SOFTIMAGE®

# SOFTIMAGE®|XSI™

Version 1.0

## Tutorials

Avid

*Tutorials* was written by François Giard, Judy Bayne, Grahame Fulller, and André Demers; edited by Edna Kruger and John Woolfrey; and formatted by Luc Langevin. Production scenes created by François Giard.

© 1999–2000 Avid Technology, Inc. All rights reserved.

SOFTIMAGE and Avid are registered trademarks and XSI is a trademark of Avid Technology, Inc. All other trademarks contained herein are the property of their respective owners.

The SOFTIMAGE|XSI application uses JScript and Visual Basic Scripting Edition from Microsoft Corporation.

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

Printed in Canada.

Document No. 0130-04618-01 0400

# Contents

# Roadmap

## About This Guide

*Tutorials* gives you all the information you need to get up and running with SOFTIMAGE|XSI.

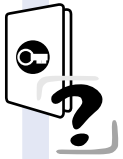
This guide is a collection of eighteen tutorials starting with the basics such as selecting, editing, and transforming objects in 3D space and then moving through the major SOFTIMAGE|XSI tool sets: modeling and deformations, animating, and rendering.

- In ***Tutorial 1: Basic Object Editing***, you are introduced to the methods and basic tools for creating, selecting, and editing objects.
- In ***Tutorial 2: Transformations***, you create a hierarchy of objects, then transform and duplicate them as a single entity. You will also deform objects and apply constraints between objects.
- In ***Tutorial 3: Scripts***, you are shown how to build scripts as an alternative way to input commands. In one exercise, you'll load and edit a script that, when executed, automatically sends you e-mail.
- In ***Tutorial 4: Surfaces***, you are introduced to some of the many ways to create and modify surfaces. You create an alien's head and body, then assemble them into a single surface mesh.
- In ***Tutorial 5: Weight Maps***, you are shown how to use weight maps to control deformations. You can apply a preset weight map such as a gradient, paint deformations, as well as envelope weights.
- In ***Tutorial 6: Constraints, Expressions & Particles***, you use a path constraint to direct the animation of two airplanes, then create an expression to cause one of the airplanes to deviate from the path in a specific manner. Finally, you will use particles to create the airplanes' trails of smoke.
- In ***Tutorial 7: Linked Parameters***, you link parameters and set relative values to automatically open doors as a sphere passes through them.
- In ***Tutorial 8: Skeleton Construction***, you create a skeleton for the Jaiqua character, using chains and objects. The skeleton will allow you to easily pose and animate Jaiqua.
- In ***Tutorial 9: A Walk Cycle***, you will create a walk cycle using the skeleton from the previous tutorial.
- In ***Tutorial 10: The Animation Mixer***, you are introduced to the animation mixer. You create action sources and clips, then reuse, cycle, trim, modify, and mix the actions.
- In ***Tutorial 11: Shape Animation***, you deform an object into different shapes and save shape keys. You also use the animation mixer again to mix between shapes. Finally, you define custom parameters to control the shape with sliders.

- In ***Tutorial 12: Low-Res/High-Res Models***, you are shown how to use high-resolution and low-resolution models to speed up your work. Then, you will use overrides to change property settings without destroying original data.
- In ***Tutorial 13: Envelope Weighting***, you assign an envelope to a skeleton. After the initial automatic weighting of points to deformers, you reassign specific points to different deformers and edit the weights of individual points.
- In ***Tutorial 14: Applying Materials***, you'll create a sphere and apply a material and surface (Phong) shader.
- In ***Tutorial 15: Working with Lights***, you'll create a spotlight and use its manipulators to define its cone and angle spreads and its start and end falloff values.
- In ***Tutorial 16: Giving Life to a Dragonfly***, you'll apply a texture to the dragonfly's eye and copy it (and its texture projection) to the other eye. Then, you'll create a transparency map on all of the dragonfly's wings using a texture image. Afterward, you will texture the dragonfly's body and apply a displacement map to it using the same texture's alpha channel. Finally, you'll merge the scene with a background scene.
- In ***Tutorial 17: Creating Render Passes***, you'll use a static scene to create a matte pass, a shadow pass, a highlight pass and a depth pass.
- In ***Tutorial 18: Editing Shaders in the Render Tree***, you'll use the render tree to create a realistic glass shader, and texture both faces of a flower petal with a different texture on each side.

# Where to Find Information

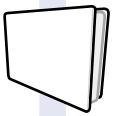
The SOFTIMAGE|XSI package includes a comprehensive set of learning materials. Use this Roadmap to find the information you need to get up and running quickly and effectively.



Start with the ***Setup Guide*** to install and license all components.
**Setup Online Help** is also available as you go through the process.
We recommend you choose Custom install so that you can perform the tutorials.

Refer to ***Release Notes***, an online listing of known problems and limitations for this version. Also includes workarounds and supplemental information. Access through the web at **www.softimage.com > support**.

Follow the ***Guided Tour*** (available from the Online Library CD).
This is a set of videoclips that provide overviews of features and tools.

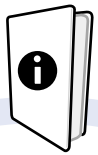


Work through ***Tutorials*** to learn the features in the context of basic productions.
This is a full-color set of lessons showing you step-by-step how to perform typical tasks.
You can install the scenes from the Software CD. (Choose Custom install when installing SOFTIMAGE|XSI). Then choose the **Content** option to install the Tutorials project.

**The Softimage Discussion Group**

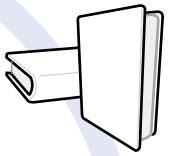
You can join the worldwide network of Softimage users exchanging ideas and techniques by e-mail. To find out more, e-mail **majordomo@softimage.com.** Leave the Subject line empty and type the word "help" in the body of your mail message.

The ***Global Index & Glossary*** is an index to all user guides and *Tutorials*; a glossary of terms; and a list of books, videos, and web sites related to the 3D animation industry.



