** > **NURBS to Mesh** from the Model toolbar. A new polygon mesh object is created and the NURBS to Mesh property editor opens:

   - **From Control Hull** controls the placement of the polygon mesh's vertices.

     When this option is off, vertices are located at the knot positions and along the interpolated NURBS surface.

     When this option is on, vertices are placed at the positions of the surface's control points. If you apply geometry approximation to the resulting mesh to create a subdivision surface, it approximates the original NURBS surface.

   - The **Step** values specify the number of vertices per knot in each of the surface's **U** and **V** directions. The minimum is one vertex per knot.

   - The parameters in the **Stitch** box control whether vertices are merged. If **Active** is on, then vertices that are closer than the specified **Tolerance** are merged.

     This is useful for eliminating superimposed points as may occur at the poles of a surface sphere.

Managing Input Objects

Whenever you generate a new object from other input objects, you can manage the inputs, for example, hide or delete them. See *Managing Generator Inputs* on page 68.

Extracting Polygons

You can extract polygons to create a new object based on selected polygons of an existing object.

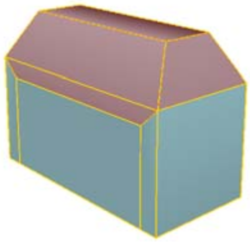
- **Extract Polygons (delete)** creates a new object and removes the selected polygons from the original object.

  Unlike most modeling operations, this does not maintain a modeling relation with the original object—in other words, modifying the original object after you have used this command does not affect the extracted object.

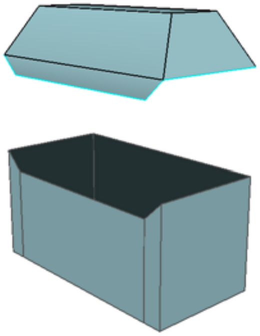
  If you don’t want to create a new object but instead want the separated polygons to remain part of the original object, you should disconnect the polygons instead of extracting them. For more information, see *Disconnecting Polygon Mesh Components* on page 235.

- **Extract Polygons (keep)** creates a new object but leaves the selected polygons on the original object.

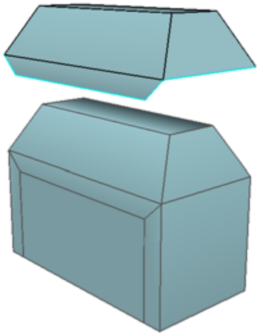
  If you don’t want to create a new object but instead want the duplicated polygons to be added to original object, you should duplicate the polygons instead if extracting them. For more information, see *Extruding, Duplicating, and Insetting Polygon Mesh Components* on page 238.



Original object with polygons selected.



Extract Polygons (delete) removes the original polygons. Here, the new object has been translated up to see the effect.



Extract Polygons (keep) leaves the original polygons intact. Again, the new object has been translated up to see the effect.

Clusters with the following properties are automatically transferred to the generated object: materials, texture UVs, color at vertices maps, property weight maps (but not envelope weights), and shapes. Creases and hard edges and vertices are also propagated.

To extract polygons

1. Select one or more polygons or polygon clusters on a polygon mesh object.
2. Do either one of the following:
   - If you want to delete the polygons from the original object, choose **Create** > **Poly. Mesh** > **Extract Polygons (delete)** on the Model toolbar.
   - If you want to keep the polygons on the original object, choose **Create** > **Poly. Mesh** > **Extract Polygons (keep)** on the Model toolbar.

   New objects are created as follows:

   - If you selected polygons or a single cluster in step 1, a single new object is created, named either **polymsh_extracted** (if you kept the original polygons) or **polymsh_detached** (if you deleted them).
   - If you selected multiple clusters, each cluster creates a separate object named either **polymsh_extracted**, **polymsh_extracted1**, **polymsh_extracted2**, and so on, or **polymsh_detached**, **polymsh_detached1**, **polymsh_detached2**, and so on. These new objects are children of a new null called **polymsh_extracted_null** or **polymsh_detached_null** as appropriate.

     This also occurs if you select a combination of clusters and polygons. The selected polygons are treated as if they were a single additional cluster.

Managing Input Objects

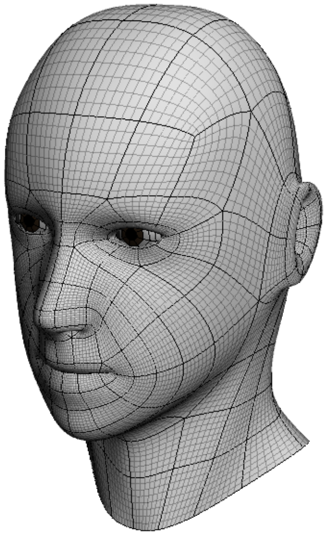
Whenever you generate a new object from other input objects, you can manage the inputs, for example, hide or delete them. See *Managing Generator Inputs* on page 68.

Chapter 9 Subdivision Surfaces

Subdivision Surfaces

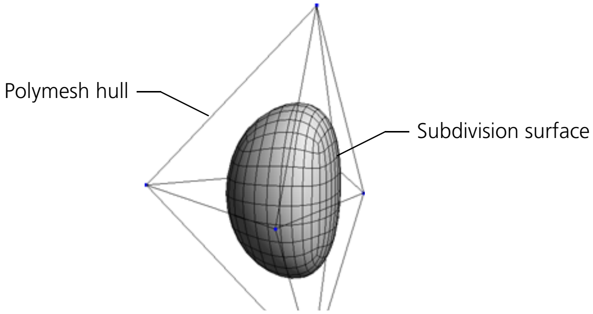
Subdivision surfaces allow you to create a high-resolution polygon mesh object from a low-resolution one and use the low-resolution mesh to control the high-resolution one. You can:

- Use one of three subdivision algorithms.
- Subdivide polygons in one of three ways:
  - Generate a new object based on an existing polygon mesh.
  - Apply a geometry approximation property to an existing polygon mesh.
  - Apply subdivision locally to selected polygons.
- Modify the hardness of edges and points to create creases and spikes.



About Subdivision Surfaces

Subdivision surfaces (sometimes called “subdees”) are a method of creating a high-resolution polygonal mesh from a low-resolution one. You can use the low-resolution mesh as a hull to control the high-resolution mesh.



Subdivision surfaces have numerous advantages. They provide the smoothness of NURBS surfaces with the local detail and texturing capabilities of polygonal meshes.

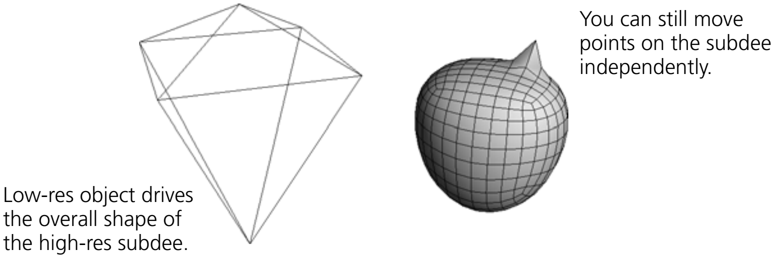
- You can work with the simpler and lighter hull rather than the heavy, denser, high-resolution version for faster interaction.
- You can transform, deform, envelope, and shape-animate the control hull and the effect propagates automatically to the subdivision surface.
- You can even add and remove points, edges, and polygons from the hull to update the surface.

Methods for Subdividing Surfaces

There are three ways to subdivide polygon meshes: generating a new object, modifying the geometry approximation, and locally subdividing selected polygons. You can also combine these methods to obtain the results you want.

Generated Objects

You can create a new high-resolution polygon mesh object from a low-resolution one. As long as there is a modeling relation between the two objects, you can modify and animate the low-resolution object to drive the high-resolution one.


You can still move points on the subdee independently.
Low-res object drives the overall shape of the high-res subdee.

You can hide the high-resolution object while you animate and modify the low-resolution version. You can also turn off the **Render Visibility** of the low-resolution model.

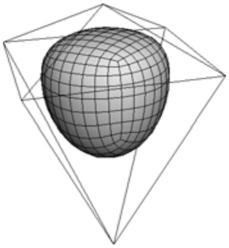
Alternatively, you could place the low-resolution object in a non-rendering layer and the high-resolution version in a non-viewable layer while you work.

In addition, you can apply a Display property so that the low-resolution version is always displayed in wireframe. These procedures are described in detail in the *Fundamentals* guide.

- The advantages of this method is that you can move points and add detail to the generated object yet still drive it with its control hull. You can also create multiple high-resolution objects from the same low-resolution one.
- A disadvantage is that your materials, textures, and vertex colors are not propagated from the original object to the high-resolution one.
- Another disadvantage is that the high-resolution object's geometry can result in heavy scenes and large files.

Geometry Approximation

You can turn a polygon mesh object into a subdivision surface by applying a geometry approximation property. This method does not create a new object; instead, it creates an alternative geometry that is used for rendering.



No new geometry is created.

The original mesh becomes the control cage of the new geometry. The advantage of this method is that materials, textures, and vertex colors are automatically propagated from the control cage to the alternative geometry. In addition, no new geometry is actually created, so scene files can still be quite small.

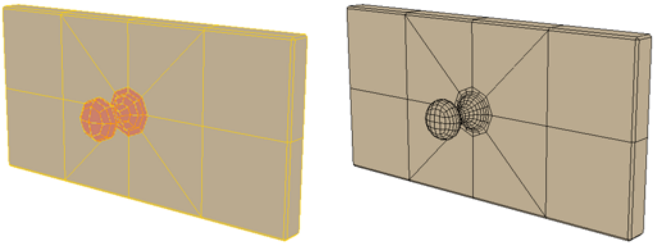
The main disadvantage is that you cannot manipulate individual points, etc., on the high-resolution geometry; only the components that correspond to components on the control cage. However, you can create a duplicate based on geometry approximation that preserves materials, textures, and vertex colors—see *Duplicating Using Geometry Approximation* on page 339.

You can apply the geometry approximation property as an override on a partition in a rendering pass. For more information about using overrides and partitions, see *Chapter 2: Passes & Partitions* in the *Rendering & Compositing* guide.

Local Subdivision

You can add subdivisions locally to selected polygons of a mesh. This method adds an operator to the polygon mesh's operator stack and modifies its topology.

It is useful for adding detail exactly where you want it. Although new geometry is created, you control the amount and the location.



Only selected polygons are subdivided.

With local subdivision, you cannot set crease values.

Combinations

You can combine the methods of subdivision to achieve the control and effect you want. For example, you can start with a very rough polygon mesh as an animation cage. Because it has few points, it is very easy to envelope and manipulate.

From this basic mesh, you can generate a new subdivision surface as a texturing cage. Because it is a new object rather than an alternative representation of the same geometry, you can move points and add local subdivisions where needed. It has more points than the animation cage, so you have more control over texture positioning, but not so many more points to make texturing too difficult.

Finally, you can apply a geometry approximation property to this intermediate object so that you can render a very smooth, seamless object.

Subdivision Rules

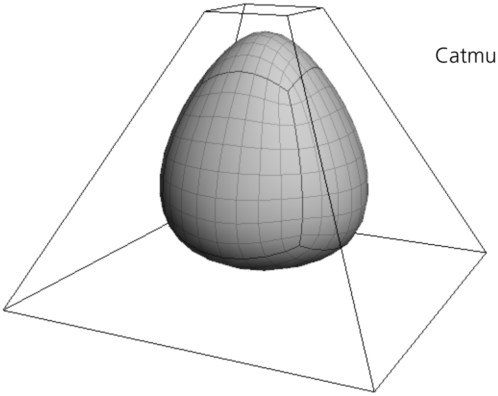
Many different algorithms have been devised for subdividing polygons and averaging vertex positions. SOFTIMAGE|XSI gives you a choice from three of these algorithms: Catmull-Clark, XSI-Doo-Sabin, and linear. The determining factor between choosing one or the other depends on your personal preference and what you are trying to accomplish, although not all rules are available with all operations.

Catmull-Clark

The Catmull-Clark subdivision algorithm produces rounder shapes. The generated polygons are all quadrilateral. The higher the subdivision level, the more this method approximates a bi-cubic standard B-spline surface. The geometry produced is compatible with the method described in “Recursively generated B-spline surfaces on arbitrary topological surfaces” by E. Catmull and J. Clark (*Computer-Aided Design* 10(6):350-355, November 1978).

In areas where the original mesh has quads, the surface has C2 (curvature) continuity; in areas where there are non-quad polygons or vertices without exactly four edges, the surface has C1 (tangential or parametric) continuity.

At level 1, each *n*-sided polygon yields *n* quads; at each additional level every quad yields four more quads.



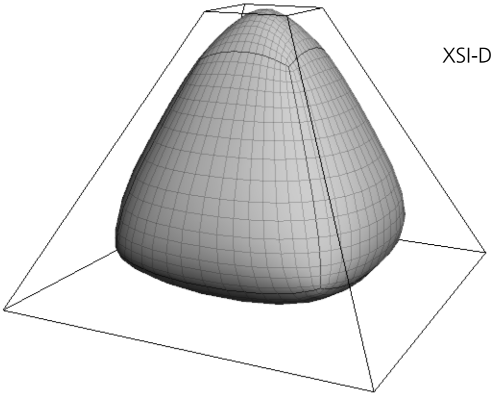
Catmull-Clark Subdivision

XSI-Doo-Sabin

The XSI-Doo-Sabin subdivision algorithm is a variation of the standard Doo-Sabin algorithm. It produces more geometry than Doo-Sabin, but it works better with cluster properties such as texture UVs, vertex colors, and weight maps, as well as with creases.

The higher the subdivision level, the more this method approximates a bi-quadratic uniform B-spline surface.

In areas where the original mesh has quads, the surface has C1 (tangential or parametric) continuity; in areas where there are non-quad polygons or vertices without exactly four edges, the surface has G1 continuity (that is, the tangents have the same direction but not necessarily the same length).

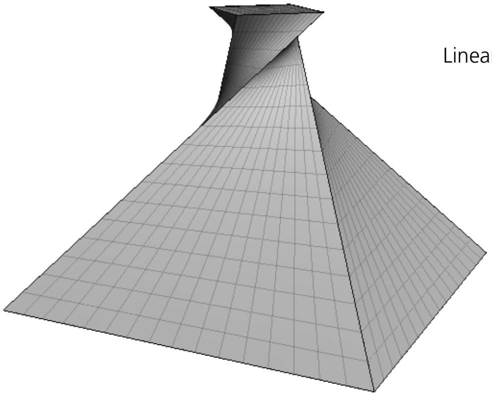


XSI-Doo-Sabin Subdivision

The standard Doo-Sabin method is also available for compatibility with scenes created in XSI version 3.5 and earlier. However, it is recommended that you use XSI-Doo-Sabin for new scenes, and the standard Doo-Sabin method is marked as obsolete.

Linear Subdivision

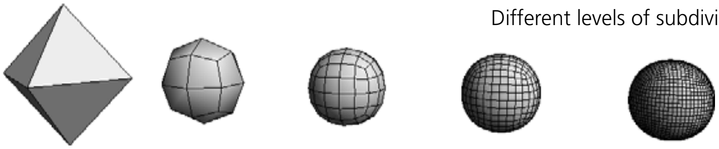
Linear subdivision does not perform any smoothing, so the object's shape is unchanged. It is useful when you want an object to deform smoothly without rounding its contours.



Linear Subdivision

Subdivision Levels

With any subdivision method, you can specify the number of levels. This controls how many times the subdivision process is iterated. Higher levels provide for greater detail and smoothness, but result in heavier geometries and longer processing.



Different levels of subdivision

The subdivision level is animatable, so for example, you can set an expression to control it based on the distance from the camera.

Creases

Subdivision surfaces typically produce a smooth result because the original vertex positions are averaged during the subdivision process. However, you can still create sharp spikes and creases in subdivision surfaces.

This is done by adjusting the hardness value of points or edges. The harder a component, the more strongly it “pulls” on the resulting subdivision surface. Values higher than the subdivision level have no further effect. For details, see *Making Creases and Spikes* on page 340.

Viewing Subdivision Surfaces

There are several visibility options you can set when working with subdivision surfaces. These affect only subdivision surfaces created using the geometry approximation method.

Polymesh Hulls and Subdivision Surfaces

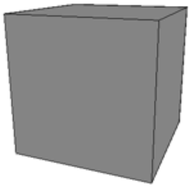
You can display the low-resolution hull and the high-resolution surface independently.

- **Polymesh Hulls** controls the display of the low-resolution hull. Turn this option off to simplify the view, for example, if the hulls make it difficult to see the final subdivision surface.
- **Subdivision Surfaces** controls the display of the final, high-resolution surface itself. Turn this option off to speed up interaction when you do not need to see the end result.

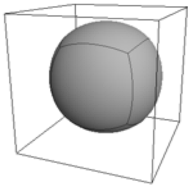
  Boundaries and hard edges are shown directly on the surface if the hull is not displayed and **Boundaries and Hard Edges** are visible in the 3D view.

- If both options are on, then the hull is always shown in wireframe, even in constant, shaded, and textured display types.

Subdivision Surface only



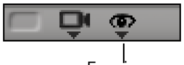
Polymesh Hull only



Both

To toggle display of polymesh hulls and subdivision surfaces

Do one of the following:



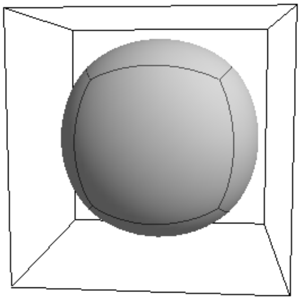
- To affect the display of selected objects in a single 3D view, click the eye icon (Show menu) and toggle **Polymesh Hulls** and/or **Subdivision Surfaces**.
- To affect the display of selected objects in all open 3D views, toggle **Polymesh Hulls** and/or **Subdivision Surfaces** from the View menu on the main menu.
- To affect the display of both selected and unselected objects in a single 3D view, open the Camera Visibility property editor by pressing Shift+s or choosing **Visibility Options** from the eye icon (Show menu), and then set the **Polymesh Hull** and **Subdivision Surface** options on the Attributes tab.

  The left checkboxes affect the display selected objects and the right checkboxes affect the display of unselected objects.
- To affect the display of both selected and unselected objects in all open 3D views, open the Camera Visibility property editor by choosing **View > Visibility Options (All Cameras)** from the main menu and then set the **Polymesh Hull** and **Subdivision Surface** options on the Attributes tab.