The **user guides** contain conceptual information and procedures on how to use specific tools. These comprise**:**

- Fundamentals
- Animating
- Modeling & Deformations
- Shaders, Lights & Cameras
- Rendering



**The Online Library CD**
The Online Library contains the Guided Tour and all the SOFTIMAGE|XSI and some mental ray documentation in electronic form in both PDF and HTML formats. (See next page for how to use.)

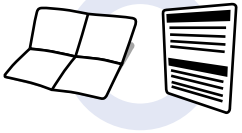
**Online Help**
On-screen reference information on interface elements, commands, and parameters. There are two ways to access it:

- Click the **?** button in any property editor or tool view.
- Choose **Help > Contents and Index** from the main-menu bar.

***Using SOFTIMAGE|3D with SOFTIMAGE|XSI*** provides tips and techniques about using the two software packages. Available from the Online Library CD and **softimage.com > support** only)


**HTML Scripting Reference**
An HTML-based reference help on the syntax for all scripting commands and arguments. It appears in your default HTML browser. Click on the icon (above) to open the script editor, then click **Help > Scripting Reference** or press **F1**.

Pin up the ***SOFTIMAGE|XSI Interface Layout*** and the ***Quick Reference Card*** to help you become familiar with the interface and keyboard shortcuts.



### Using the Online Library

The Online Library contains the *Guided Tour* and all the SOFTIMAGE|XSI and some mental ray documentation in electronic form in both PDF and HTML formats.

For full-text searching and printing, we recommend PDF format. If you do not have Acrobat Reader installed, you can install it it free of charge from the Online Library CD: Follow the instructions in the readme file on the CD.

*To access the Online Library*

1. Insert the Online Library CD in your disk drive.
2. Open one of the following documents:
   - **mainmenu.pdf** (PDF format)
   - **mainmenu.htm** (HTML format)

## 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 emphasized words. |
| `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 contain 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. You will find these very helpful when working with the two products.

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 a task!

## 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.

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. |

# Section 1 Getting Started

# Installing the Tutorial Project

The tutorials in this workbook use scene content that is supplied on the Software CD. For these scenes to be available, you must have chosen the **Custom** install option when you installed the software.

If you did not choose the Custom install option, you can use SOFTIMAGE|XSI's Setup utility to load the tutorial scenes:

- For Windows NT, you can access Setup from **\SOFTIMAGE Products\XSI_1.0\Setup**
- For IRIX, you can run Setup from the directory where your SOFTIMAGE|XSI software was installed.

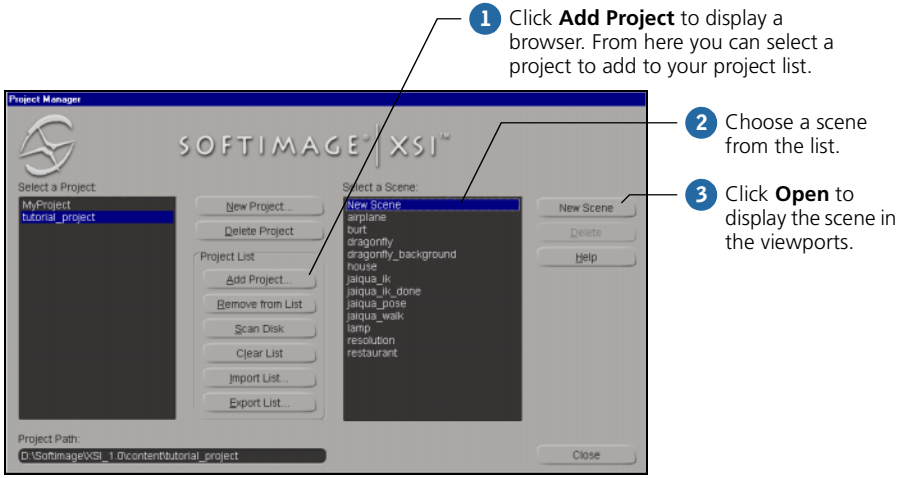
Once the Setup dialog box is displayed, click **Add/Remove**. In the Maintenance Mode dialog box, select the **Content** check box and click **Continue**. The utility then installs the tutorial scenes in your **\XSI_1.0\content\tutorial_project** folder.

The next section shows you how to add the tutorial project to your project list so you can access its scenes.

# Adding a Project to the Project Manager

The first time you run SOFTIMAGE|XSI, the Project Manager appears. The Project Manager gives you access to all projects and scenes. You will add a project to the Project Manager's project list so that you can then select its scenes and use them in the tutorials in this workbook.

1. In the Project Manager, click **Add Project**.
2. Navigate to **XSI_1.0\content**, select **tutorial_project**, and click **Select**. Tutorial_project is added to your list of projects.
3. Select a scene from the list of project scenes (on the right side) and click **Open**.
4. To add another project to the Project Manager, choose **File** > **Project Manager** from the main-menu bar at the top of the window.
5. To open another SOFTIMAGE|XSI scene, choose **File** > **Open**.