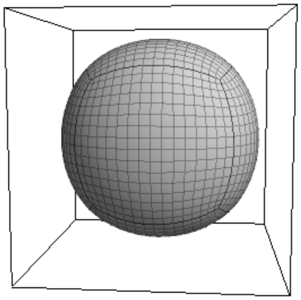
  The left checkboxes affect the display selected objects and the right checkboxes affect the display of unselected objects.
- To quickly toggle the display of polymesh hulls in all open 3D views, press Ctrl+plus (+ key on the numeric keypad).

Show Simplified Subdivision Wireframe

You can choose to display a simplified or a detailed version of the subdivision wireframe. For example, you may prefer to see the detailed version while you are modeling and the simplified version while you are animating.



Show Simplified Subdivision Wireframe on
Only the edges corresponding to the hull polygons are displayed.



Show Simplified Subdivision Wireframe off
All edges are displayed. The edges corresponding to the hull polygons are darker.

To toggle the simplified wireframe

1. Do one of the following:

   - To affect a single 3D view, choose **Display Options** from the Display type menu in the view.

   - To affect all open 3D views, choose **View** > **Display Options (All Cameras)**.

   The Camera Display property editor opens.

2. On the Display Options tab, toggle one or both of the **Show Simplified Subdivision Wireframe** options:

   - The option in the **Wireframe Mode** group controls the display when the 3D view is drawn in Wireframe mode.

   - The option in the **Shaded Mode Wireframe Highlight** group controls the wireframe that is drawn on top of objects in Shaded mode.

To quickly toggle the simplified wireframe for all open 3D views

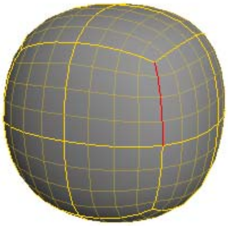
- Press Alt+plus (+ key on the numeric keypad).

  This toggles the **Show Simplified Subdivision Wireframe** option for both **Wireframe Mode** and **Shaded Mode Wireframe Highlight** in all 3D views.

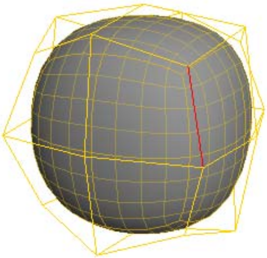
Selecting and Manipulating Components on Subdivision Surfaces

For subdivision surfaces created with the geometry approximation method, you can select and move components directly on the subdivision surface, when both subdivision surfaces are visible and polymesh hulls are not visible.

- You can select those edges that correspond to edges on the hull. Those edges are indicated by thicker lines.

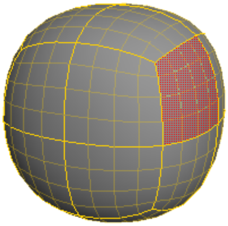


This selected edge on the subdivision surface...

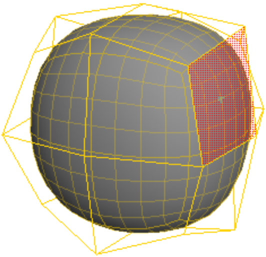


... corresponds to this edge on the polymesh hull.

- You can select the “patches” that correspond to polygons on the hull. These are the areas bounded by thicker lines.

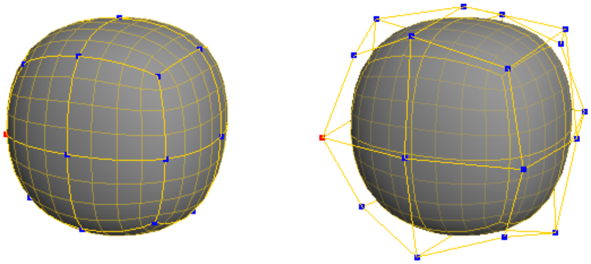


This selected patch on the subdivision surface...



... corresponds to this polygon on the polymesh hull.

- You can select points that correspond to points on the hull. These are the points at the intersections of the thicker lines.



This selected point on the subdivision surface...

... corresponds to this point on the polymesh hull.

Once a component on the subdivision surface is selected, you can perform almost any modeling operation: transform, deform, delete, subdivide, split, bevel, extrude, and so on. This means that you can effectively model without displaying the hull.

Creating Subdivision Surfaces

You can create subdivision surfaces by generating a new object or by applying a geometry approximation property.

You can also duplicate based on geometry approximation and subdivide selected polygons locally. For a discussion about the differences between these methods see *Methods for Subdividing Surfaces* on page 327.

Generating a New Subdivision Surface Object

This method creates a new polygonal mesh object by subdividing and averaging the selected one.

Make sure that immediate mode is off. Otherwise, there will be no modeling relation and the original low-resolution object will not control the high-resolution one.

To generate a subdivision surface object

1. Select a polygon mesh object.
2. Choose **Create** > **Poly. Mesh** > **Subdivision** from the Model toolbar. A new object is created and the Mesh Subdivide With Center property editor opens.
3. Set the options as desired:
   - **Subdivision Level** is described in *Subdivision Levels* on page 332.
   - **Subdivision Rule** is described in *Subdivision Rules* on page 330.

Transferring Attributes

Use the buttons under **Transfer Attributes** in the Mesh Subdivide With Center property editor to transfer clusters and cluster properties—such as materials, texture UVs, vertex colors, weight maps, and shape sources (cluster keys)—from the input object to the subdivided object. For more information, see *Transferring and Merging Attributes* on page 279.

Managing Input Objects

Whenever you generate a new object from other input objects, you can manage the inputs, for example, hide or delete them. See *Managing Generator Inputs* on page 68.

Applying Geometry Approximation

This method does not create a new object; instead, it creates an alternative subdivision geometry that is used for rendering.

To apply subdivisions using geometry approximation

1. Select a polygon mesh object.
2. Do one of the following:

   - If the object has no local geometry approximation property, apply one by choosing **Get** > **Property** > **Geometry Approximation** from any toolbar.

     A default property is applied to the object—this removes any previously applied property or break the link to an inherited property.

     or

   - If the object already has a local geometry approximation property, display its property page by choosing **Edit** > **Properties** > **Viewing Properties** or **Rendering Properties**, then clicking on the Geometry Approximation tab.

   If you are not certain whether an object has a local property, expand its node in an explorer (make sure that **Properties** is active on the Filters menu). Inherited properties are displayed in italic typeface and local properties are displayed in Roman typeface.

3. On the Polygon Mesh page, set the subdivision levels to **1** or more. A value of 0 corresponds to no subdivisions. The subdivision level is described in more detail in *Subdivision Levels* on page 332.

You can also select a polygon mesh (or any of its components) and press the + and – keys on the numeric keypad to increase or decrease the subdivisions. This affects for both the OGL Level and the Render Level.

These keys automatically apply a local geometry approximation property on the object if none exists yet.

Duplicating Using Geometry Approximation

You can duplicate a polygon mesh based on its geometry approximation property. This creates a new high-resolution object that preserves the materials, textures, and vertex colors of the original. Like other forms of duplication, there is no modeling relation between the original and the copy. This works only with Catmull-Clark subdivision.

To duplicate using geometry approximation

1. Select a polygon mesh that has been subdivided using the geometry approximation method.
2. Choose **Edit** > **Duplicate/Instantiate** > **Duplicate Using Geometry Approx.** A new polygon mesh is created based on the geometry approximation settings of the original.

Subdividing Polygons Locally

This method adds an operator to the stack of a polygon mesh object and modifies its topology.

Texture UVs and properties are subdivided linearly with no smoothing.

To subdivide polygons locally

1. Select some polygons on a polygon mesh object.
2. Choose **Modify** > **Poly. Mesh** > **Local Subdivision** from the Model toolbar.
3. Set the options as desired.
   - **Level** is described in *Subdivision Levels* on page 332.
   - **Rule** is described in *Subdivision Rules* on page 330.

Making Creases and Spikes

Subdivision surfaces typically produce a smooth result because the original vertex positions are averaged during the subdivision process. However, you can still create sharp spikes and creases in subdivision surfaces.

To create creases and spikes in subdivision surfaces, you select some points or edges on the hull and then choose **Modify** > **Component** > **Mark Hard Edge/ Vertex** to apply full hardness or **Modify** > **Component** > **Set Edge/Vertex Crease Value** to control the strength of the crease. You can also paint the crease values if desired.

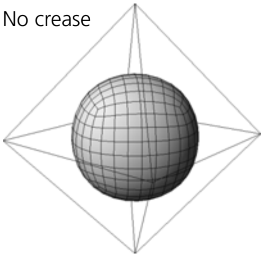
Types of Creases

You can produce different types of creases, depending on whether you apply crease values to points or edges.

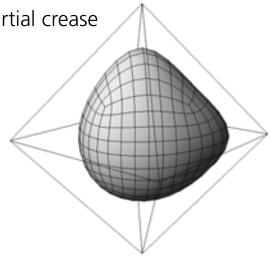
Creases on Points

Using points produces spikes in the subdivision surface:

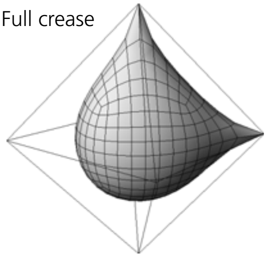
No crease



Partial crease



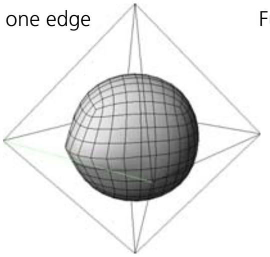
Full crease



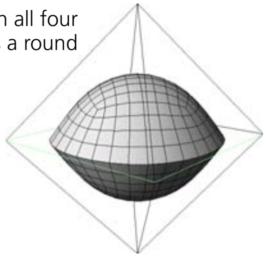
Creases on Edges

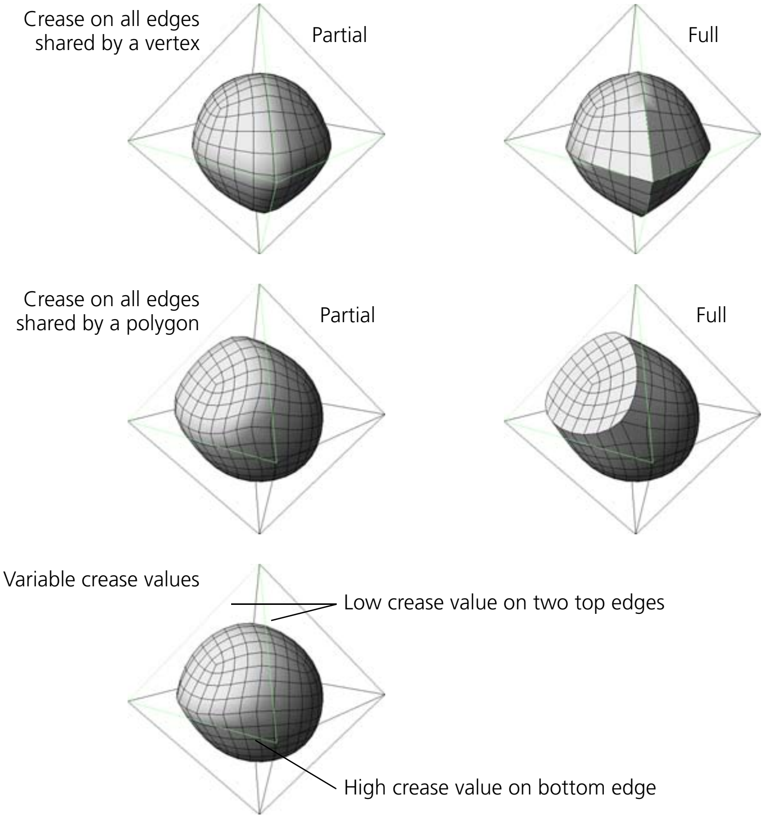
With edges, there are many possibilities for creating creases:

Full crease on one edge



Full crease on all four edges a round




Crease on all edges shared by a vertex
Partial
Full
Crease on all edges shared by a polygon
Partial
Full
Variable crease values
Low crease value on two top edges
High crease value on bottom edge

Marking Hard Edges and Points

When you mark an edge or point as hard, you toggle its crease value to full. There is no gradation of values.

To mark hard edges and points

1. Select some points or edges on the control hull of the subdivision surface.
2. Choose **Modify** > **Component** > **Mark Hard Edge/Vertex**. The subdivision surface is “pulled” toward the selected components and the Mark Hard Edge/Vertex property editor opens.

To temporarily mute the effect, turn off the **Hard** option.

Setting Crease Values on Edges or Points

When you set a crease value on an edge or point, you can specify the amount. The effective range of the crease value is between 0 and the subdivision level.

To set crease values on edges or points

1. Select some points or edges on the control hull of the subdivision surface.
2. Choose **Modify** > **Component** > **Set Edge/Vertex Crease Value**. The Set Edge Crease Value property editor opens.
3. Adjust **Crease Value** to pull the subdivision surface toward the selected components. Values greater than the subdivision level of the surface have no effect.

Painting Crease Values

You can paint on a weight map to modulate crease value as described below.

For more information about weight maps and painting in general, see *Chapter 11: Weight Maps* in the *Fundamentals* guide.

To paint crease values on points

1. Select a polygon mesh.
2. If desired, select some points or a point cluster.
3. Choose **Get** > **Property** > **Weight Map**.

   - If a cluster is selected, a weight map is created and applied to the cluster.
   - If points are selected, a cluster is automatically created and a weight map is applied to it.
   - If the object is selected, a cluster is automatically created containing all its points and a weight map is applied to the cluster.

   The Weight Map property editor opens. You can leave the values at their defaults for now.

4. Re-select the polygon mesh, and select the edges or points on which you want to apply crease values.

5. Choose **Modify** > **Component** > **Set Edge/Vertex Crease Value**. The Set Edge Crease Value property editor opens.

6. Right-click the connection icon to the right of the **Crease Value** slider and choose **Connect**. Expand the node of the appropriate cluster and pick the weight map you created in step 3.

7. The Weight Map and Weight Map Generator pages are added to the property editor. Set the values as desired:

   - **Crease Value** is multiplied by the value from the weight map to give the crease value of a particular point. Crease Value must be greater than 0 to have any effect.

   - The value from the weight map ranges between the **Minimum** and the **Maximum** set on the Weight Map Generator page.

   - The **Base Weight** is the default, unpainted weight for each point in the weight map. It is expressed as a ratio between the Minimum and Maximum values.

   - Values higher than the subdivision level have no further effect.

8. Make sure that **Use Weight Map** is on in the Set Edge Crease Value page.

9. Activate the Paint tool by pressing **w**, then click and drag to paint in the 3D windows. In normal (additive) paint mode:

   - To add weight, use the left mouse button.

   - To remove weight, either use the right mouse button or press Shift while using the left mouse button.

   - To smooth weight values, press Alt while using the left mouse button.

   - To change the brush radius, press **r** while dragging with the left mouse button.

   - To change the opacity, press **e** while dragging with the left mouse button. The opacity controls the amount of weight added or removed with each brush stamp.

   - To change other brush properties, display the Brush Properties property editor by choosing **Get** > **Property** > **Brush Properties** or pressing Ctrl+w.

Tips & Tricks for Subdivision Surfaces

Here are some things to keep in mind when working with subdivision surfaces.

Hull Polygons

In areas where the hull has non-quadrilateral polygons, the subdivision algorithms produce a lower degree of continuity in the final surface.

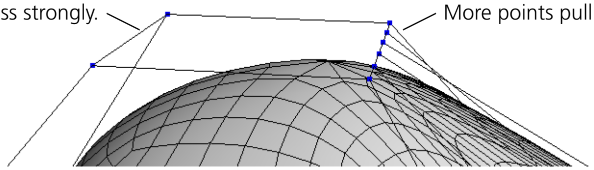
For the smoothest results, follow these rules as much as possible when modeling the hull:

- Use quadrilateral polygons.
- Make sure that vertices are shared by exactly four edges.

Hull Density

Every point on the hull "pulls" the subdivision surface towards itself. If there are areas on the hull with a higher density of points than other areas, they pull the surface and distort its shape.

This could be a good or a bad thing, depending on the result you want to achieve. If it is a problem, try to keep the geometry density well-balanced when modeling the hull.



Fewer points pull less strongly.

More points pull more strongly.

Rendering Subdivision Surfaces

Subdivision surfaces can generate huge numbers of triangles. If you find that your **.mi2** files are too big, display the Render Options property editor by choosing **Render** > **Options** from the Render toolbar and make sure that both **Approximate Polygons** and **Approximate Faces** are off on the Export MI2 tab. The quality of the render does not change.

Section III • NURBS Modeling

Chapter 10 Fundamentals of NURBS Modeling

Fundamentals of NURBS Modeling

NURBS are Non-Uniform Rational B-Splines. They are a class of geometry that is easily manipulated by computers, allowing for a great deal of flexibility in modeling. They let you create smooth curves and surfaces with only a small number of points.

This chapter describes some of the basic underlying concepts behind NURBS modeling, including:

- Degree and continuity
- Points, knots, and segments
- Parameterization
- Multiknots
- Open and closed NURBS
- Geometry approximation on NURBS

The chapters that follow describe how to model NURBS curves and surfaces in XSI.

Introduction to NURBS

NURBS are Non-Uniform Rational B-Splines. They are a class of geometry that is easily manipulated by computers, allowing for a great deal of flexibility in modeling. They let you create smooth curves and surfaces with only a small number of points.

NURBS curves cannot be rendered, but they can be used to build renderable surfaces, define motion paths, control deformations, and so on. Curves are described in more detail in *Chapter 11: Curves* on page 359.

Surfaces are one of the basic types of renderable geometry in SOFTIMAGE|XSI. Surfaces can model precise shapes using less geometry than polygon meshes and they're ideal for smooth, manufactured objects like car and aeroplane bodies. Surfaces are described in more detail in *Chapter 12: Surfaces* on page 385.

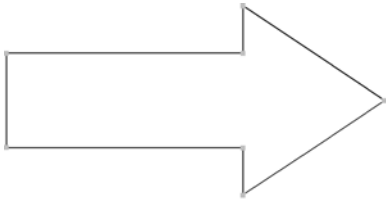
Surfaces are also the building blocks of surface meshes. Surface meshes provide a way to assemble multiple surfaces into a single object that remains seamless under animation and deformation. They are quilts of surfaces joined at their boundaries, just as polygon meshes are quilts of polygons joined at their edges. Surface meshes are described in more detail in *Chapter 13: Surface Meshes* on page 411.

NURBS curves and surfaces share many basic concepts, such as degree, continuity, and parameterization. These concepts are described in the remainder of this chapter.

Degree and Continuity

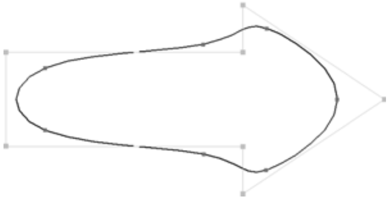
The degree of a NURBS determines its continuity or smoothness. In XSI, NURBS can have one of two degrees:

- **Linear**—straight segments joined at the control points. Linear NURBS have positional continuity (also known as C0, degree 0, or zero-order continuity); this means that although it is connected, it is not necessarily smooth. At least two control points are required to define a linear curve.



Linear NURBS curve

- **Cubic**—smooth NURBS that are interpolated between the control points. Cubic NURBS have curvature continuity (also known as C2, degree 2, or second-order continuity); this means that there are no abrupt changes in the position, tangent, or curvature—in other words, it is very smooth. At least four control points are required to define a cubic curve.



Cubic NURBS curve

Like curves, surfaces can be either cubic or linear. In addition, they can have different degrees in their U and V directions, so that they can be smooth in one direction and jagged in the other.

You can convert between cubic and linear by fitting—see *Fitting Curves onto Curves* on page 371 and *Fitting Surfaces* on page 395.

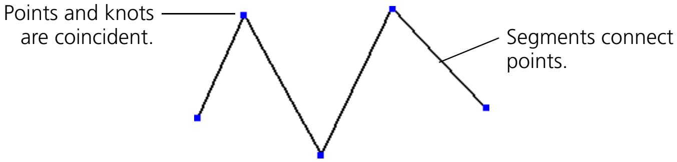
In addition to linear and cubic, SOFTIMAGE|XSI supports NURBS of other degrees imported from other programs.

- You can import quadratic NURBS from SOFTIMAGE|3D.
- Other types of curves and surfaces—Bézier, Cardinal, and B-Spline—are converted to cubic NURBS when imported.
- Clusters are lost during conversion.

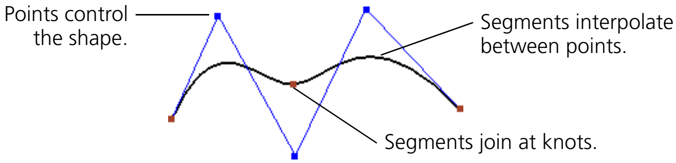
Points, Knots, and Segments

The shape of a NURBS is controlled by its points, sometimes also called *control points* or *control vertices* (*CV*s). The points define *segments* and the segments are joined at *knots*.

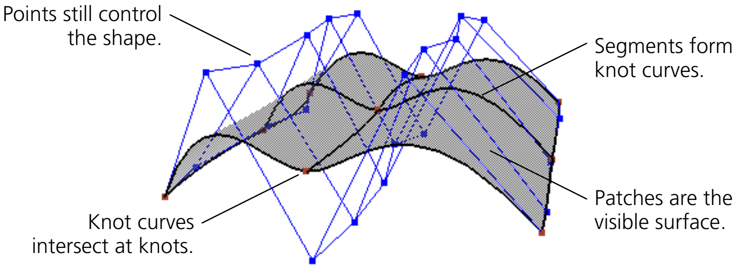
In a linear curve, the segments are straight lines between the control points and the control points and knots overlap exactly.



In a cubic curve, the segments are interpolated between control points and the knots lie on the curve.


Points control the shape.
Segments interpolate between points.
Segments join at knots.

In a surface, the segments form *knot curves* in two directions (knot curves are sometimes called *isoparams* or *isoparms*). The knot curves intersect at the knots, forming four-sided *patches*.

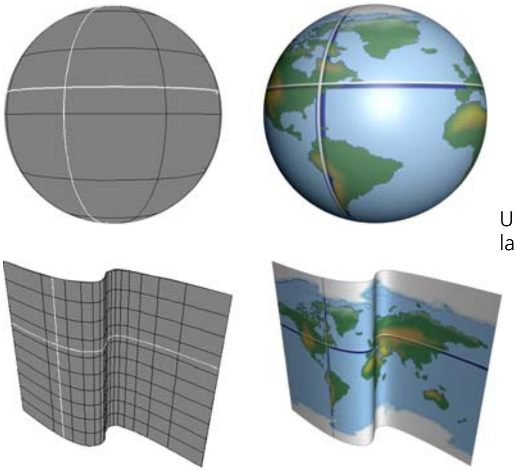

Points still control the shape.
Segments form knot curves.
Patches are the visible surface.
Knot curves intersect at knots.

Parameterization

NURBS are parameterizable, which means that any point along a NURBS can be mathematically defined by one or more parameters.

Curves are defined by a single U parameter along their length; you can describe any point along a curve in terms of its U parameter.

Similarly, surfaces are defined by both U and V parameters; you can describe any point along a curve in terms of its (U, V) coordinates.



U and V are similar to latitude and longitude.

Depending on how a NURBS is parameterized, different values of U and V correspond to different points. You can reparameterize curves as described in *Reparameterizing Curves* on page 382, and surfaces as described in *Reparameterizing Surfaces* on page 401. Surfaces can have one parameterization type along U and a different one along V.

There are four parameterization methods available:

- With **uniform** parameterization, the difference between successive knots is the same, regardless of the actual length of the curve segment.

  Uniform parameterization is especially suitable when merging and lofting.

  While it may seem strange that non-uniform rational B-Splines can have a uniform parameterization, it is not a contradiction; "non-uniform" here means "not necessarily uniform."

- With **non-uniform** parameterization, the difference between successive knots can be anything.

- With **chord length** parameterization, 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. This is particularly useful when using bitmapped textures.

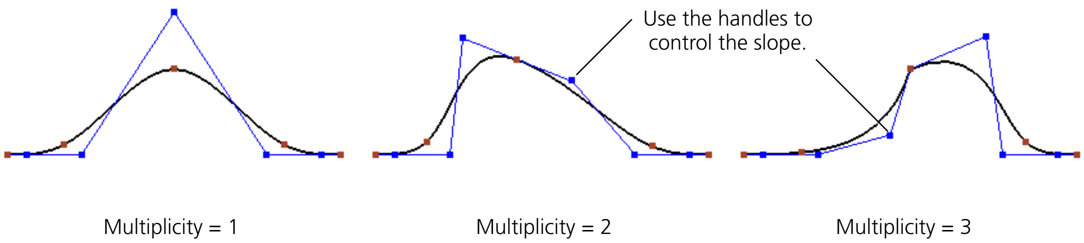
- With **centripetal** parameterization, the knot spacing is related to the square root of the length of each curve segment. In some cases, this may make smoother objects. On surfaces of very unevenly spaced control points, it may give slightly better texture mapping results than chord length.

Normally, when you create a NURBS object, its parameterization is uniform. However, that may change and become non-uniform as you perform modeling operations on it.

Multiknots

Multiknots are knots that behave as if there were two or more overlapping knots. These let you create sharp discontinuities on cubic NURBS. The higher the multiplicity, the sharper you can make the bend. You can create multiknots when you add knots to a curve (see *Adding and Deleting Knots Manually* on page 379) or knot curves to a surface (see *Adding Knot Curves* on page 398).

Multiknots



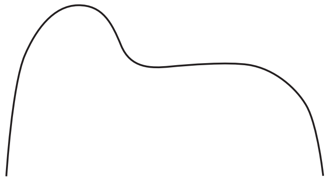
In XSI, the maximum total multiplicity of a knot on a cubic NURBS is 3.

- A knot with a multiplicity of 1 has C2 continuity (curvature).
- A knot with multiplicity 2 has C1 continuity (tangency).
- A knot with multiplicity 3 has C0 continuity (position).

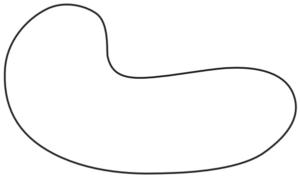
  A knot with multiplicity 3 is like a Bézier point, with one control point exactly at the position of the knot on the curve and the other two control points acting like unlinked handles. You can use them to create curves with sharp points.

Open and Closed NURBS

NURBS can be open or closed. When closed, the minimum and maximum boundaries are joined by segments or patches.

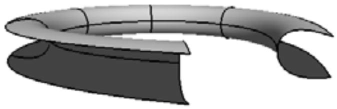


Open curve

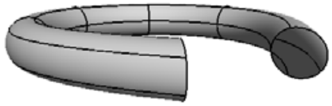


Closed curve

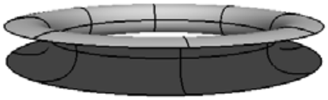
A surface can be open in both U and V like a grid, closed in both like a torus, or open in one and closed in the other like a tube.



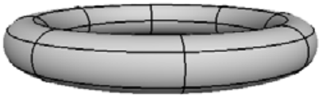
Open in U and V.



Open in U,
closed in V.



Closed in U,
open in V.



Closed in U and V.

Geometry Approximation on NURBS

NURBS curves and surfaces are approximated by straight segments to speed up processing and display. You can control the accuracy using geometry approximation.

Applying and Editing Geometry Approximation

By default, the scene root node has a Geometry Approximation property which is inherited by every object in the scene. You can apply a local Geometry Approximation property to selected objects, hierarchies in branch mode, groups, layers, and partitions. This allows you to control the display and rendering of different objects in different ways.

To apply a local Geometry Approximation property

1. Select an element: object, hierarchy (in branch mode), group, layer, or partition.
2. Choose **Get** > **Property** > **Geometry Approximation** from a toolbar.

To edit an existing Geometry Approximation property

1. Select an element.
2. Display its viewing or rendering properties, then click the Geometry Approximation tab.

Geometry Approximation on NURBS

Geometry Approximation is a multipurpose property that controls the display and rendering of polygon meshes, NURBS curves and surfaces, and objects with displacement maps. There are several tabs:

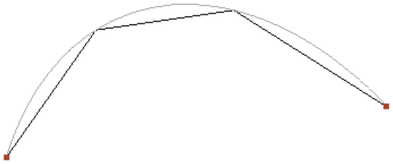
- **Hardware Display** controls how NURBS are displayed in the OpenGL views and is described in the following section.
- **Surface**, **Surface Trim,** and **Displacement** control how various elements are rendered. For more information, see *Setting an Object's Geometry Approximation* in Chapter 3 of the *Rendering & Compositing* guide.
- **Polygon Mesh** controls how polygons and subdivision surfaces are displayed and rendered. For more information, see *Viewing and Rendering Polygon Meshes* on page 196.

Hardware Display

The Hardware Display tab of the Geometric Approximations property editor lets you set the step value for approximating different elements:

- **Surface** objects along their U or V directions.
- **Curve** objects.
- **Surface Curve** elements. These include trims as well as surface curves—see *Projecting and Trimming with Curves* on page 404.

In all cases, the step value determines how many straight segments to use to mathematically approximate the exact shape of each NURBS segment. The higher the step value, the closer the approximation and you get smoother results. However, the trade-off is a longer processing time.



Step = 3

Note that this setting affects only the display; for example, objects on paths will follow the curve as mathematically defined, not as displayed by the path line.

You can also select a curve or surface (or any of its components) and press the + and – keys on the numeric keypad to increase or decrease the steps. These keys automatically apply a local geometry approximation property on the object if none exists yet.

Chapter 11 Curves

Curves

Curves are cubic or linear NURBS. You cannot render them, but you can use them to build renderable surfaces, define motion paths, control deformations, and so on. You can:

- Draw curves directly in the 3D views.
- Create curves from other objects in your scene.
- Modify the curve geometry in a variety of ways.

About Curves

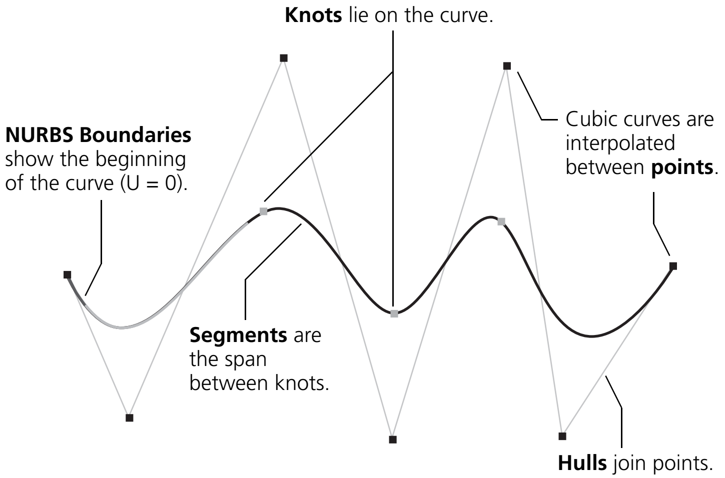
In SOFTIMAGE|XSI, curves are linear (degree 1) or cubic (degree 3) NURBS (Non-Uniform Rational B-Splines). NURBS are a class of curves that computers can easily manipulate, allowing for a great deal of flexibility in modeling.

You cannot render curves because they have length but no width. Instead, you use curves in other ways, for example:

- To build surfaces as described in *Chapter 12: Surfaces* on page 385.
- To define extrusions as described in *Duplicating and Extruding Components Along a Curve* on page 242 and *Chapter 4: Generating Objects from Curves* on page 73.
- To define shapes that can be tessellate into polygon meshes as described in *Converting Curves to Polygon Meshes* on page 306.
- To control deformations as described in *Deforming by Curves* on page 141 and *Spine Deformations* on page 149.
- As paths and trajectories for animation as described in *Chapter 7: Animating along Paths and Trajectories* in the *Animation* guide.

Curve Components and Attributes

Curves have many components and attributes. You can display these components and attributes in the 3D views and select them for various tasks.


Knots lie on the curve.
NURBS Boundaries show the beginning of the curve (U = 0).
Cubic curves are interpolated between **points**.
Segments are the span between knots.
Hulls join points.

Points

Points, sometimes called control points or CVs (control vertices), define the curve mathematically.

In linear curves, the points lie on the curve itself. You need at least two points to define a linear curve.

In cubic curves, the curve is interpolated between the points. You need at least four points to define a cubic curve.

You can modify curves by moving, adding, and deleting points. For more information, see *Adding and Deleting Components on Curves* on page 377.

Knots

Knots are the points at which curve segments meet. Each curve is actually a series of connected segments and every set of four consecutive points defines a segment.

You can add or remove knots as described in *Adding and Deleting Knots Manually* on page 379, but you cannot manipulate knots directly.

Segments

Segments are spans of a curve between consecutive knots. For display purposes, they are subdivided and approximated by several straight line segments. To specify how many straight line segments are used, see *Geometry Approximation on NURBS* on page 356.

NURBS Hulls

NURBS hulls (called simply *lines* in previous versions of Softimage 3D software) are display lines that join consecutive control points. It can be useful to display them when working with curves and surfaces.

NURBS Boundaries

Any point along a curve can be defined in terms of a single parameter U. You can display NURBS Boundaries to show the beginning (U = 0) of a curve. This display option is sometimes also called *boundary flags* or *edge flags*.

Isopoints

Isopoints are arbitrary locations on a curve or surface. You can select them for use with certain modeling operations (such as Extract Segment), but you cannot manipulate them directly.

Subcurves

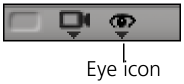
Subcurves are individual NURBS curves that make up curve objects. While primitive curves like a circle are composed of a single subcurve, text and imported EPS curves may be composed of many subcurves.

Viewing Curve Components and Attributes

You can display or hide curve components, like points and hulls, in the 3D views using exactly the same way as other components and attributes. A quick summary is provided here—for complete details about viewing, see *Chapter 5: Viewing 3D Objects* in the *Fundamentals* guide.

To display or hide components in a single 3D view

Do one of the following:



- Click the eye icon (Show menu) and choose the corresponding component option. You can press Shift to keep this menu open while you toggle multiple items.

  or

- Press Shift+s or click the eye icon and choose **Visibility Options**. The Camera Visibility property editor contains many more options than the menu, for both selected and unselected objects.

To display or hide components in all open 3D views

Do one of the following:

- Choose **View** on the main menu and then choose the corresponding component option.

  or

- Choose **View** > **Visibility Options** (**All Cameras**). The Camera Visibility property editor contains many more options than the menu for both selected and unselected objects.

Selecting Curve Components

You select curve components in exactly the same way as anything else in XSI: using the selection filters, tools, and modes available on the Select panel. Selection is fully described in *Chapter 8: Selecting* of the *Fundamentals* guide and is quickly summarized here. In addition, this section describes the **Select Adjacent Points** command as it relates to curve components.

To select components

1. Select the object that owns the components you want to select.

   To select components on multiple objects, select all the associated objects first.

2. Set the selection filter to the desired component type. See *Summary of selection filters* on page 364.

3. If desired, change the selection tool. See *Summary of selection tools* on page 365.

4. Click and drag on the components in a 3D view.

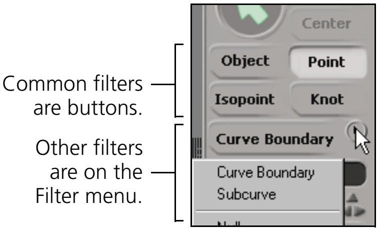
   The specific mouse buttons and modifier keys that you use to perform different actions—like selecting, adding to the selection, toggling, and deselecting—depend on the selection interaction model.

   For more complete details, see *Selecting Components Using the Different Interaction Models* in Chapter 8 of the *Fundamentals* guide.

To select different types of component simultaneously, first select the components of one type, press Shift while changing the selection filter, and then add the other components to the selection.

Summary of selection filters

Curve Component Filters



Component selection filters determine what type of component you can select on the active geometric objects. The most commonly used filters are available as buttons on the Select panel, while more filters are available from the Filter menu below the buttons.

The component selection filters are context-sensitive; they change depending on what type of object is active. The illustration above shows the component filters available when a curve object is selected.

There is an additional filter on the Filter menu that allow you to select border points. This is an example of custom a filter. For information about creating your own filters, see the SDK documentation.

Summary of selection tools

The selection tools determine the mouse interaction used select objects. You can activate a selection tool with the items on the **Select** > **Tools** menu. Alternatively, you can press one of the following keys:

- **F7** for the **Rectangle** selection tool.
- **F8** for the **Lasso** selection tool.
- **F9** for the **Freeform** selection tool.
- **F10** for the **Raycast** selection tool.
- **Shift+F10** for the **Rectangle-Raycast** tool.
- **F11** for the **Paint** selection tool (for curves, this works with points only).

Summary of selection modes

Selection modes are a combination of a filter and tool mapped to a single key for convenience. The only component selection mode that is useful for curve components is **t** for the Rectangle Point.