# The Interface

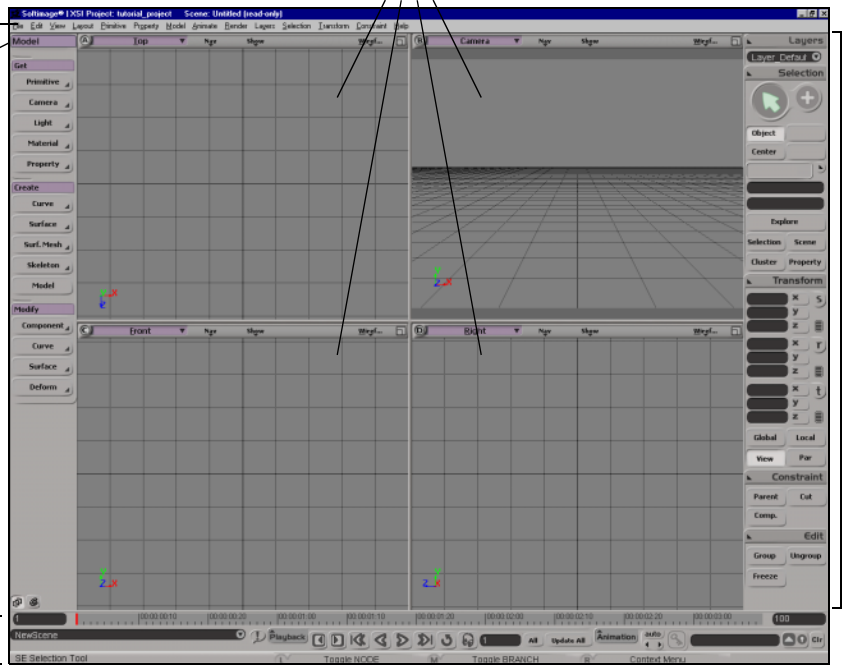
Below is the default layout. For more information about the interface, please see the *SOFTIMAGE®|XSI™ Interface Layout* foldout, *Chapter 2: The Interface* of the *Fundamentals* guide, and Online Help, accessible at any moment: from the main-menu bar, choose **Help > Contents and Index.**

**Viewports** let you view the contents in your scene in many different ways. You can resize, hide, and mute viewports in any combination.

Press F12 to display the viewport under the cursor at full screen. Press F12 again to revert to a four-viewport display.

**Main-menu bar** provides access to all the primary commands.

**Toolbar** displays one of three toolsets: Model, Animate, or Render. Click the toolbar label or press 1, 2 or 3 on the keyboard to switch between toolsets.



**Lower interface** is where you can create and activate scripts, as well as edit animation and play it back.

**Main command area** lets you transform, edit, select, constrain, and organize scene elements.

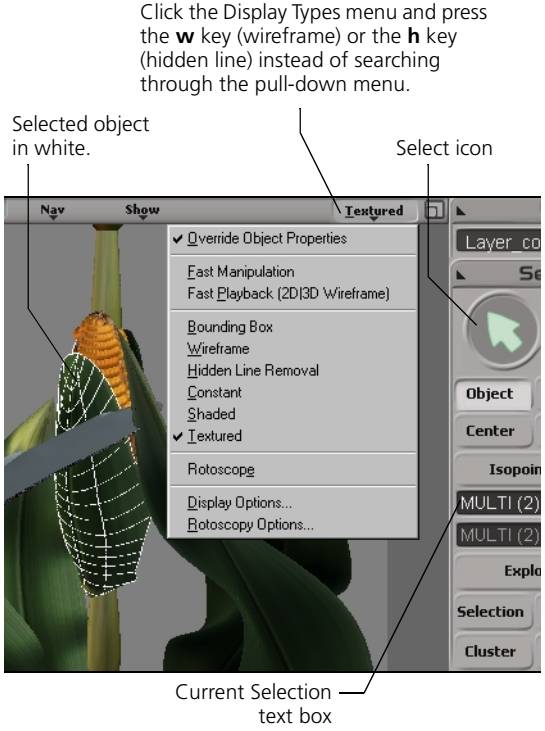
# Interaction Tools

Even after more than a decade, we still access the 3D world via a 2D screen. SOFTIMAGE|XSI offers many tools to make this interaction easier.

The tools described in this section help you navigate in a scene and select its contents.

## Supra Keys

Many tools can be activated by means of supra keys, which are a kind of keyboard shortcut that replace "point-and-click" menu selection. The **z** key for instance, activates the zoom tool. Supra keys are "sticky," meaning that if you press a key quickly, the tool stays activated. If, however, you leave your finger on the key while using the tool, the tool deactivates when you release it. For example, there's no need to keep the **z** key pressed while you're zooming in on an object. An icon appears instead of the mouse pointer to tell you which tool you've activated. You can cancel a tool any time by clicking the same supra key again, pressing another supra key to activate another tool, or pressing the Esc key.




### Using Supra Keys to Zoom, Orbit, and Pan

To navigate in a scene, move the mouse pointer into any viewport, then do the following:

- To orbit the scene, press the **o** key and click and drag the mouse. Note that you can only dolly within the Camera view.
- To pan the scene, press the **z** key and click+drag the mouse.
- To zoom into the scene, press the **z** key and middle-click+drag the mouse. To zoom out, right-click+drag the mouse.
- To dolly in the scene, press the **p** key and click+drag the mouse. Note that you can only dolly within the Camera view.
- The **s** key is another useful key because it acts as a pan, dolly, and orbit all in one. Use it in the Camera view and it will not cause any distortion in your viewport: press the **s** key and left-click to drag and pan, middle-click to dolly, and right-click to orbit.

When you press any key, watch the Mouse/Status Line at the bottom of the window for instructions.

## Selecting Objects in a Scene

To select objects in the scene, all you have to do is click on them—you may have to press the space bar first (the default selection supra key), which activates object selection. If the big arrow (Select icon) in the upper-right corner of the window is highlighted in green, you know you're in selection mode.

Typing an object's name in the Current Selection text box is another method of selecting an object. You can also combine an object's name with a wildcard (*) to select several objects that have a common name. For example, typing `arm*` selects every object in a scene whose name begins with the word "arm."

## Access Keys

If you open the menus of just about any application, you will see command names with underlined characters. These are *access keys* (sometimes called mnemonics) and are simply quick ways to choose a menu item using the keyboard.

You will quickly learn which ones are most useful to you. For example, to switch a given viewport to hidden line display, click on the **Display Type** menu on a viewport menu bar and press the **h** key, or **w** for wireframe, and **s** shaded. To create a sphere, click on the **Primitive** menu and press **s** twice for **Surface** > **Sphere**, or press **c** twice for **Curve** > **Circle**, etc. You could also map a shortcut key directly to these commands—it's all a matter of what you prefer and find the most useful.

To view all of the default keys and add your own shortcut keys, choose **File** > **Keyboard Mapping** from the main-menu bar.