To select adjacent points

- With one or more points on a curve selected, choose **Select** > **Select Adjacent** > **Points**.

Creating Curves

You can create a curve by:

- Drawing a curve.
- Creating a curve from other objects in your scene. For example, you can extract a portion of another curve, extract a curve from a surface or extract the intersection of two surfaces as a curve.

You can also get curve objects (as described in *Chapter 2: Starting From Scratch: Primitives and Text* on page 35) or import models and scenes with curves from SOFTIMAGE|3D (as described in *Chapter 24: Importing and Exporting* of the *Fundamentals* guide).

Drawing Curves

There are several ways to draw curves:

- By placing the control points or knots explicitly. This method can be used to draw linear or cubic curves. The curve is a weighted interpolation between control points.
- By holding the mouse button down and dragging continuously, as if you were sketching with a pen. When you release the mouse button, XSI creates a cubic curve that approximates what you sketched.

You should always draw curves in a counterclockwise direction. This ensures that the normals of any surface or polygon mesh you create from the curves will be oriented correctly. Otherwise, you may need to invert the curves or generated objects later.

Drawing Curves by Placing Control Points or Knots

You can draw cubic or linear curves by clicking to place control points or to place knots.

- When you draw a cubic curve by placing control points (also known as *control vertices* or *CVs*), the curve does not pass through the locations you click but is a weighted interpolated between the control points. As you add more points, the existing knot positions may change but the point positions do not.
- When you draw a cubic curve by placing knots, the curve always passes through the locations you click. As you add more knots, the positions of the control points are automatically adjusted.
- When you draw a linear curve, the straight segments meet at the locations you click.

For cubic curves, it doesn’t matter whether you place control points or knots; it’s a question of personal preference or drawing style. For linear curves, control points and knots are coincident.

When drawing curves, snapping can be very useful for controlling the position of points. For more information, see *Chapter 18: Snapping* of the *Fundamentals* guide.

You can also activate symmetry when drawing curves; see *Drawing Curves Symmetrically* on page 368.

To draw a curve by placing control points or knots

1. Choose the appropriate command from the **Create** > **Curve** menu on the Model or Animate toolbar:

    - **Draw Cubic by CVs**

    - **Draw Cubic by Knot Points**

    - **Draw Linear**

    The mouse pointer changes to a pen.

2. Click in any 3D view to add the first point, and continue clicking to add more points. Before you release the mouse button, you can drag the mouse to adjust the point's location.

    - To add a point at the end of the curve, use the left mouse button.

    - To add a point between two existing points, use the middle mouse button.

    - To add a point before the first point, first right-click and choose **LMB = Add at Start** and then use the left mouse button. To return to adding points at the end of the curve, first right-click and choose **LMB = Add at End**.

    - Other useful commands are available on the context menu when you right-click: **Open/Close** (see *Opening and Closing Curves* on page 381), **Invert** (see *Inverting Curves* on page 380), **Start New Curve**, and, of course, **Exit Tool**.

    While drawing a curve, you can move any control point by pressing and holding **m** while dragging the point. This activates the Move Point tool in supra mode.

3. When you have finished drawing, exit the curve drawing tool by choosing another tool, pressing Esc, or right-clicking and choosing **Exit Tool**.

    Alternatively, if you want to draw another curve right away, middle-click **Create** > **Curve** on the toolbar or right-click and choose **Start New Curve**.

You may be surprised that a cubic curve is drawn after the second point, even though it takes four points to define a cubic curve segment. This is because points are automatically being created so that there are always at least four.

- When you click to place the first point, four points are added at the same location.
- When you click to place the second point, the three extra points at the first location are removed and three are added at the second location so that the total remains equal to four.
- When you place the third point, the two extra points at the second location are removed and two more are added at the third location.
- After you add the fourth point, there are no extra points.

Drawing Curves Symmetrically

When drawing curves, you can activate symmetry, which automatically creates a second curve object that mirrors the one you draw across the current symmetry plane.

Note that symmetry does not work when sketching curves, nor when adding points to existing curves.

To activate or deactivate symmetry

- Click **Sym** on the Transform panel.

To set the current symmetry plane

Do one of the following:

- Right-click **Sym** and choose **YZ**, **XZ**, or **XY**.

  or

- Right-click **Sym** and choose **Symmetry Properties**, then set S**ymmetry Plane** to the desired value in the Symmetry Options property editor.

Sketching Curves

You can draw a cubic curve by dragging the mouse freehand as if it were a pen.

To draw a cubic curve

1. Choose **Create** > **Curve** > **Sketch** on the Model or Animate toolbar. The mouse pointer changes to a pen.
2. Click in any geometry view and drag the mouse freehand.
3. Release the mouse button to create the curve. The sketch tool remains active and you can continue to draw more curves.

You can use **Create** > **Curve** > **Fit on Curve** to resample a sketched curve. For more information, see *Fitting Curves onto Curves* on page 371.

Creating Curves from Other Objects

You can use other objects in your scene to create curves. For example, you can:

- Extract an arbitrary segment of an existing curve.
- Extract a curve from edges on a polygon mesh object.
- Extract a curve from a surface.
- Extract a curve from subcurves of another curve object.
- Fit a curve onto another curve.
- Create a curve from the intersection of two surfaces.
- Blend two curves; that is, create a third joining curve between them.
- Create a fillet between two intersecting curves.
- Merge two curves; that is, create a curve that spans the originals.
- Generate a curve from the animated translation of an object.

In each case (except the last), a modeling relation exists between the new curve and the original objects used to define it (unless, of course, you are modeling in immediate mode). Modifying the originals in any way also modifies the created curve.

In addition, you can select the created curve and open the property editor of the corresponding operator to adjust its parameters. The modeling relation exists until you freeze the generator operator (the bottom node in the created curve's geometry stack).

If you delete the input objects without freezing the operator stack of the output object first, the output object is removed from the scene. For more information, see *Freezing the Operator Stack* on page 63.

If this happens accidentally, press Ctrl+z to undo.

Managing Input Objects

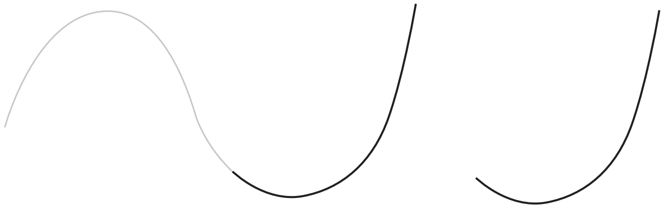
Whenever you generate a new object from other input objects, you can manage the inputs, for example, hide or delete them. See *Managing Generator Inputs* on page 68.

Extracting Curve Segments

To extract an arbitrary segment of an existing curve

1. Select an isopoint on a curve.
2. Choose **Create** > **Curve** > **Extract Segment** from the Model toolbar.
3. Pick another isopoint on the curve.

This creates a new curve and the Extract Curve Segment property editor opens. If desired, use the controls on the Curve Point Adaptor pages to adjust the start and end points.



Extracting Curves from Edges

You can extract a linear curve from edges on a polygon mesh object.

To extract curves from edges

1. Select some edges on polygon mesh objects.
   - Disconnected edges on the same object become separate subcurves of a single curve object.
   - Edges on separate polygon meshes become separate curve objects.
2. Choose **Create** > **Curve** > **Extract from Edges**.

   New curves are created and the Extract Edge Loop property editor opens. By default, there is a single linear span for each edge in the original polygon mesh.
3. Adjust the **Mesh Subdivision Level**. Higher values produce smoother, denser curves, as if the original polygon mesh was subdivided using the Catmull-Clarke rule.

Extracting Curves from Surfaces

To create a curve object based on a curve component of a surface

Extracting a curve from a surface

1. Select one or more isoline, knot curve, boundary, surface (projected), or trim curve on a surface.
2. Choose **Create** > **Curve** > **Extract from Surface** from the Model toolbar.

   If you did not select any curves to extract in step 1, you are prompted to pick a curve now.

   By default, the picking session selects a U isoline with the Rectangle tool, but you can change the filter using the buttons on the Select panel and you can change the tool using the commands on the **Select** > **Tools** menu or the keyboard equivalent (for example, F10 for the Raycast tool).

A new object is created for each selected curve and the Extract Curve from Surface property editor opens. The curves are automatically closed if the surface is closed in the corresponding direction; otherwise the curves are open.

3. If desired, use the controls on the Surface Curve Adaptor page to adjust the extracted curve.

   In the case of assembled surface meshes, the **Extract on Whole Mesh** option lets you extract the curve across multiple subsurfaces. For more information about surface meshes in general, see *Chapter 13: Surface Meshes* on page 411.

Extracting Curves from Subcurves

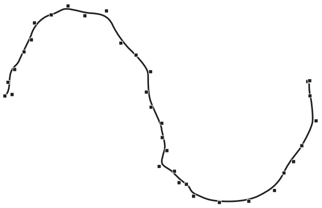
Some curve objects, particularly text and imported EPS curves, contain multiple subcurves. You can extract subcurves to create new curve objects.

To extract subcurves

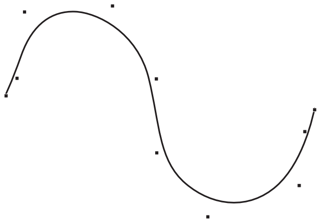
1. Select one or more subcurves, or a cluster of subcurves, on a curve object.
2. Choose **Create** > **Curve** > **Extract from Subcurves** from the Model toolbar.

Fitting Curves onto Curves

You can fit one curve onto another. This is useful for cleaning curves that were drawn with the Sketch tool and for increasing or decreasing the number of points on a curve. You can specify the number of points and the degree of the new curve. The parameterization of the new curve is uniform.



Original sketched curve



New curve fitted onto sketched curve

To fit one curve onto another

1. Select a curve.
2. Choose **Create** > **Curve** > **Fit on Curve** from the **Model** toolbar.

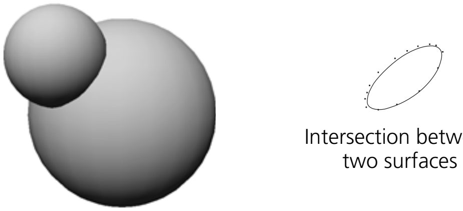
   If you didn't select a curve in step 1, you can pick one now.
3. A new curve is created and the Fit Curve property editor opens. Set values as desired:

   - **Number of Points** controls the number of control points on the created curve and (indirectly) how accurately the new curve follows the original.

- **Continuity** controls the smoothness of the curve. Choose **Position** (C0 or linear), **Tangent** (C1 or quadratic), or **Curvature** (C2 or cubic).
- **Close** does not create a true periodic curve; instead, it superimposes the last point on the first.

Creating Curves from Intersecting Surfaces

You can create a curve that represents the intersection of two surfaces. For best results, first make sure that the surfaces have the same parameterization. For more information, see *Chapter 12: Surfaces* on page 401.



Intersection between two surfaces

To intersect surfaces

1. Select two intersecting surfaces.
2. Choose **Create** > **Curve** > **Intersect Surfaces** from the Model toolbar.

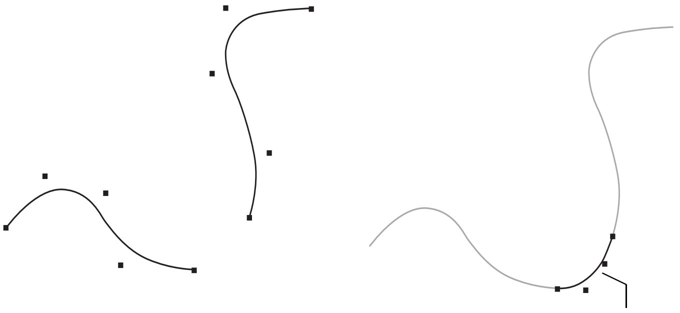
   If you didn't select both surfaces in step 1, you can pick them when prompted.
3. This creates a curve and the Intersect Surfaces property editor opens.
   - If the two surfaces intersect in multiple places, the new curve will be composed of multiple disjoint segments. However, the curve is one single 3D object as shown in the explorer.
   - If the curve cannot be calculated, for example if the two surfaces do not intersect, the result is a unit circle at the global origin.
4. You may need to adjust the tolerance parameters for the curve.

If the intersection curve does not seem right, try increasing the subdivisions of the surfaces.

If you intersect assembled surface meshes with multiple subsurfaces, a single NURBS Curve List object is created with separate subcurves that start and end at the junctions of the subsurfaces.

Blending Curves

Blending two curves creates a third curve that joins the two originals. Unlike Merging, this feature simply joins two existing curves without creating a new curve.



Original curves

New blend curve

To blend curves

1. Select two curve boundaries.
2. Choose **Create** > **Curve** > **Blend** from the Model toolbar.

   - If you didn't pick both curve boundaries in step 1, you can pick them now.
   - To blend between the two boundaries of a single curve, you must choose the command before picking one or both curve boundaries.

3. This creates a third curve and the Blend Curves property editor opens.

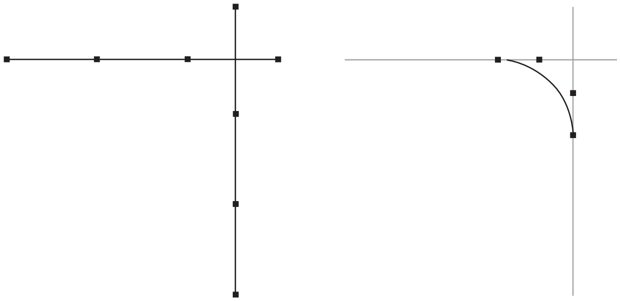
   If necessary, adjust the parameters of the merged curve. For more information, refer to Online Help by clicking on the ? in the property editor.

   If you selected curve objects instead of their boundaries, the blend curve may not be connected at the desired ends. Use the controls in the Curve Point Adaptor pages to adjust the start and end points.

As an alternative to blending curves, you can merge them as described on page 375 or stitch them as described on page 382.

Filleting Curves

You can create a curve that is a fillet, or smooth blending arc, between two curves. Depending on how the curves are drawn and the options you set, the two original curves do not need to intersect.



Intersecting curves

Fillet between them

To fillet curves

1. Select two intersecting curves.
2. Choose **Create** > **Curve** > **Fillet Intersection** from the Model toolbar.

   If you didn't pick both curves in step 1, you can pick them now.
3. An arc is drawn from the beginning of the first curve selected to the end of the second and the Fillet Curves property editor appears.

   If the two curves intersect in multiple places, the new curve will be composed of multiple disjoint segments. However, the curve is one single 3D object as shown in the explorer.
4. Adjust the **Radius** and the **Tolerance**. If no fillet can be created with the specified settings, a unit circle is created at the origin.

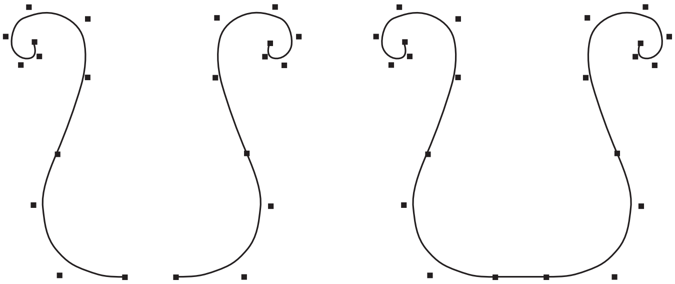
   To change where the fillet is created on the crossing curves, invert one or both curves using **Modify** > **Curve** > **Inverse**.

   For a non-circular fillet, you can move points or otherwise deform the created curve. The modeling relation with the original curve remains, so that if you modify the original curves the deformed fillet changes accordingly.

As an alternative to filleting curves, you can blend curves as described in the previous section or merge them as described in the next section.

Merging Curves

Merging two curves creates a third curve that spans the original two. Unlike Blending, this feature actually creates an extra curve.



Two separate curves Single merged curve

To merge curves

1. Select two curve boundaries.
2. Choose **Create** > **Curve** > **Merge** from the Model toolbar.
   - If you didn't pick both curve boundaries in step 1, you can pick them now.
   - To merge across the two boundaries of a single curve, you must choose the command before picking one or both curve boundaries.
3. This creates a third curve spanning the originals and the Merge Curves property editor opens.

   If necessary, adjust the parameters of the merged curve. For more information, refer to Online Help by clicking on the ? in the property editor.

   If you selected curve objects instead of their boundaries, the merged curve may not span the desired ends. Use the controls in the Curve Point Adaptor pages to adjust the start and end points.

As an alternative to merging curves, you can also stitch them as described on page 382 or blend them as described on page 373.

Creating Curves from Animation

If you have animated the translation of an object, you can plot the motion of its center to generate a curve. For example, this can be used to create a trajectory curve. You can also plot the movement of a selected point or cluster, similar to the Tag2Path command in SOFTIMAGE|3D.

To create a curve from an animation

1. Select an object, point, or cluster.
2. Choose **Tools** > **Plot** > **Curve** from the Animate toolbar. The Plot Curve dialog box opens.
3. Set the options as desired:
   - The **Start Frame** and **End Frame** specify the frames at which to begin and end plotting.
   - The **Step Value** controls the sampling of the generated curve. If this value is 1, a point on the curve is created for every frame; if this value is two, a point is created for every other frame; and so on.
   - **Curve Name** is the name to use for the generated curve object.
4. Click OK. The scene is played back and a curve is created. The original animation remains.

Modifying Curves

You can modify curves in a variety of ways using the commands in the **Modify** > **Curve** menu of the Model toolbar. Each of these commands modifies the topology of a curve. You can:

- Add and remove components like control points and knots.
- Invert the U direction.
- Open and close curves at the U = 0 position.
- Shift the U = 0 position on closed curves.
- Clean curves by resampling them.
- Reparameterize curves.
- Stitch the end of one curve to another.

In addition, any operation that you can perform on points and clusters in general can also be applied to points and clusters on curves: selecting, moving, transforming, and so on.

See *Chapter 5: Manipulating Components* on page 89 for general information about selecting and manipulating various types of components.

Adding and Deleting Components on Curves

There are several ways to add and delete control points and knots on curves:

- You can add both control points and knots interactively. This allows you to control the placement simply by clicking. See *Adding Points and Knots on Curves Interactively* on page 377.
- You can delete control points interactively or manually. See *Deleting Points on Curves* on page 379.
- You can add and delete knots manually. This allows you to control the multiplicity of the knots. See *Adding and Deleting Knots Manually* on page 379.

Adding Points and Knots on Curves Interactively

The procedure for adding control points and knots interactively to an existing curve is similar to drawing a new curve from scratch.

When adding points or knots, snapping can be very useful for controlling the position of points. For more information, see *Chapter 18: Snapping* of the *Fundamentals* guide.

To draw a curve by placing control points or knots

1. Select a curve.

2. Choose the appropriate command from the **Modify** > **Curve** menu on the Model or Animate toolbar:

   - For cubic curves, **Add-Point Tool by CVs** lets you place control points (also known as *control vertices* or *CVs*). The curve does not pass through the points you click but is a weighted interpolation between the points. As you add points, the existing knot positions may change but the point positions do not.

   - For cubic curves, **Add-Point Tool by Knot Points** lets you place knots. The curve always passes through the locations you click. As you add knots, the positions of the control points are automatically adjusted.

   - For linear curves, knots and control points are coincident so the two commands do the same thing.

   The mouse pointer changes to a pen.

3. Click in any 3D view to add points. Before you release the mouse button, you can drag the mouse to adjust the point's location.

   - To add a point at the end of the curve, use the left mouse button.

   - To add a between two existing points, use the middle mouse button.

   - To add a point before the first point, first right-click and choose **LMB = Add at Start**, then use the left mouse button. To return to adding points at the end of the curve, first right-click and choose **LMB = Add at End**.

   - Other useful commands are available on the context menu when you right-click: **Open/Close** (see *Opening and Closing Curves* on page 381), **Invert** (see *Inverting Curves* on page 380), **Start New Curve**, and, of course, **Exit Tool**.

   While adding points or knots, you can move any control point by pressing and holding **m** while dragging the point. This activates the Move Point tool in supra mode.

4. When you have finished, exit the tool by choosing another tool, pressing Esc, or right-clicking and choosing **Exit Tool**.

Deleting Points on Curves

You can delete points on curves manually using the Delete key or interactively using the Delete Point tool.

You cannot remove points from a cubic NURBS curve if less than four points would remain.

You cannot remove points from a linear curve if less than two points would remain.

To delete points on a curve using the Delete key

1. Select a curve.
2. Select one or more points to delete.
3. Press Delete.

To delete points on a curve using the Delete Point tool

1. Select a curve.
2. Choose **Modify** > **Curve** > **Delete Point Tool** from the Model toolbar.
3. Click a point to delete it.

   You can continue to click other points to delete them. When you have finished deleting points, exit the Delete Point tool by choosing another tool or pressing Esc.

Adding and Deleting Knots Manually

You can add or remove knots on a curve manually. This lets you control the multiplicity of the knots. The control points are automatically adjusted.

To add a knot to a curve

1. Do one of the following:
   - To create new knots, select one or more isopoints on a curve.
   - To increase the multiplicity of existing knots, select one or more knots.
2. Choose **Modify** > **Curve** > **Insert Knot**. The Insert Crv Knot Op property editor opens.
3. Set the options as desired:
   - **Multiplicity** controls the continuity at the knot. For more information, see *Multiknots* on page 354.
   - The parameters on the Curve Point Adaptor page let you fine-tune the position of the new knot along the length of the curve.

To remove a knot from a curve

1. Select one or more knots on a curve.
2. Press Delete or choose **Modify** > **Curve** > **Remove Knot**. The selected knot is removed and the Remove Crv Knot Op property editor opens.
3. Set the options as desired:
   - **Multiplicity** specifies the multiplicity to remove.

     If you remove all of a knot's multiplicity, the knot itself is removed.

     If a knot has multiplicity greater than 1 and you remove only part of it, the continuity increases. For more information, see *Multiknots* on page 354.
   - The parameters on the Curve Point Adaptor page let you change which knot to remove.

You cannot remove knots from a curve if less than two knots would remain (the start and end points).

In addition, you cannot remove knots if they are curve boundaries.

Inverting Curves

Inverting a curve reverses its parameterization so that U increases in the opposite direction. The result is as if you had drawn the curve clockwise instead of counterclockwise or vice versa.

For example, if an object uses the curve as a path, it moves in the opposite direction once you invert the curve. Similarly, if a surface has been built from the curve and its operator stack was not frozen, its normals become reversed.

To invert curves

1. Select one or more curves.
2. Choose **Modify** > **Curve** > **Inverse** from the Model toolbar.

   If you didn't select a curve in step 1, you can pick one now.

   The curves are inverted and the Inverse Curve property editor opens.

While drawing a curve or adding points to it, you can also invert it simply by right-clicking and choosing **Invert** from the menu.

See *Drawing Curves by Placing Control Points or Knots* on page 366 and *Adding Points and Knots on Curves Interactively* on page 377.

Opening and Closing Curves

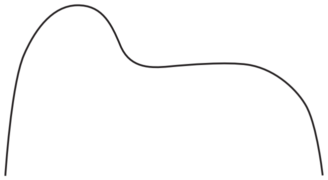
You can open a closed curve and close an open curve. Curves are always opened at the U = 0) position.

To open/close a curve

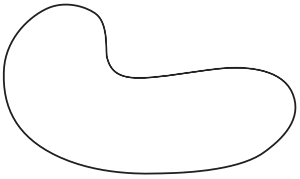
1. Select one or more curves in **Object** mode.
2. Choose **Modify** > **Curve** > **Open/Close**.

   If you didn't select a curve in step 1, you can pick one now.

   The curve is opened if it was closed, and closed if it was open.



Open curve



Closed curve

While drawing a curve or adding points to it, you can also open or close it simply by right-clicking and choosing **Open/Close** from the menu. See *Drawing Curves by Placing Control Points or Knots* on page 366 and *Adding Points and Knots on Curves Interactively* on page 377.

Shifting U on Curves

On closed curves, you can shift the start point (U = 0 position) along the length of the curve.

To shift a start point

1. Select one or more closed curves.
2. Choose **Modify** > **Curve** > **Shift U**.

   If you didn't select a curve in step 1, you can pick one now.

   The Curve Shift property editor opens.
3. Adjust the **Shift U** parameter. As you change values, the start point jumps to the nearest knot.

When shifting, it's helpful to show boundary flags in the 3D views.

Cleaning Curves

Cleaning a curve reduces the number of control points while keeping the same general shape.

To clean a curve

1. Select one or more curves.

2. Choose **Modify** > **Curve** > **Clean** from the Model toolbar.

   If you didn't select a curve in step 1, you can pick one now.

   The Clean Curve property editor opens.

3. Set the **Tolerance**. This controls the maximum difference between the original curve and the resampled shape.

Reparameterizing Curves

Parameterization refers to the way that any point along the length of a curve is described in terms of the parameter U. Any given point along the curve can have different U values under different parameterizations.

When you reparameterize a curve, the new parameterization is in effect at the position of the Reparameterize operator in the operator stack. If you move points or perform other operations on the curve after reparameterizing, the result may have a different parameterization.

For a description of the available parameterization methods, see *Parameterization* on page 352.

Reparameterization does not maintain discontinuities like multiknots.

To reparameterize curves

1. Select one or more curves.

2. Choose **Modify** > **Curve** > **Reparameterize** from the Model toolbar.

   If you didn't select a curve in step 1, you can pick one now.

   The Reparameterize Curve property editor opens.

3. Choose a parameterization method.

Stitching Curves

Stitching two curves glues their ends together. Unlike blending (page 373) or merging (page 375), it does not create a new curve; instead, it deforms the originals.

To stitch a curve

1. Select two curve boundaries.

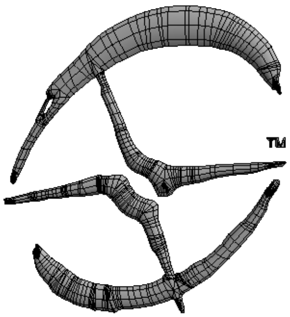
2. Choose **Modify** > **Curve** > **Stitch** from the Model toolbar.

   If you didn't select both curve boundaries in step 1, you can pick them now.

   The Stitch Curves property editor opens.

3. Adjust the parameters to obtain the desired shape. For more information, refer to Online Help by clicking on the ? in the property editor.

Importing EPS Files



You can import curves saved as EPS (encapsulated PostScript) and AI (Adobe Illustrator) files from a drawing program. Once in XSI, you can convert them to polygon meshes as described in *Converting Curves to Polygon Meshes* on page 306 to create planar or extruded logos.

To import EPS and AI files

- Choose **File** > **Import** > **EPS File** from the main menu. A browser opens in which you can specify a path and file name.

  A curve object is created immediately in the scene. The default name is the file name without its extension.

Preparing EPS and AI Files for Import

There are some restrictions on the files you can import. Follow these guidelines:

- Make sure the file contains only curves. Convert text and other elements to outlines.
- Save as version 8 or previous.
- Do not include a TIFF preview header.

Limitations

Imported curve objects typically contain multiple subcurves. Because of this, there are certain limitations.

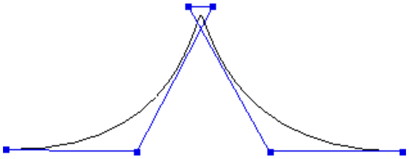
In particular, some of the **Create** commands on the Modeling toolbar do not work with such curve objects as inputs. For example if you extrude such a curve along an axis, only the first subcurve is extruded.

Curve Tips & Tricks

There are a few things to keep in mind when working with curves.

Making Sharp Points

As an alternative to using multiknots, you can create sharp corners in cubic NURBS curves by placing control points so that the hull intersects itself. Be careful not to place the points too far apart or the curve will loop.



Importing Curves from SOFTIMAGE|3D

Cardinal, Bézier, and B-Spline curves are not supported. If you import a scene or model containing these, they are converted to cubic NURBS curves.

Using Weight Maps on Curves

You can apply a weight map on a curve, but you cannot display it or paint on it. You can only use the constant or gradient base map generated by the WeightMapOp operator.

Displaying Information on Curves

Press Shift+Enter or choose **Edit** > **Info Selection** to get useful information about the selected curve object, including its length in Softimage units.

Chapter 12 Surfaces

Surfaces

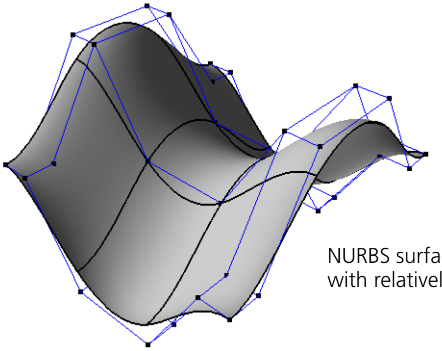
Surfaces are one of the basic types of renderable geometry in SOFTIMAGE|XSI. They are cubic or linear NURBS patches. You can:

- Build surfaces other objects in your scene.
- Modify the topology of surfaces in a variety of ways.
- Project curves onto surfaces and use the projected curves to trim surfaces.

Surfaces also form the basis for building seamless assembled surface meshes, which are described in *Chapter 13: Surface Meshes* on page 411.

About Surfaces

Surfaces are one of the basic types of renderable geometry in SOFTIMAGE|XSI. Surfaces can model precise shapes using less geometry than polygon meshes and they're ideal for smooth, manufactured objects like car and aeroplane bodies.



NURBS surfaces allow for smooth geometry with relatively few control points.

In SOFTIMAGE|XSI, surfaces are NURBS patches. Mathematically, they are an interconnected patchwork of smaller surfaces defined by intersecting NURBS curves.

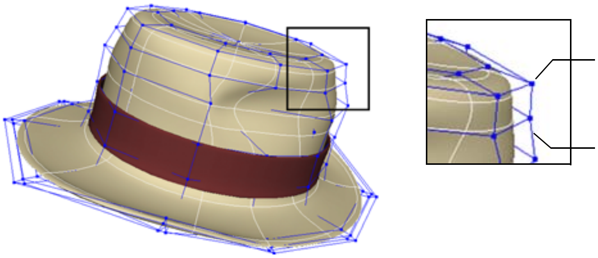
Surfaces are also the building blocks of surface meshes, which are described in *Chapter 13: Surface Meshes* on page 411.

Components of Surfaces

Surfaces have many components and attributes. You can display these components and attributes in the 3D views and select them for various tasks.

Points

Points are the control points of the curves that define the surface.



Points define and control the surface.

You can display lines between points.

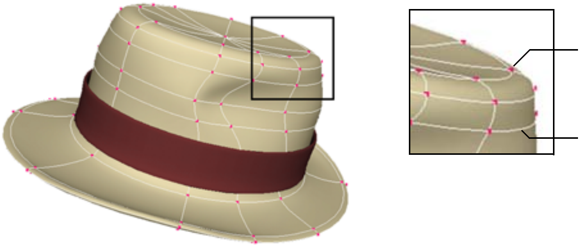
You cannot add or remove points directly, but you can add and remove knots as described in *Adding Knot Curves* on page 398 and *Removing Knot Curves* on page 398—this has the effect of adding or removing points indirectly.

NURBS Hulls

NURBS hulls (called simply *lines* in previous versions of Softimage 3D software) are display lines that join consecutive control points. It can be useful to display them when working with curves and surfaces.

Knots and Knot Curves

Surface knots are the knots of the curves that define the surface; they lie on the surface where the U and V curve segments meet. Knot curves (sometimes called *isoparams* or *isoparms*) are sets of connected knots along U or V—they are the "wires" shown in wireframe views. You can select knot curves and use them, for example, to build other surfaces using the Loft operator. You can add and remove knots as described in *Adding Knot Curves* on page 398 and *Removing Knot Curves* on page 398.



You cannot transform or deform knots directly.

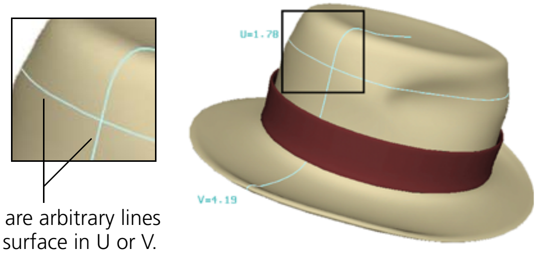
NURBS Boundaries

The minimum and maximum U and V values define the boundaries of a surface. On surfaces that are open in one or both directions, these are the outer edges. You can use the Boundaries selection filter to help you pick boundaries for lofting and other operations.

When working with surfaces, it may be helpful to display NURBS boundaries. This shows the U = 0 boundary in red and the V = 0 boundary in green. This display option is sometimes also called *boundary flags* or *edge flags*.

Isolines

Isolines are not true components. They are, in fact, arbitrary lines of constant U or V on a surface. You can use the **U** and **V Isoline** selection filter to help you pick isolines for lofting and other operations.



Isolines are arbitrary lines on the surface in U or V.

Surface Curves and Trim Curves

Surface curves and trim curves are components that result when you project a curve object onto a surface. For more information, see *Projecting and Trimming with Curves* on page 404.

Samples

Samples are used to adjust textures on objects. You can select them in the 3D views and manipulate them in the texture editor. For more information, see *Editing a Texture on an Object* in Chapter 8 of the *Shaders, Lights & Cameras* guide.

Subsurfaces

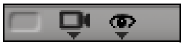
Subsurfaces are the individual NURBS surfaces that are components of assembled surface meshes. For more information, see *Chapter 13: Surface Meshes* on page 411.

Viewing Surface Components

You can display or hide surface components in the 3D views in exactly the same way as other components and attributes. A quick summary is provided here—for complete details about viewing, see *Chapter 5: Viewing 3D Objects* in the *Fundamentals* guide.

To display or hide components in a single 3D view

Do one of the following:



Eye icon

- Click the eye icon (Show menu) and choose the corresponding component option. You can press Shift to keep this menu open while you toggle multiple items.

  or

- Press Shift+s, or click the eye icon and choose **Visibility Options**. The Camera Visibility property editor contains many more options than the menu, for both selected and unselected objects.

To display or hide components in all open 3D views

Do one of the following:

- Choose **View** on the main menu and then choose the corresponding component option.

  or

- Choose **View** > **Visibility Options** (**All Cameras**). The Camera Visibility property editor contains many more options than the menu, for both selected and unselected objects.

Selecting Surface Components

You select surface components in exactly the same way as anything else in XSI: using the selection filters, tools, and modes available on the Select panel. Selection is fully described in *Chapter 8: Selecting* of the *Fundamentals* guide and is quickly summarized here.

In addition, this section describes some special selection commands as they relate to surface components: **Select U/V Rows** and **Select Adjacent Points**.

To select components

1. Select the object that owns the components you want to select.

   To select components on multiple objects, select all the associated objects first.

2. Set the selection filter to the desired component type—see *Summary of selection filters* on page 391.

3. If desired, change the selection tool—see *Summary of selection tools* on page 391.

4. Click and drag on the components in a 3D view.

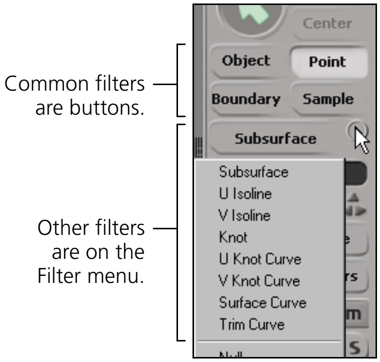
   The specific mouse buttons and modifier keys that you use to perform different actions—like selecting, adding to the selection, toggling, and deselecting—depend on the selection interaction model.

   For complete details, see *Selecting Components Using the Different Interaction Models* in Chapter 8 of the *Fundamentals* guide.

To select different types of component simultaneously, first select the components of one type, press Shift while changing the selection filter, and then add the other components to the selection.

Summary of selection filters

Surface Component Filters



Common filters are buttons.

Other filters are on the Filter menu.

Component selection filters determine what type of component you can select on the active geometric objects. The most commonly used filters are available as buttons on the Select panel, while more filters are available from the Filter menu below the buttons.

The component selection filters are context-sensitive; they change depending on what type of object is active. The illustration above shows the component filters available when a surface object is selected.

There is an additional filter on the Filter menu that allow you to select border points. This is an example of custom a filter. For information about creating your own filters, see the SDK documentation.

Summary of selection tools

The selection tools determine the mouse interaction used select objects. You can activate a selection tool with the items on the **Select** > **Tools** menu. Alternatively, you can press one of the following keys:

- **F7** for the **Rectangle** selection tool.
- **F8** for the **Lasso** selection tool.
- **F9** for the **Freeform** selection tool.
- **F10** for the **Raycast** selection tool.
- **Shift+F10** for the **Rectangle-Raycast** tool.
- **F11** for the **Paint** selection tool (for surfaces, this works with points only).