*For more information, please see....*

- *Supra Keys* in Chapter 2 of the *Fundamentals* guide.
- The *SOFTIMAGE®\|XSI™ Quick Reference Card*

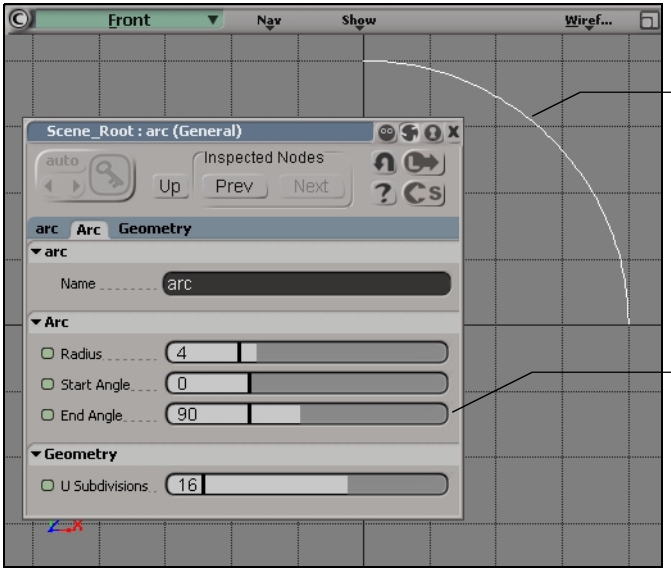
# Section 2 **Basics**

# Tutorial 1: Basic Object Editing

By using a simple primitive, this tutorial will quickly step you through the basic tools for editing and selecting objects. As you do this, you will be introduced to the various attributes and properties of 3D objects.

## Editing Properties

1. Choose **File** > **New Scene** from the main-menu bar to create a new workspace.
2. From the Model toolbar, choose **Get** > **Primitive** > **Curve** > **Arc**.
3. Once the default arc has been created, its property editor opens. Here, you can edit the primary properties of the arc. You can, for example, modify the arc's **Start Angle** and **End Angle** parameters by dragging their sliders.

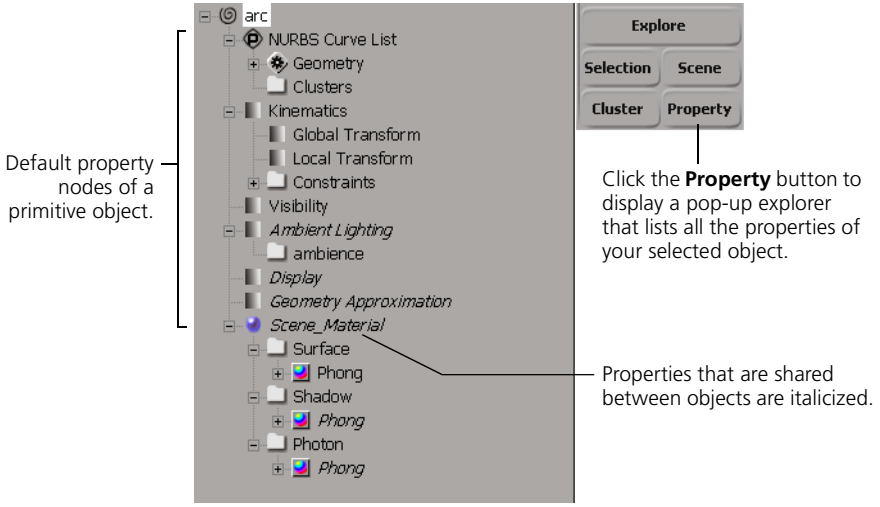



Press **f** to frame the selection in the viewport.

Press **s** to track (dolly) and orbit.

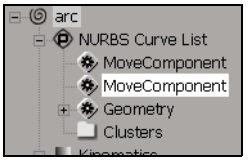
Edit the parameters and get immediate visual feedback in the viewports.

4. Click the **Property** button in the Selection panel of the main command area to display a list of the object's properties.




You can edit properties within a property editor and directly in a viewport.

5. Press **m** and drag to move a point on the arc from within the Camera view.
6. Modify the **Start Angle** and **End Angle** of the arc by dragging sliders in its property editor.
7. Move another point on the arc.
8. Click the **Property** button again to display the arc's properties and note that two MoveComponent operator nodes were added under the object's primitive node.




9. Select one of the MoveComponent node labels and press the Delete key on the keyboard. You will see that the corresponding "move point" operation was deleted from the arc curve.

# Tutorial 2: Transformations

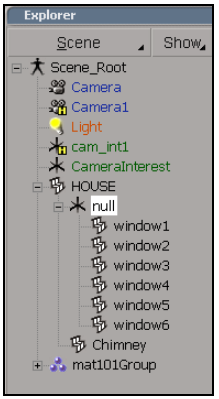
Creating relationships between objects is inevitable when building a scene, especially if the objects will be animated. As in life, one of the most basic relationships between objects is that of **parent-child**. This relationship also embodies the first rule of propagation: if the parent is blue and the child has no local color (surface shader), it will inherit its parent's blue color. Likewise, if a parent object is translated, its child will inherit the translation as well. A proper hierarchical setup can help you when you are selecting, transforming, and manipulating your scene objects.

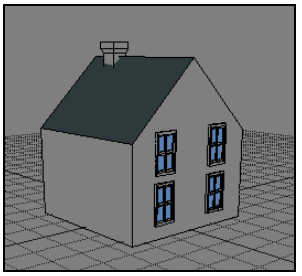
This tutorial shows you how to:

- Create hierarchical relationships between objects through parenting.
- Transform and duplicate objects.
- Edit object properties using the property editor.
- Use simple deforms and constraints.