Selection modes

Selection modes are a combination of a filter and tool mapped to a single key for convenience. The only component selection mode that is useful for surface components is **t** for the Rectangle Point.

To select adjacent points

- With one or more points on a surface selected, choose **Select** > **Select Adjacent** > **Points**.

Selecting Rows of Points in U or V

You can use the loop selection feature of the selection tools to select rows of control points or knot points in U or V on NURBS surfaces.

The procedures are described below. For more in formation about the loop selection feature in general, see *Selecting Ranges and Loops of Components* in Chapter 8 of the *Fundamentals* guide.

There are different procedures depending on the component selection model you have chosen. For more information about component selection model, see *Selecting Components Using the Different Interaction Models* in Chapter 8 of the *Fundamentals* guide.

- The U and V rows stop at the boundaries between subsurfaces on assembled surface meshes.
- You cannot use the Paint selection tool with the loop selection.

If Extended Component Selection is off

If both **Select** > **Extended Component Selection** and **Softimage|3D Selection Model** are off:

1. Do one of the following:
   - Select one point, then Alt+middle-click another point on the same row.

     or

   - Alt+middle-click to select two points on the same row with a single mouse movement.
2. Repeat to modify the selection. Use any of the following additional modifier keys:
   - Press Shift to add to the selection.
   - Press Ctrl to toggle-select.
   - Press Ctrl+Shift to deselect.

If Extended Component Selection is on

If either or both of **Select** > **Extended Component Selection** and **Softimage|3D Selection Model** are on:

1. Do one of the following:

   - Select one point, then Ctrl+click another point on the same row.

     or

   - Ctrl+click to select two points on the same row with a single mouse movement.

2. Repeat to modify the selection. Use the following mouse buttons:

   - Use the left mouse button to add to the selection.

   - Use the middle mouse button to deselect.

   - Use the right mouse button to toggle-select.

Building Surfaces

You can create surfaces from other surfaces and objects in your scene in several different ways:

- Fitting a new surface to an existing one to resample the original or convert between cubic and linear.
- Blending two surfaces to create a third surface that fills the gap between them.
- Filleting to smooth the intersection of two surfaces.
- Merging to create a third surface that spans two others.

In addition to the techniques described here, you can also:

- Build surfaces from curves, as described in *Chapter 4: Generating Objects from Curves* on page 73.
- Get primitive surfaces, as described in *Getting Primitives* on page 37.
- Import models and scenes with surfaces from SOFTIMAGE|3D, as described in *Chapter 24: Importing and Exporting* of the *Fundamentals* guide.

About Building Surfaces

This section includes some basic information that is common to all or most of the operators described in the sections that follow.

Controlling Subdivisions

Most of the operators that generate surfaces use a standard set of controls to determine the subdivisions (knot curves) of the result.

- If **Subdivision Type** is **Per Span**, the basic subdivisions are based on the "natural" subdivisions as determined by the shapes of the inputs. You can subdivide each segment further using the **U** and **V span** sliders.

  In this case, discontinuities resulting from multiknots in the inputs are preserved in the generated objects.
- If **Subdivision Type** is **Absolute**, you can use the **U** and **V abs** sliders to set the exact number of subdivisions in the resulting object.

  However, the result is automatically reparameterized and discontinuities are not preserved.

Managing Input Objects

Whenever you generate a new object from other input objects, you can manage the inputs, for example, hide or delete them. See *Managing Generator Inputs* on page 68.

Fitting Surfaces

You can create a new surface by fitting it to an existing surface. You can specify the degree and the subdivisions of the new surface. This lets you resample a surface or convert between cubic and linear. The parameterization of the new surface is uniform.

To fit a surface

1. Select a surface.
2. Choose **Create** > **Surf. Mesh** > **Fit**.

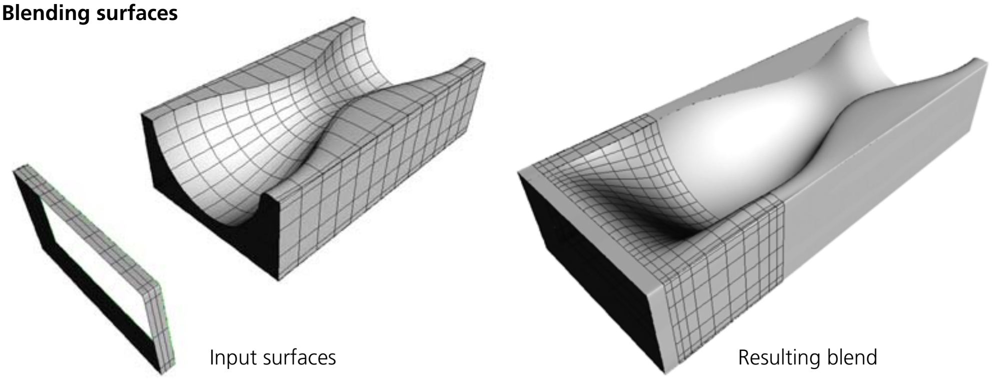
   If you didn't pick a surface in step 1, you can pick one now. This creates a new surface and the Fit Surface property editor opens.
3. Set the options as desired. For more information, refer to Online Help by clicking on the ? in the property editor.

Blending Surfaces

Blending creates a new surface that fills the gap between boundaries on two other surfaces.

The created surface is always cubic in the blend direction. It can be linear or cubic in the other direction, depending on the degree of the boundaries; if both boundaries are linear, then the surface is linear in that direction, other wise it is cubic. If desired, you can use Fit to create a new surface with different degrees as described in *Fitting Surfaces* on page 395.

Blending surfaces



To blend surfaces

1. Select a boundary on each of two surfaces.
2. Choose **Create** > **Surf. Mesh** > **Blend** from the Model toolbar.

   If you didn't select the curves in step 1, you can pick them now. A blend is created and the Blend Surfaces property editor opens.
3. Set the options as desired:

   - Use the options on the **Subdivision** page to specify the resolution in U and V.

- Use the options on the **Shape** page to adjust the overall shape of the blend. For more information, refer to Online Help by clicking on the ? in the property editor.
- If it is not possible to create a surface with the selected options, a unit sphere is created at the global origin by default.

Filleting Intersections

A fillet is a surface that smooths the intersection of two others, like a molding between a wall and a ceiling.

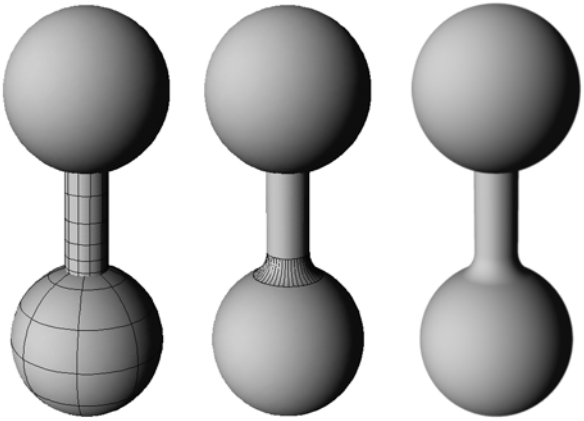
To fillet an intersection

1. Select two intersecting surfaces.
2. Choose **Create** > **Surf. Mesh** > **Fillet Intersection** from the Model toolbar.

   If you didn't select two surfaces, you can pick them now. A fillet is created and the Fillet Intersecting Surfaces property editor opens.
3. Adjust the options as desired:
   - Specify the number of **Subdivisions** in **U** and **V**.
   - Specify the **Radius** and **Radius Type** of the fillet. The fillet can have a **Constant** radius or one that interpolates from a **Start** to an **End** value in a **Linear** or **Cubic** way.

     If you set the radius value too high, it may be impossible to fit the fillet on the input surfaces; if this happens, the result is a unit sphere at the global origin.

Filleting surfaces



Input surfaces   Resulting fillet   Shaded view

If you fillet assembled surface meshes with multiple subsurfaces, a single NURBS Surface Mesh object is created with separate patches that begin and end at the junctions of the subsurfaces.

Merging Surfaces

Merging two surfaces creates a third surface that spans the originals. You have the option of specifying an intermediary curve for the merged surface to pass through. For best results, first reparameterize the surfaces uniformly as described on page 401.

The degree depends on the inputs. If both objects are linear in one direction, then the merged surface is also linear in that direction, otherwise it is cubic. If desired, you can use Fit to create a new surface with different degrees as described in *Fitting Surfaces* on page 395.

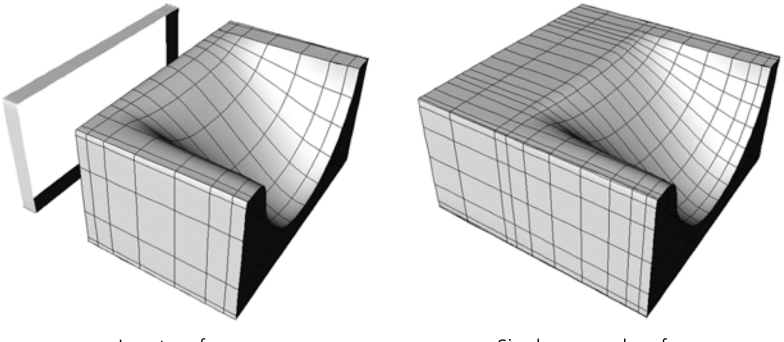
To merge two surfaces

1. Select two surface boundaries then (optionally) add an intermediary curve to the selection.
2. Choose **Create** > **Surf. Mesh** > **Merge** from the Model toolbar.
   - If you did not select two surface boundaries and an intermediary curve, you can pick them now.
   - Right-click to terminate the picking session without choosing an intermediary curve.

   A merged surface is created and the Merge Surfaces property editor opens.
3. Set options as desired:
   - Use the options on the **Subdivision** page to specify the number of knot curves in the result.
   - Use the options on the **Shape** page to adjust the overall shape of the blend. In particular, if you selected a curve to pass through, set **Seam** to **Curve**.

     For more information, refer to Online Help by clicking on the ? in the property editor.

Merging surfaces



Input surfaces

Single merged surface

Modifying Surfaces

You can modify surfaces in a variety of ways using the commands in the **Modify** > **Surface** menu of the Model toolbar. You can:

- Add knot curves.
- Remove knot curves.
- Invert the surface normals.
- Open or close surfaces.
- Shift the U or V boundaries of closed surfaces.
- Swap the U and V directions of surfaces.
- Clean surfaces by resampling them.
- Reparameterize surfaces in U or V.
- Extend surfaces to a new boundary curve.
- Stitch the boundaries of two surfaces together.
- Snip a portion of a surface.

Adding Knot Curves

You can add knot curves to increase the resolution of a surface. You can also add multiple knot curves at the same position (multiknots) to create sharp ridges.

1. Select a U or V isoline or a U or V knot curve on a surface.
2. Choose **Modify** > **Surface** > **Insert Knot**.

   The knot curve is added and the Insert Surface Knot property editor opens.

3. Set the options:

   - Use the **Knot Multiplicity** slider to specify the continuity of the knot. For more information, see *Multiknots* on page 354.
   - Use the options in the Surface Curve Adaptor page to fine-tune the position of the new knot curve.

If you add a knot curve to a subsurface of an assembled surface mesh, it is automatically propagated across any junctions to maintain the alignment of control points on subsurfaces.

For more information about surface meshes in general, see *Chapter 13: Surface Meshes* on page 411.

Removing Knot Curves

You can remove knot curves to decrease the resolution of surfaces. For multiknot curves, you can choose how many knot curves to remove.

1. Select a U or V knot curve on a surface.
2. Press Delete or choose **Modify** > **Surface** > **Remove Knot**.

If you did not select a curve in step 1, you can pick a surface and knot curve now. The knot curve is removed and the Remove Surface Knot property editor opens.

3. Set the options as desired:

   - The **Knot Multiplicity** parameter lets you remove some or all of the knot's multiplicity (in the case of multiknots).

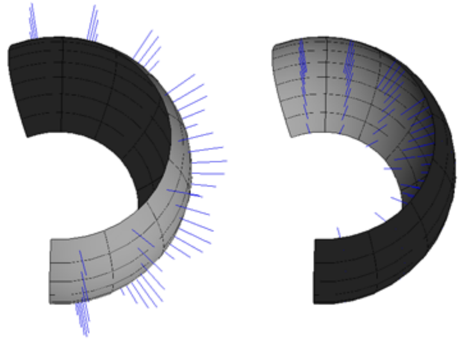
   - The parameters on the Surface Curve Adaptor page let you change which knot curve to remove.

- You cannot remove a knot curve if it is a boundary. You can use Snip to achieve the desired effect—see *Snipping Surfaces* on page 403.

- If you remove a knot curve on a subsurface of an assembled surface mesh, it is automatically propagated across any junctions to maintain the alignment of control points on subsurfaces.

Inverting Normals

If the normals of a surface are pointing in the wrong direction, you can invert them.



Inverting a surface

1. Select one or more surfaces.

2. Choose **Modify** > **Surface** > **Inverse** from the Model toolbar.

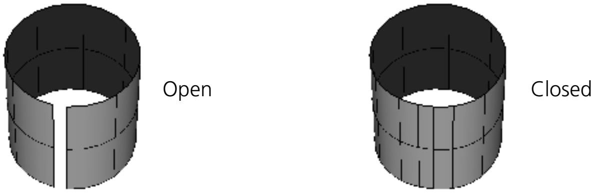
   If you didn't select a surface in step 1, you can pick a surface now. The surface is inverted and the Inverse Surface property editor opens. Leave the **Inverse Surface** option checked.

When you invert a surface in SOFTIMAGE|XSI, the U direction becomes V and V becomes U.

This is different from SOFTIMAGE|3D, where U becomes –U and V is unchanged.

Opening and Closing Surfaces

You can open a closed surface and close an open surface. A surface can be open in both U and V like a grid, closed in both like a torus, or open in one and closed in the other like a tube.



1. Select one or more surfaces.
2. Choose **Modify** > **Surface** > **Open/Close** from the Model toolbar.

   If you didn’t pick anything in step 1, you can pick a surface now. The Open/Close Surface property editor opens.
3. Specify the directions in which to open or close the surface.

   Closed surfaces always open up at their U = 0 or V = 0 boundary—you can shift the boundary position of a closed surface as described in *Shifting UV on Surfaces* on page 400.

   To open a surface along an arbitrary isoline, first add a knot curve at that position then shift the boundary to the new knot curve.

Shifting UV on Surfaces

If a surface is closed in a direction, you can shift the boundary (U = 0 or V = 0 position) along the surface.

To shift UV along surfaces

1. Select a U or V knot curve or a U or V isoline on a surface.
2. Choose **Modify** > **Surface** > **Shift UV**.

   If you didn’t select anything in step 1, you can pick a surface and knot curve now.

   The corresponding U or V boundary is shifted to the selected knot curve (or in the case of an isoline, to the knot curve before the isoline according to the surface’s parameterization) and the Surface Shift property editor opens.
3. If necessary, adjust the parameters. As you change values, the corresponding boundary jumps to the knot curve before the parameter value.

Swapping UV on Surfaces

You can swap the U and V directions on a surface. This simply changes which direction is considered U and which is V for the purpose of other modeling and texturing operations; it does not change the object's shape. The new U direction is the old –V and the new V is the old U, preserving the direction of the normals.

1. Select one or more surfaces.
2. Choose **Modify** > **Surface** > **Swap UV** from the Model toolbar.

   If you didn't pick anything in step 1, you can pick a surface now. The U and V parameters are swapped and the Surface UV Swap property editor opens. Leave the **Swap UVs** option checked.

Cleaning Surfaces

Cleaning a surface reduces the number of control points while keeping the same general shape.

To clean surfaces

1. Select one or more surfaces.
2. Choose **Modify** > **Surface** > **Clean** from the Model toolbar.

   If you didn't select anything in step 1, you can pick a surface now. The surface is resampled and Clean Surface property editor opens.

3. Specify the directions in which to clean. You can also set **Tolerance** separately for U and V; the tolerance controls the maximum difference between the original curve and the resampled shape.

Reparameterizing Surfaces

Parameterization refers to the way that any position on a surface is described in terms of the parameters U and V. Different parameterizations give different behaviors as you manipulate the surface.

When you reparameterize a surface, the new parameterization is in effect at the position of the Reparameterize operator in the operator stack. If you move points or perform other operations on the surface after reparameterizing, the resulting parameterization might not be the same.

When you create a UV texture projection, it is based on the parameterization at the moment of creation. For example, if you want a chord length parameterization for your projection, make sure to reparameterize the surface right before you create the projection. If you reparameterize after applying a UV projection, the texture may shift because the texture returns the color information for a particular UV position but the reparameterization moves that UV position to another location on the surface.

For a description of the available parameterization methods, see *Parameterization* on page 352.

Reparameterization does not maintain discontinuities like multiknots.

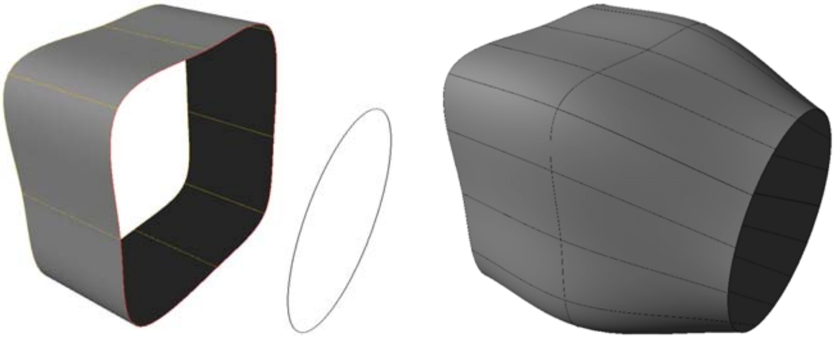
To reparameterize surfaces

1. Select one or more surfaces.
2. Choose **Modify** > **Surface** > **Reparameterize** from the Model toolbar.

   If you didn't pick anything in step 1, you can pick a surface now. The Reparameterize Surface property editor opens.
3. Select a parameterization method.

Extending Surfaces

You can extend a surface object to a curve.



To extend a surface

1. Select a boundary on the surface, then add a curve to the selection. You can select curve objects as well as isolines, knot curves, boundaries, surface curves, and trim curves.
2. Choose **Modify** > **Surface** > **Extend to Curve** on the Model toolbar.

   If you didn't pick both a boundary and a curve in step 1, you can pick the missing inputs in order now. The surface is extended and the Extend to Curve property editor opens.
3. Adjust the parameters as desired. For more information, refer to Online Help by clicking on the ? in the property editor.