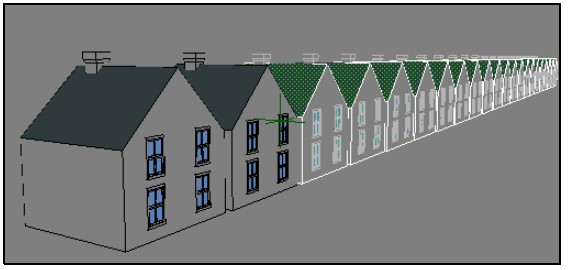
# Overview

**1** **Load the House scene, then parent objects to create a house hierarchy.**

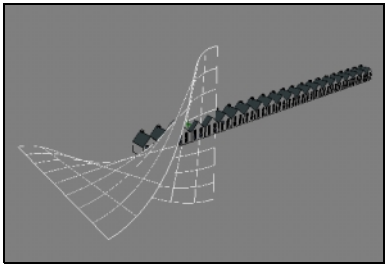


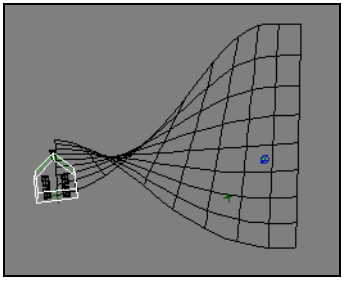



**2** **Select the house hierarchy and duplicate it to create a group of 22 houses.**

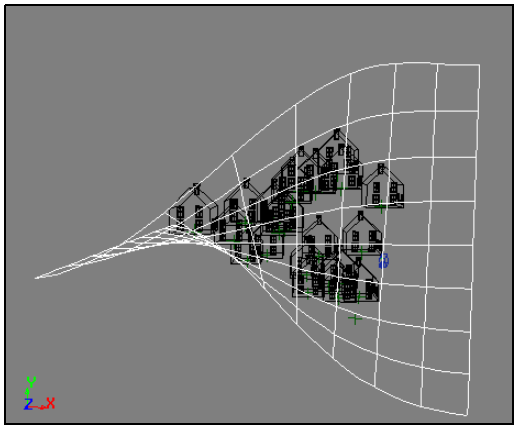


**3** **Create a deformation object and constrain the houses to its surface.**

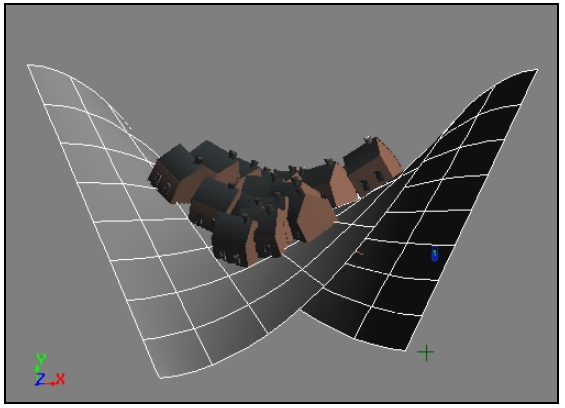




4. **Set random and relative values on the surface constraint to position the houses.**



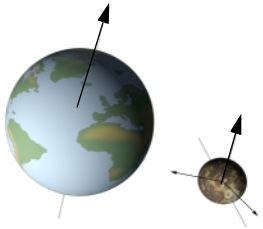
5. **Set tangency and normals constraints to fine-tune the end result.**



## The Hierarchy

Each object has a one-point position defined in space called the **object center.** This position is always calculated according to a reference. This reference (global or local) is, by default, the 3D space itself. When an object is the child of another object, its local reference is the parent object's center. The global reference always stays relative to the center of 3D world space (shown by a cross in the middle of the default SOFTIMAGE|XSI grid).

Choose **View** > **Centers** from the main-menu bar to display object centers in all the viewports.



**Parent/Child Example:**
Here, the moon has been parented to the earth.

The moon's local reference is now the earth's center.

When the earth's center is transformed, the moon's center will be transformed as well.

## Creating a Hierarchy

There are a number of possible ways to build these parenting relations and use them efficiently.

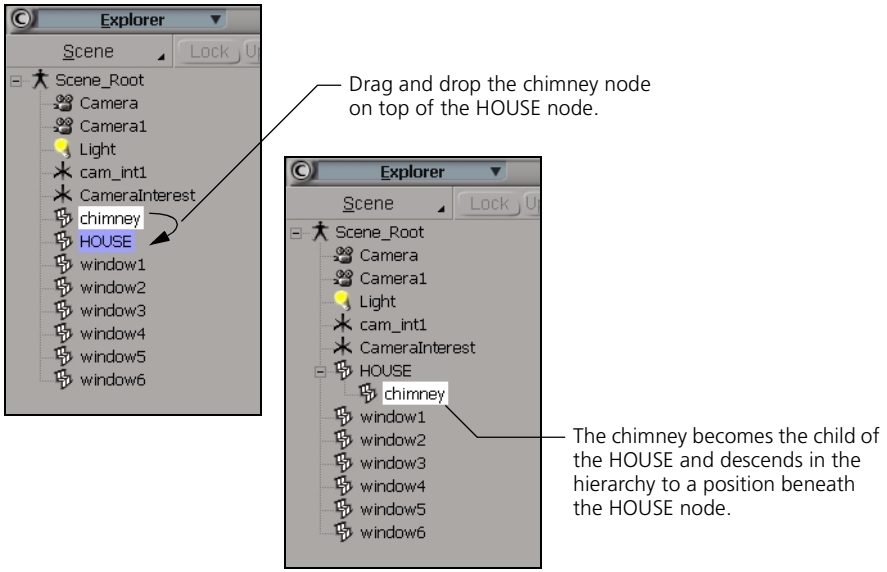
1. Open the HOUSE scene from the tutorial database:
   **<install directory>\content\tutorial_project\Scenes**
2. From the main-menu bar, choose **View** > **Views** > **Explorer** to open the explorer in a floating window.

   Notice how all parts of the house appear in the scene root at the same level.

### *Drag and drop relationships*

An easy way to make a hierarchical relationship is to drag and drop an object node (child) on top of another node (its parent) using the explorer.

3. In the explorer, select the chimney object node.
4. Drag and drop the chimney over the HOUSE object node. The chimney becomes the child of the HOUSE and descends in the hierarchy to a position beneath the HOUSE node.




5. Congratulations on your first child!

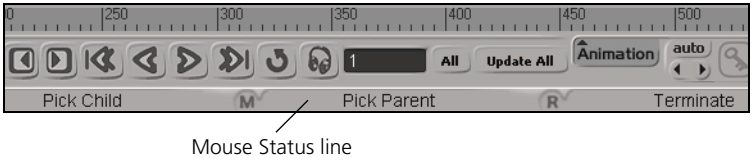
### *Parenting by command*

The **Parent** button in the Constraint panel of the main command area lets you create hierarchical relationships from a viewport.

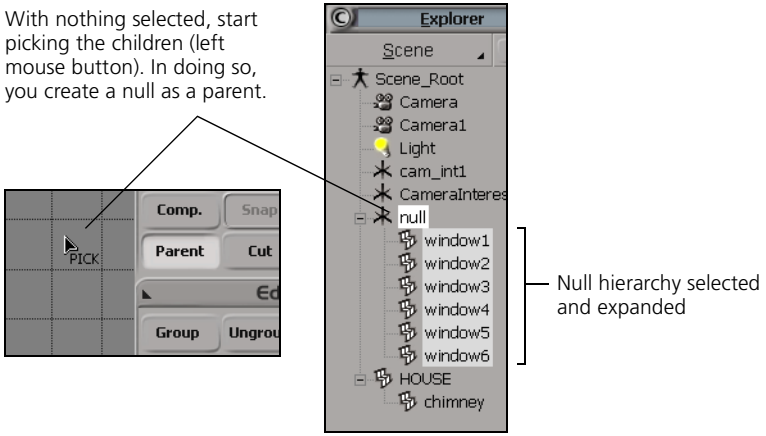
You can accelerate the parenting process by using the forward slash (/) supra key instead of the **Parent** button.

6. Deselect all objects by clicking and dragging in an empty area of the viewport. You can also choose **Selection** > **Deselect all**.
7. Click the **Parent** button. Notice how the pointer changes while in a viewport when Parenting mode is enabled.

When you click the **Parent** button, your mouse buttons are mapped to a specific parenting command: the left mouse button picks the child objects to be parented, the middle mouse button picks the parent object for the selected object, and the right mouse button ends the picking session for Parent mode. Look at the mouse status line for your current mouse button status.



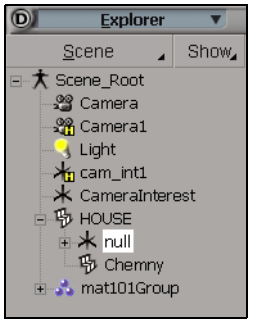

8. From any viewport, left-click to pick each of the six window objects as children. Because no parent was first selected, the parenting command automatically creates a null object to be used as the parent of the first selected child.




9. Exit Parent mode by deselecting the **Parent** button, pressing Esc, or right-clicking. This is a very important step to do, otherwise you may continue to add children to your hierarchy, select different parents, etc.

10. Expand the HOUSE and null objects. The tree in the explorer should be similar to that shown in the previous illustration.

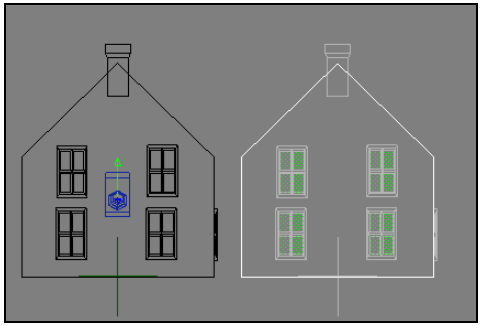
11. Drag and drop the null over the HOUSE to complete the house hierarchy. You can now select individual objects in the hierarchy (left-click); select an object and its children or *branch-select* (middle-click); select the whole hierarchy or *tree-select* (right-click).




12. Select the HOUSE hierarchy: in a viewport, right-click and drag across the house object to select its hierarchy.

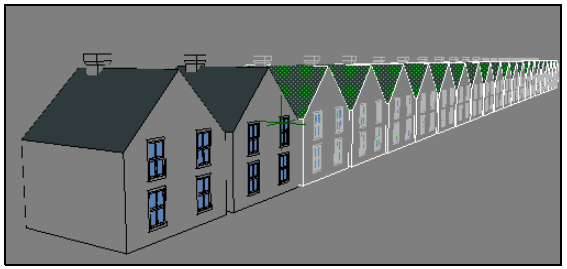
## Duplicating an Object

13. With the HOUSE hierarchy selected, press Ctrl+d to instantly duplicate the hierarchy.
14. Press the **v** key to enter translation mode, then position the newly duplicated house next to the original.



15. From the Edit panel in the main command area, choose **Edit** > **Duplicate Multiple**. Enter `20` in the **Number of Copies text** box and click OK. Twenty duplicates of the house are created, each one offset the same amount from the others.

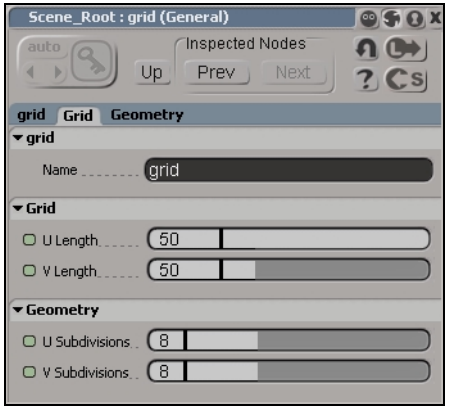
The offset used by the duplicate multiple is taken from the first offset created between the two first houses. Choose **Edit** > **Duplicate/Instantiate Options** to enter a different offset.