Stitching Surfaces

You can stitch the boundaries of two surfaces together. This is similar to merging (described on page 397), but instead of creating a new surface, the original input surfaces are modified. You also have the option of making the stitch pass through an intermediary curve. For best results, first reparameterize the surfaces uniformly as described on page 401.

To stitch surfaces

1. Select two surface boundaries then (optionally) add an intermediary curve to the selection.

2. Choose **Modify** > **Surface** > **Stitch** from the Model toolbar.

   - If you did not select two surface boundaries and an intermediary curve, you can pick the missing inputs in order now.
   - Right-click to terminate the picking session without choosing an intermediary curve.

   The surfaces are stitched and the Stitch Surfaces property editor opens.

3. Set options as desired. For more information, refer to Online Help by clicking on the ? in the property editor.

   - Use the Tolerance option on the **Clean** page to specify the maximum error for calculating the merged surface. Lower values result in more subdivisions.
   - Use the options on the **Shape** page to adjust the overall shape of the blend. In particular, if you selected a curve to pass through, set **Seam** to **Curve**.

Snipping Surfaces

You can snip a surface along an arbitrary isoline, keeping one portion and discarding the other.

Modify > **Surface** > **Snip** is similar to the **Draw** > **Cut** command in SOFTIMAGE|3D. You can duplicate the snipped surface and change **Keep First Surface** in the duplicate to get both halves of the original surface.

To snip a surface

1. Select a U or V isoline or a U or V knot curve on a surface.
2. Choose **Modify** > **Surface** > **Snip** from the Model toolbar.

   If you didn’t pick anything in step 1, you can pick a surface and knot curve now. The surface is snipped and the Snip Surface property editor opens.

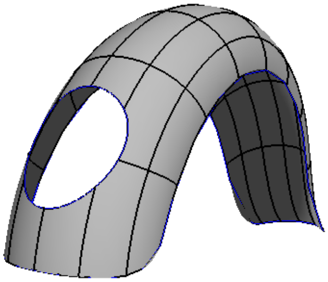
3. Set the options as desired:

   - **Keep First Surface** toggles which portion of the surface is kept and which is discarded.
   - **Snip All Srf Mesh** extends the snip operation across multiple subsurfaces of an assembled surface mesh.
   - The options on the Surface Curve Adaptor page let you fine-tune the position of the snip.

If you select multiple isolines or knot curves before choosing the command, you may not get the results you want. This is because the first snip operation may remove the surface that you wanted to snip with the remaining isolines.

Projecting and Trimming with Curves

You can project curves onto surfaces and then use the result to remove a portion of the surface, or for any other modeling purpose. This is useful for modeling manufactured objects like car parts with holes or for creating smooth surfaces that aren't four-sided like a standard NURBS patch.

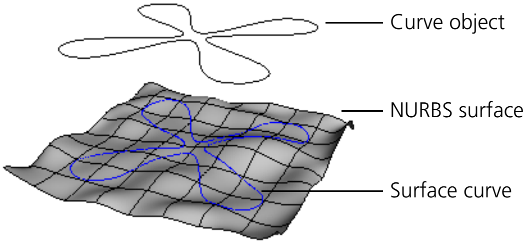


What Are Surface and Trim Curves?

Both surface and trim curves involve projecting a curve object onto a NURBS surface. The difference is whether the result is used to remove a portion of the surface or not.

Surface Curves

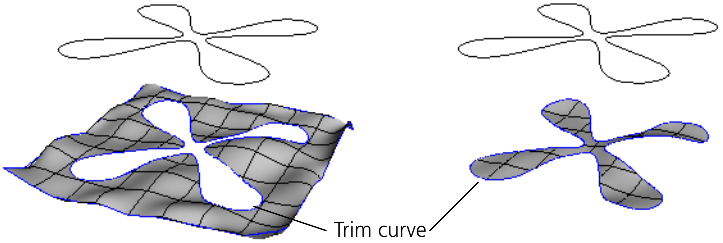
If the curve object is just projected and nothing more, the result is called a *surface curve*. It is a new component of the surface. This surface curve can be used like any other curve component of the surface (isoline, knot curve, and so on) for modeling operations like Loft, Extend to Curve, and others.



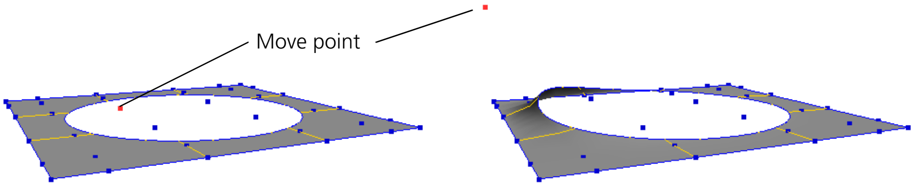
In SOFTIMAGE|3D, surface curves are called *projected curves*.

Trim Curves

If you use the projection to remove part of the surface, it is called a *trim curve*. Trims are useful for making holes in surfaces and for making NURBS patches that don't have the usual four-sided boundary.



Trimming affects the visible portion of the surface. All the underlying points are still there and you can still affect the surface's shape by moving points in the trimmed area. However, you can still use trimmed surfaces as collision objects for particle, soft body, and cloth simulations, and the simulation uses the exact shape of the trimmed surface.

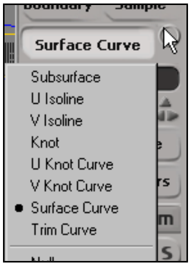


Working with Trim and Surface Curves

Viewing Trim and Surface Curves

By default, surface and trim curves are displayed in dark blue when you select the surface or when it is active for component selection. On trimmed surfaces, the U and V boundaries are also displayed in dark blue, but you can ignore that—they are not truly considered trims.

You can control the resolution used to draw trim and surface curves. The Hardware Display tab of Geometry Approximation property editor controls the resolution used in the OpenGL views, while the Surface Trim tab controls the resolution used for rendering. For more information, see *Geometry Approximation on NURBS* on page 356.



Selecting Trim and Surface Curves

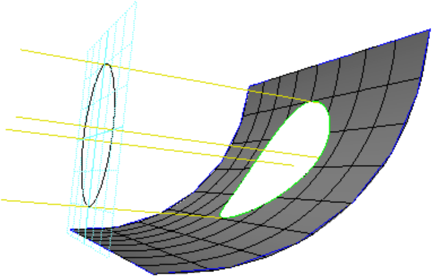
When working with trim and surface curves, you can use the Trim Curve and Surface Curve component selection filters that are available on the Select panel when you select a surface object. For more information about selecting components in general, see *Chapter 8: Selecting* of the *Fundamentals* guide,

Extracting Trim and Surface Curves

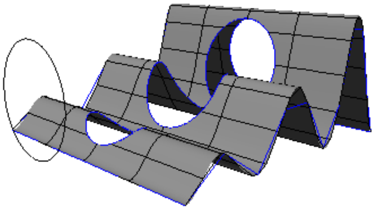
Surface and trim curves can be extracted in exactly the same way as other components of surfaces (like isolines and knot curves). See *Extracting Curves from Surfaces* on page 370.

How Curves Are Projected

In XSI, curve objects are mapped onto surfaces using a parallel projection. The projection follows the normal of the average plane of the curve and extends infinitely both directions.



The curve is projected onto both the front and back sides of a surface and can intersect the surface multiple times if the surface is folded. The curve can punch a hole through the surface or just take a bite out of the edge.



Before You Trim or Project Curves

Here are some things to keep in mind when creating curves to use for projecting:

- Try to keep the points of the curve in the same plane as much as possible. This helps you visualize what the effect will be once it is projected.
- Make sure that the curve does not intersect itself in its average plane. If the curve projects badly onto the surface, parts of it may be ignored.

Projecting or Trimming by Curves

The **Modify** > **Surface** > **Trim by Projection** command is used both to project a curve onto a surface and to trim a surface by a curve.

To project a curve onto a surface or trim a surface by a curve

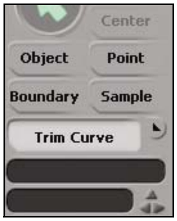
1. Select a NURBS surface object.
2. Choose **Modify** > **Surface** > **Trim by Projection** from the Model toolbar. A picking session starts.
3. Pick a curve object. The curves are projected onto the surface and, by default, the surface is trimmed using all projected curves.
4. In the Trim Surface by Space Curve property editor, do any of the following:
   - To trim the surface using only some of the projected curves, click **Pick Trims** and then pick the desired surface curves. Right-click when you have finished picking.
   - To trim the surface using all the projected curves, click **Trim with All**. This is useful if you used Pick Trims or Project All and then changed your mind.
   - To project the curve onto the surface, click **Project All**. The surface curves become components of the surface and can be used in modeling operations like loft and others.
5. Adjust any of the other options as needed:
   - When trimming rather than simply projecting, use **Is Boundary** to determine which portion of the surface to keep: make a hole by keeping the area outside the curve (off) or crop the outside leaving only the interior (on).
   - Use Projection Precision to control the precision used to calculate the projection. If the shape of the projected curve is not accurate, increase this value.

     However, high values take longer to calculate and may slow down your computer. For best performance, set this parameter to the lowest value that gives good results.

Deleting Trims

Deleting a trim allows you to remove a trim operation even after you have frozen the surface's operator stack. If you have not frozen the stack, you can alternatively reopen the Trim Surface by Space Curve property editor and use **Pick Trims**, or delete the Trim Surface by Space Curve operator altogether.

To delete trim operations



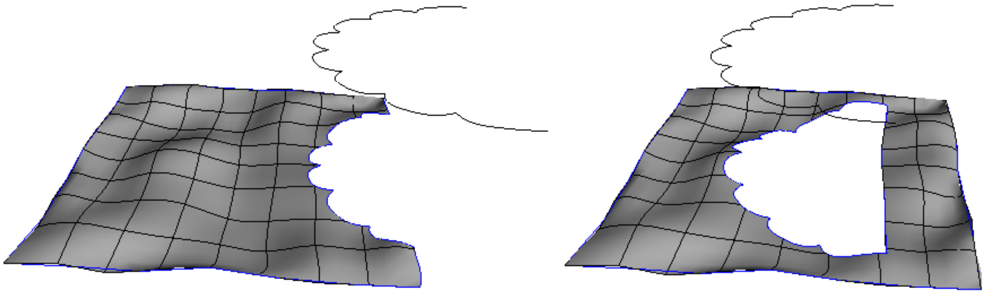
1. Select a trimmed NURBS surface.
2. Set the selection filter to **Trim Curve** and select one or more trim curves on the surface.
3. Choose **Modify** > **Surface** > **Delete Trim** from the Model toolbar.

   If you did not select any trim curves in step 2, you can pick one now.

Notes on Trimming and Projecting

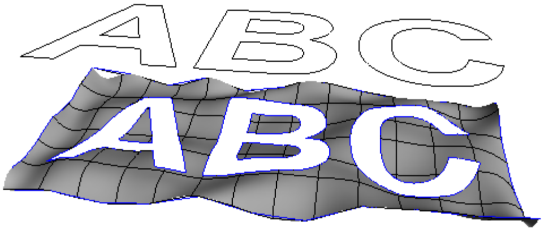
Here are some notes about trimming and projecting:

- You can trim with open curves. This can be useful if you are just taking a bite out of an edge of a surface; you don't need to close the curve first. However if the entire curve can be projected onto the surface, a straight segment connects the end points.

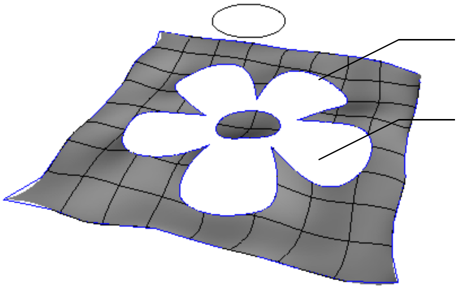


- You can use a single curve object composed of multiple subcurves, such as a curve created with text or by importing an EPS file. In such cases, each subcurve creates a separate surface curve or trim curve component.

  When trimming, disjointed subcurves (such as the exterior curves of different characters) produce separate holes, while interior subcurves produce "islands" within the holes (or vice versa, if **Is Boundary** is on).



- You can use a second trim operation to create an island in the trimmed area. For the second trim operation, make sure that **Is Boundary** is on.



First trim operation makes a hole.

Second operation makes an island.

- When projecting curves onto assembled surface meshes, multiple trim or surface curves will be created if the projection intersects a boundary between subsurfaces.
- You can animate the surface and curve objects to change the resulting trim or surface curve over time.

Surface Tips & Tricks

Here are a few things to keep in mind when working with surfaces.

Optimizing Display and Interaction

As you manipulate objects and play back animation, you may notice that surface objects are displayed in coarse mode. This is an option that allows for faster interaction.

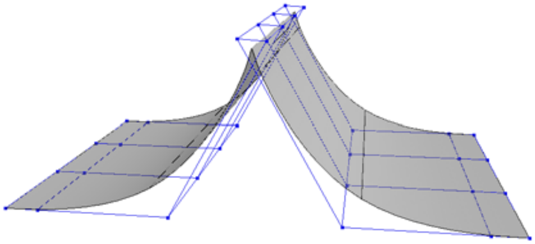
You can toggle **Use Coarse Step by Default for Interaction/Playback** in your Display preferences. You must restart SOFTIMAGE|XSI before this change takes effect.

You can also set the **Step** value to **Coarse**, **Medium**, or **Full** during **Interaction** and **Playback** for **Selected Objects** and **Unselected Objects** in individual 3D views or all open 3D views at once.

For more information, see *Setting Object Display* in Chapter 5 of the *Fundamentals* guide.

Making Sharp Ridges

As an alternative to using multiknot curves, you can create sharp ridges in cubic NURBS surfaces by placing control points so that the hulls intersect themselves. Be careful not to place the points too far apart or the surface will contain a loop.



Importing Surfaces

If you import a Cardinal, Bézier, or B-Spline patch object from SOFTIMAGE|3D, it is converted to a NURBS surface. The following components will be lost:

- Clusters, cluster animation, and cluster constraints will not be included in the conversion.
- If the object is an envelope, the original weight assignment is lost.

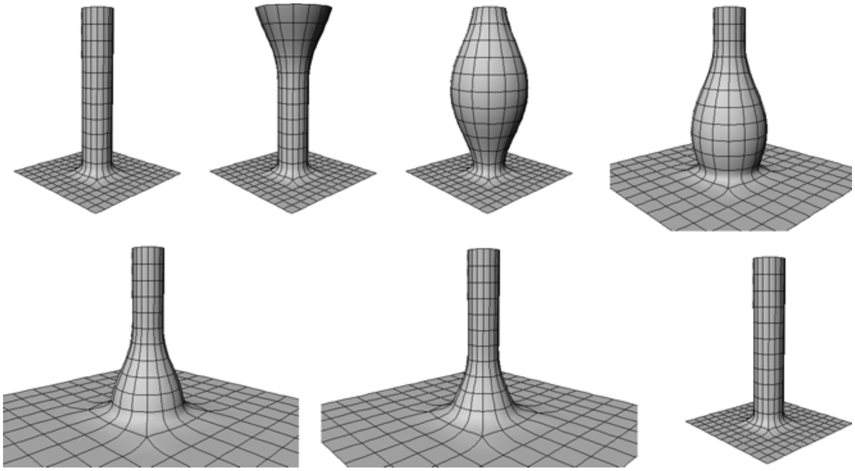
  To avoid this, convert patches to NURBS and use the **Skin** > **Weight Copy** and **Weight Paste** commands in SOFTIMAGE|3D before importing the converted NURBS envelopes.

Chapter 13 Surface Meshes

Surface Meshes

Surface meshes provide a way to assemble multiple surfaces into a single object that remains seamless under animation and deformation. You can:

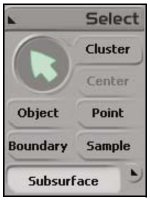
- Snap boundaries together as preparation for assembling.
- Assemble surfaces into a single surface mesh.
- Modify the surface mesh and its subsurfaces in a variety of ways.



About Surface Meshes

Surface meshes are quilts of surfaces joined at their boundaries, just as polygon meshes are quilts of polygons joined at their edges. They are a special type of renderable geometry. Surface meshes are ideal for complex envelopes on skeletons.

Subsurfaces



The component surfaces of a surface are called *subsurfaces*. The continuity at their seams is maintained automatically no matter how the surface mesh is deformed. Subsurfaces can also have their own local clusters, materials, and textures.

The main distinction between surface meshes and ordinary surfaces (such as a primitive or a surface you create) is the number of subsurfaces. Ordinary surfaces have just one subsurface—they can be thought of as extremely simple surface meshes. You can select subsurfaces with the Subsurface filter in the Select panel.

Junctions

Junctions are the boundaries where subsurfaces meet. For smooth continuity to be maintained across a junction, the control points of the subsurface boundaries must line up. Indeed, much of the art of modeling with surface meshes lies in constructing subsurfaces so that their boundary points line up properly. For some information about different types of junctions, see *Junction Types* on page 416.

SCM (Surface Continuity Manager)

The surface continuity manager (SCM) is the operator that ensures that surface meshes are smooth across their junctions. In explorer views, it is called **SCM Fixer Op2** and always remains at the top of the operator stack. Only one surface continuity manager can manage the continuity of a surface mesh at any time.

SCM is applied automatically when you assemble a surface mesh. However, if you freeze the operator stack, you will need to reapply it manually. For more information, see *Applying SCM Manually* on page 426.

Non-fixing Points

Sometimes you want continuity across some junctions but not others, or you want to create a hole in a junction by joining only some of the points. There are two ways to exclude points from surface continuity management:

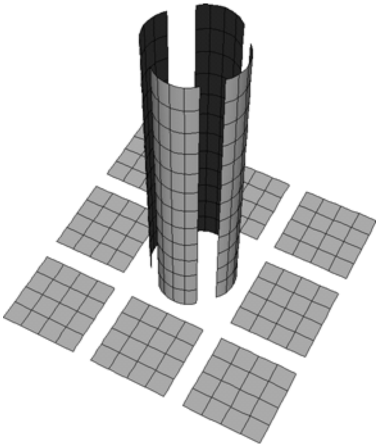
- If at least one pair of corresponding points across a junction are farther apart than a specified tolerance, the entire boundary is excluded from continuity management.
- You have the option of excluding specific points from continuity management. The other points on the same junction are not affected. This lets you create holes for mouths, eyes, and so on. For more information, see *Excluding Points from Continuity Managements* on page 425 for the complete procedure.

Overview of Building Surface Meshes

To build a surface mesh

1. Create a collection of separate surfaces. These will become the surface mesh's subsurfaces.

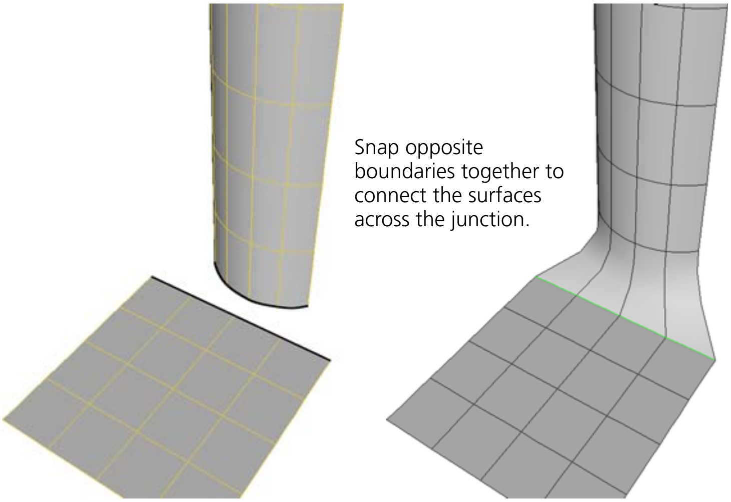
   For example, you can import the component surfaces from scanning software, you can shrinkwrap curves onto a high-resolution polygon mesh and use the curves to build the surfaces, or you can build surfaces from isolines or other curves on existing surfaces.



Line the surfaces up into a basic configuration.

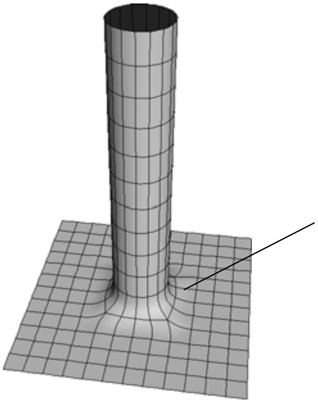
This illustration shows a common configuration for a leg or arm.

2. Optionally, apply the Snap Boundary operator to help align control points along the surfaces' boundaries.



Snap opposite boundaries together to connect the surfaces across the junction.

3. Assemble the surfaces into a single surface mesh. The continuity manager ensures that the continuity is preserved at the seams.



Notice how the assembled surface mesh blends smoothly across the junctions.

You can then deform and animate the surface mesh as desired. Each of these stages is explained in more detail in the sections that follow.

If you ever freeze the assembled surface, you will need to reapply the surface continuity manager manually.

Considerations for Modeling Component Surfaces

There are several considerations when modeling the surfaces that will be assembled into a surface mesh:

- All surfaces must be either cubic NURBS in both the U and V directions, or linear NURBS in both U and V.

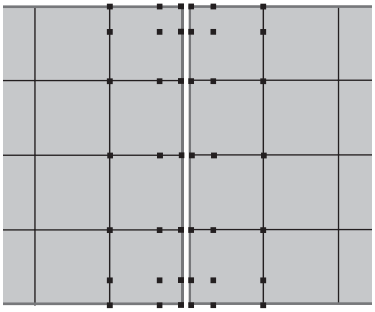
  The SCM (Surface Continuity Manager) does not work with quadratic NURBS.
- Surface meshes work by locking boundary points together then managing the continuity across the junction. Each boundary point can match a single point on another surface's boundary. You can insert and remove knots to create the necessary boundary points. For more information, see *Adding Knot Curves* on page 398 and *Removing Knot Curves* on page 398.
- All junction boundaries must be open.
- SCM uses the next row of points after the boundary to calculate continuity. Make sure that there are enough rows between two junctions: preferably two or more.
- The assemble operation uses distance to determine whether two points across a junction should be locked together, so make sure points are overlapping or very close together. You can apply the Snap Boundary operator to help line up points along boundaries.

Junction Types

Surfaces can meet at an edge (I Junction), a T junction, or a star junction. Alternatively, an edge of one or more surfaces may be collapsed to a single point as at the poles of a sphere.

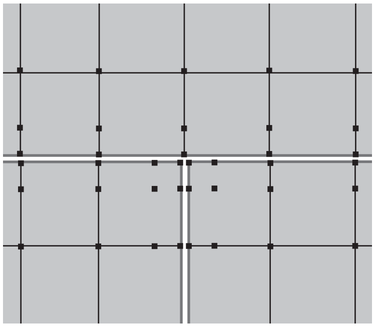
I Junctions

In an I junction, two surfaces join along a common border. This is the simplest of all possible junctions.



T Junctions

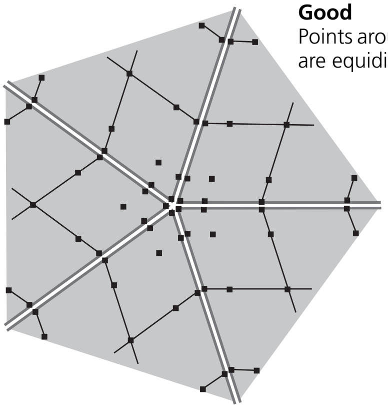
A T junction joins three surfaces. The border of one surface is joined to the borders of the other two. In addition, the other two surfaces share a border.



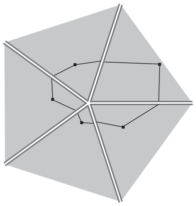
Star Junctions

A star junction brings several surfaces together at a point. You can have three surfaces (Y junction), 4 surfaces (X junction), or more. When using star junctions:

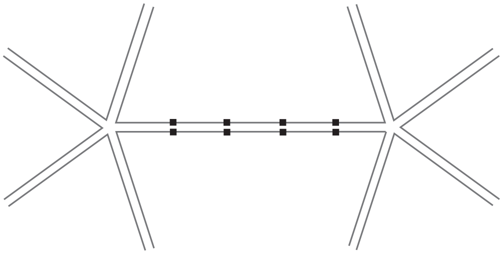
- Make sure that the points around the star junction are more or less equidistant.
- Make sure that there are at least four control points between two star junctions.
- Avoid placing star junctions in areas of high deformation.



Good
Points around the junction are equidistant.



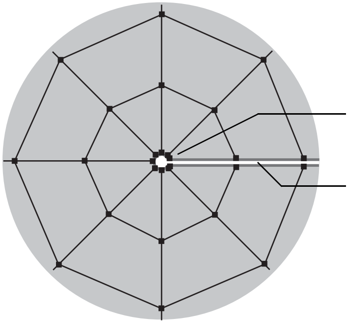
Bad
This junction will not deform well.



Make sure there are at least four pairs of points between star junctions.

Collapsed Junctions

In a collapsed junction, one boundary of a surface collapses to a point (like at the poles of a sphere) and two opposite boundaries joins in a cone shape.

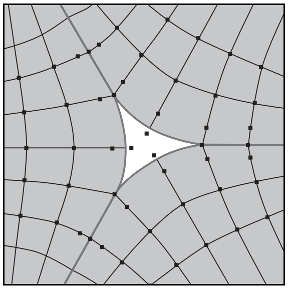


Boundary collapsed to a single point.

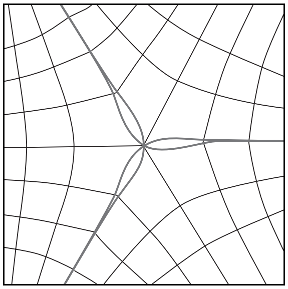
Opposite boundaries lined up.

Multiknots

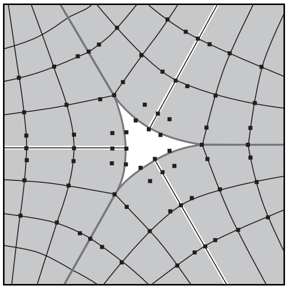
You can use multiknots to create discontinuities that help to line up surfaces. This is especially useful to eliminate holes where three or more surfaces meet in a complex way.



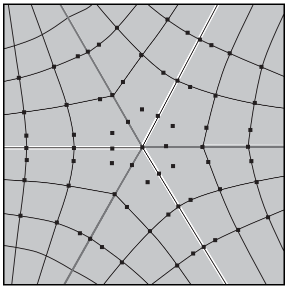
Three surfaces meeting at a point can pose a problem.



Joining them at the central point causes overlapping edges.



To solve the problem, create knot curves of multiplicity 3 (shown here with an outline).



The multiknots create a discontinuity that helps align the surfaces. Continuity is maintained across the multiknot curves as if they were junctions.

Local Materials and Textures

Individual subsurfaces can have local materials and textures. However, UV texture projections are not supported.

If you have already applied materials and textures to the individual surfaces, you have the option of copying them locally when you assemble. Be aware however, that the texture projections remain attached to the original surfaces—thus if you move the assembled surface mesh without moving the projection, the texture will slide.

To avoid sliding while moving the assembled surface mesh

- Do one of the following:
  - Freeze the assembled surface mesh. This also freezes the texture projections on it. However, you will need to manually reapply the surface continuity manager as described in *Applying SCM Manually* on page 426.

    or
  - Constrain or parent the original texture projections to the assembled surface mesh.

Snapping Boundaries

Snapping boundaries constrains the control points on the boundary of one surface to the matching control points on the boundary of another. This is useful when you are aligning points on boundaries before assembling a surface mesh.

To snap boundaries

1. Select two boundaries: one on each of two surfaces.
2. Choose **Create** > **Surf Mesh** > **Snap Boundary** from the Model toolbar.

   If you didn't select both boundaries in step 1, you can pick the remaining ones now.

   The points of the first boundary are aligned to the points of the second boundary and the Snap Boundary property editor opens.
3. Adjust the parameters as desired:
   - **Subsurface** specifies the index of the subsurface that "owns" the boundary if one or other of the surfaces you selected was already a complex surface mesh.
   - **Boundary** specifies which boundary to snap on the corresponding surface.
   - **Inverse Boundary** lines up points in the opposite direction.
   - **Offset** is used when the two boundaries do not have the same number of points. For more information, see the next section *Snapping Boundaries with Different Numbers of Points*.

The Snap Boundary operator is persistent—any time you move the second surface, the boundary points of the first one follow it.

Snapping Boundaries with Different Numbers of Points

If two boundaries don't have the same number of points, you can still snap them. In addition, you can specify an offset to determine which points get snapped together.

To snap boundaries with different numbers of points

1. First select the boundary with fewer points, then add the other boundary to the selection.
2. Choose **Create** > **Surf Mesh** > **Snap Boundary** from the Model toolbar.

   If you didn't select both boundaries in step 1, you can pick the remaining ones now. Remember to pick the one with fewer points first.

   The points of the first boundary are aligned to the points of the second boundary and the Snap Boundary property editor opens.
3. In the Snap Boundary property editor, adjust the **Offset** slider to control which points get snapped to which. Adjust the other parameters as desired.

Assembling Surface Meshes

Once you have created surfaces and aligned their boundaries, you can assemble them into a single surface mesh.

To assemble surface meshes

1. Select all the surfaces.
2. Choose **Create** > **Surf Mesh** > **Assemble** from the Model toolbar. The Assemble NurbsMesh (SCM) dialog box opens.
3. Set the following parameters:

   - Specify a positional tolerance for matching boundary points. Any boundary points that are farther apart than this tolerance value will not be joined and the entire boundary will be excluded from continuity management.

     If the tolerance is too big, too many points will be stitched together and you will not get the desired results.

   - You also have the option of keeping local materials and any clusters you have defined on the surfaces. In addition you can choose whether the center of the new surface mesh is created at its geometric center or at the global scene center.

4. Click OK.

   A surface mesh is created and an SCM Fixer Op2 applies automatically. Any tagged points are added to a special cluster named **NonFixingPointsCluster**.

   The original surfaces remain, but there is no modeling relation between them and the surface mesh.

5. Set the following options:

   - **Always Evaluate** toggles the continuity manager on and off. You can switch this off for faster performance while you work, then switch it back on again before you render.

   - **Continuity** determines the continuity across every junction managed by the SCM Fixer Op: **C0** (or positional continuity; that is, no holes but not necessarily smooth) or **C1** (or tangential continuity; that is, smooth).

- **Don’t Fix the Tagged Points** controls whether points in the NonFixingPointsCluster are excluded from continuity management and create a hole. See *Excluding Points from Continuity Managements* on page 425.
- **Allow Modification of Boundary Points** determines whether you can manipulate boundary points directly by moving them. For more information, see *Moving Boundary Points* on page 424.

To display these options again later, select the assembled surface mesh, then choose the Selection button on the Select panel and click the SCM Fixer Op2 icon.

- The SCM Fixer Op2 node is always evaluated last—it is always at the top of the operator stack even if you apply other operators after it.
- If there appear to be seams, these could be artifacts. Increase the geometry approximation for a smoother result.

  For more information about geometry approximation in general, see *Setting an Object’s Geometry Approximation* in Chapter 3 of the *Rendering & Compositing* guide.

Modifying Surface Meshes

You can deform and transform surface meshes, as well as subsurfaces and subsurface clusters, just like other objects and components. In addition, some procedures the are specific to assembled surface meshes are described below.

Displaying SCM Options

After you have assembled a surface mesh, you can redisplay the surface continuity manager options.

To display SCM options

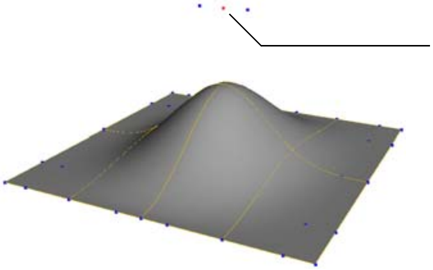
1. Select the assembled surface mesh.
2. Choose the **Selection** button in the Select panel. A pop-up explorer opens.
3. Click the SCM Fixer Op2 icon. The property editor opens.

Moving Boundary Points

There are two ways to move the points on a junction between subsurfaces, depending on whether the **Allow Modification of Boundary Points** option in the SCM Fixer Op2 property editor is on:

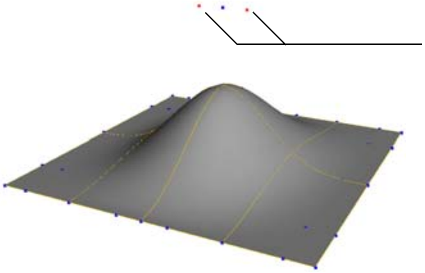
- If it is on, move the boundary point directly.
- If it is off, select the two "ghost" points on either side of the boundary point and translate those instead.

  This was the method for moving boundary points in SOFTIMAGE|3D and the original version of XSI because of the way that the surface continuity manager calculated the continuity across boundaries.



Allow Modification of Boundary Points on

Translate the center point of a junction directly.



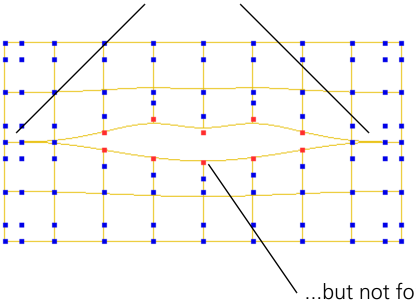
Allow Modification of Boundary Points off

Translate the two ghost points.

Excluding Points from Continuity Managements

All assembled surface meshes have a special cluster called NonFixingPointsCluster. If a point on a subsurface boundary is in this cluster, its continuity is not managed by SCM when **Don’t Fix the Tagged Points** is on. The other points on the same junction are not affected.

This lets you create holes in the surface mesh for mouths, eyes, and so on.



Maintain smooth continuity at either end...

...but not for the tagged points in the middle.

You can add, remove, and modify these holes by adding or removing points from this cluster. You can also toggle the holes on or off for the entire surface mesh.

To add or remove points to or from NonFixingPointsCluster

1. Select NonFixingPointsCluster. The easiest way to do this is:
   - Select the assembled surface mesh.
   - Choose the Clusters button on the Select panel.
   - Click the name NonFixingPointsCluster (not the icon).
2. Add the points to the selection. You can do this by pressing Shift while selecting in Point mode.
3. With both the cluster and the points selected, do one of the following:
   - To add the points to the cluster, choose **Edit** > **Add to Cluster** or click + on the Edit panel.

     or

   - To remove the points from the cluster, choose **Edit** > **Remove from Cluster** or click **–** on the Edit panel.

To toggle holes on or off

To toggle holes on or off for the entire assembled surface mesh, display the SCM Fixer Op2 property editor as described in *Displaying SCM Options* on page 424 and change the **Don’t Fix the Tagged Points** option.

Applying SCM Manually

After you freeze an assembled surface mesh, you need to reapply the surface continuity manager manually. When you apply the SCM Fixer Op to a surface mesh, it maintains the continuity across the assembled boundaries so that the surface mesh appears seamless as it is deformed.

To manually apply SCM

1. Select the assembled surface mesh.
2. Choose **Create** > **Surf Mesh** > **Continuity Manager** from the Model toolbar.
3. If desired, pick additional surface meshes. A separate SCM Fixer Op2 property will be applied to each object.

   When you have selected all the additional surface meshes, right-click the mouse button.
4. Set the options as you did when you originally assembled the mesh. For more information, see *Assembling Surface Meshes* on page 422 or refer to Online Help.

Only one SCM Fixer Op2 node can manage the continuity of a surface mesh at any time.

Inserting and Removing Knot Curves on Surface Meshes

When you insert or remove a knot curve on a a subsurface of an assembled surface mesh, it is automatically propagated across any junctions to maintain the alignment of control points on subsurfaces. For detailed procedures, see *Adding Knot Curves* on page 398 and *Removing Knot Curves* on page 398.

Extracting Curves on Surface Meshes

When you extract a curve on a subsurface of an assembled surface mesh, there is an option to extend it across the whole mesh including any junctions it encounters. For more information, see *Extracting Curves from Surfaces* on page 370.

Applying Local Materials and Textures

You can apply materials and textures locally to selected subsurfaces on a surface-mesh object. This allows you to put different materials and textures on different subsurfaces. Remember that in order to apply a local texture, you must first apply a local material. Local UV texture projections are not supported.

For more information, see *Assigning and Editing Materials* in Chapter 3 and *Applying Texture Projections* in Chapter 7 of the *Shaders, Lights & Cameras* guide.

Moving the SCM Operator

You cannot move the Surface Continuity Manager operator (SCM Fixer Op2). It is always at the top of the stack. For more information about moving operators in the stack, see *Changing the Order of Operators* on page 60.

Index

Hello