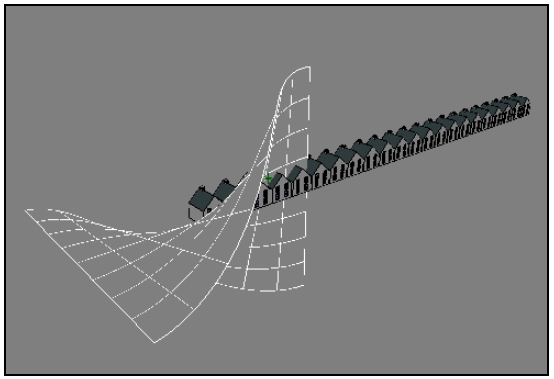
## Adding a Twist Deformation

Now create a grid that you will twist.

16. From the Model toolbar, choose **Get** > **Primitive** > **Surface** > **Grid**. In the property editor that appears, click the **Grid** tab and set the Grid **U Length** and **V Length** to 50.



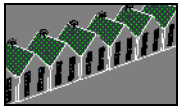

17. With the grid still selected, choose **Modify** > **Deform** > **Twist**. Close the displayed property editor to accept the default values.



18. Type **house*** in the Current Selection text box of the Selection panel, then press Enter. This selects all objects in the scene with the word "house" in its name.


Type house*


Press Enter to display MULTI selection.



All the house objects in the scene are selected.

## Using a Surface Constraint

With all the houses selected, you will constrain them to the grid's surface.

19. From the Constraint panel in the main command area, choose **Constraint** > **Surface**.
20. Pick the grid. This causes all the houses to be constrained to the same position on the grid surface. This also opens the Surface Constraint property editor for the multiple selection.

In the next steps, you will edit the surface constraint for the multiple selection in a multi-selection property editor. Any changes made to the surface constraint properties will be inherited by all objects in this selection. If you lose the selection focus or you inadvertently close the [**Multi**]**: Surface Cns** property editor, you can recover it by using the **Prev** and **Next** Inspected Nodes buttons at the top of the property editor, or you can do the following:

21. Select a single house.

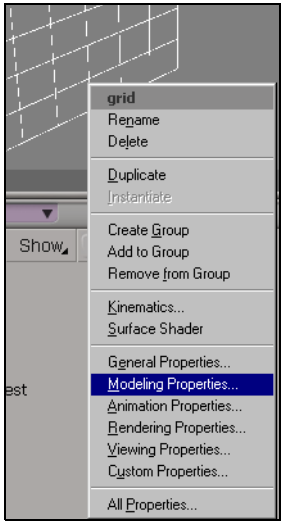
22. Click the **Selection** button in the Selection panel and select its **Kinematics** > **Constraints** > **Surface Cns** node for the single house.

23. In the Current Selection text box, type `house*` and press Enter to open a surface constraint multiple property editor.

## Muting an Operator

You can temporarily mute the Twist operator to see exactly where on the surface you are positioning the houses. To do this, you'll first need to find the twist operator.

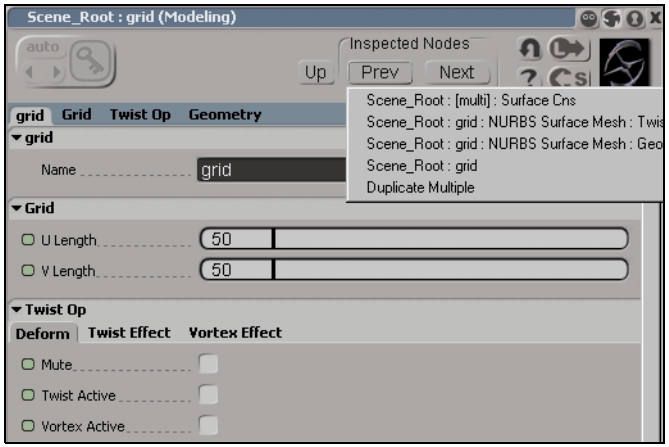
24. Select the grid. Alt+right-click the grid object and choose **Modeling Properties** from the pop-up menu.




25. In the property editor that opens, click the **Twist Op** tab and select the **Mute** check box.

## Adding Random Values

26. Right-click the **Prev** Inspected Nodes button and select the **[Multi]: Surface Cns** property editor.




27. In the Surface Constraint property editor, enter `r(0.5)` for the **U Location**. This randomly positions objects along half of the U direction of the surface.

The UV Location values of the Surface Constraint editor range from 0 to 1. Entering a value such as 100 would give a cropped result (cropped values are left at nearest min/max). This is why this example uses a decimal value (0.5).

28. Enter `r(0.5)` for the **V Location**. Leave the Constraint property editor open for now.

- Entering **r** in a text box generates a random number.
- Entering **r(x)** generates a number between 0 and x. The x value can also be negative; for example, **r(–100)** will generate a value between –x and 0.
- Entering **r(x,y)** generates a number between x and y, where x or y could be less than 0. For example, **r(–100, 100)** and **r(100, –100)** will both provide similar results—values between –100 and 100.

Once the random equation is entered, the property editor considers the result only. Therefore just the result, not the equation itself, can be edited.

## Adding Relative Values

All the houses are now randomly distributed along the first quarter of the UV surface. You will notice that the random value cannot be displayed in the UV parameters because more than one value is applied, but you can still increment each of these values.

29. Enter `0.25+` for the **U Location** to offset all the given values in the U. Repeat for the **V Location**.
30. Now remove the mute from the Twist operator: select the grid, Alt+right-click the grid, and choose **Modeling Properties** from the pop-up menu.
31. In the property editor that appears, click the **Twist Op** tab and deselect the **Mute** check box to make the Twist deform active.

## Tangency and Normal Constraints

32. Click the **Prev** Inspected Nodes button until you return to the [**Multi**]**: Surface Cns** property editor.
33. Click the **Normal** tab and select the **Active** check box to constrain in the direction of the normals.
34. Click the **Tangency** tab and select the **Active** check box.
35. Enter `r` in the Z **Axis to Align** (and press Enter) to rotate the houses randomly.

Take a look at the scene in Shaded view. You may need to orbit the camera to see the houses clearly.

## Conclusion

Constraints are particularly useful for creating complex behavior by having objects automatically react to another's animation. SOFTIMAGE|XSI provides you with a range of constraint types from which to choose.

See *Chapter 4: Animating with Constraints* in the *Animating* guide for more information on creating and editing constraint behavior between objects.

# Tutorial 3: Scripts

Anything that you can do interactively in SOFTIMAGE|XSI can also be done by writing scripts. Simple scripts are basic SOFTIMAGE|XSI commands. More advanced scripts use a host language such as Visual Basic to combine SOFTIMAGE|XSI commands.

SOFTIMAGE|XSI logs every command into a command history. You can retrieve these commands and use them to create simple, one-command scripts that automate repetitive tasks, or use them as building blocks in your own, more elaborate scripts.

In this tutorial you will:

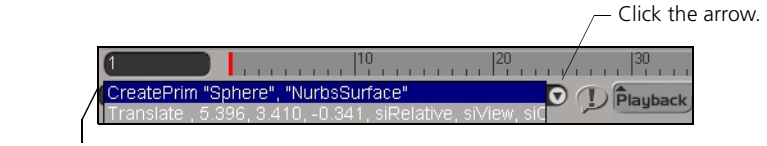
- Open the script editor and repeat commands from the history.
- Create a button for a custom command.
- Try out a script that interacts with an e-mail application to send mail from within SOFTIMAGE|XSI.

## Using the Command History

Opens the script editor.

1. Open the script editor by clicking on the Scripting icon—the exclamation point (!)—beside the Playback controls below the timeline.
2. Clear the history log by choosing **Edit** > **Clear History Log** from the script editor command bar.
3. From any toolbar, choose **Get** > **Primitive** > **Surface** > **Sphere** to create a sphere. Notice how the command is instantly logged in the History pane of the script editor.
4. Translate the sphere in any direction.
5. From the Command Box (at the extreme left of the timeline), click the arrow to show the drop-down menu and select the line:

   ```
   CreatePrim "Sphere", "NurbsSurface"
   ```




6. Press Enter. The Create Sphere command is repeated.

   You have just used scripting in its most basic form: using the Command History to execute a command.

## Creating Custom Commands

7. Choose **View** > **Custom Toolbars** > **New Toolbar** from the main-menu bar and name the new toolbar **Delta** (or anything else you want). Click OK. Leave the toolbar open.
8. Select the first two lines in the history log:

   ```
   CreatePrim "Sphere", "NurbsSurface"
   Translate, xxxx, xxxx, xxxx, xxxx, xxxx, xxxx, xxxx
   ```

   These two commands create and translate a NURBS sphere.
9. Drag and drop these two lines onto the toolbar.
10. In the Add Script Command dialog box that opens, type `Quick Sphere` as the command name. This is the name that will appear on the button in the toolbar. On the line below you can enter a command name for scripting: type **`Quicksphere`**.

11. Click OK to close the Add Script Command dialog box. The new button is added to the toolbar.
12. From the Edit panel of the main command area, choose **Edit** > **Select All**, then **Edit** > **Delete Selected** (do not save the scene).
13. Click on the **Quick Sphere** button in the toolbar. Your custom command creates and translates a sphere. Close the toolbar.
14. Once again, choose choose **Edit** > **Select All**, then **Edit** > **Delete Selected**.
15. In the Command Box, type **`Quicksphere`** and press Enter to invoke your saved script. This is another method of calling and executing a frequently used script from the Command Box.


## Interacting with Other Applications

Scripts allow you to interact with other applications that recognize the host script language. For example, you can have SOFTIMAGE|XSI send you an e-mail when a render is complete. To learn more about this feature, you will load the e-mail example available in **<install directory>\content\tutorial_project\scripts\MAIL_TEST.vbs**

### E-mail Example

This example works for Windows NT only and requires Microsoft Outlook as the e-mail application.

1. Choose **View** > **Custom Toolbars** > **New Toolbar** from the main-menu bar to create a toolbar. Name it `Test`.
2. Open the script editor by clicking the Scripting icon (**!**) to the left of the timeline.
3. Click the **New** button.

Always click **New** before working on a new script even if the script-editor window is empty. The script-editor title bar should display “untitled”—this prevents you from accidentally overwriting script files.

4. Open a browser, and drag and drop the MAIL_TEST script from the data folder (**<install directory>\content\tutorial_project\Scripts**) onto the script editor.

5. Remove the “xxx” item in the message line and type in your e-mail address; for example, “me@company.com”.

This script will execute once when it is parsed. You must enter a valid e-mail address for the script to parse properly. After the script has been parsed, it will use the address you enter in the next few steps.

6. Click the **Save** button and close the script editor.
7. From the browser, drag and drop the MAIL_TEST script onto the toolbar you created.
8. In the **Add Script Command** dialog box that appears, click the **Parse Script** button. Once the script is parsed, its variables are displayed.
9. Click inside the **Value** text box for each parameter and choose **Prompt on Run** from the pull-down menu for each of the parameters: dst, subj, and body. Click OK to close.
10. Click the **MAIL_TEST** button in the toolbar.
11. Enter your e-mail address for “dst,” your subject title for “subj,” and your e-mail text for “body.” Click OK.

You have mail!

## Conclusion

Scripting is a powerful and flexible tool for doing almost anything. Almost all features in SOFTIMAGE|XSI are available through scripting. You can copy and paste commands from the history to automate repetitive tasks, or write your own scripts from scratch. And you are not limited to a proprietary scripting language—you can use any scripting language that supports ActiveX, including VBScript, JScript, PerlScript, and Python ActiveX Scripting.

For more information, see *Chapter 7: Commands & Scripts* of the *Fundamentals* guide.

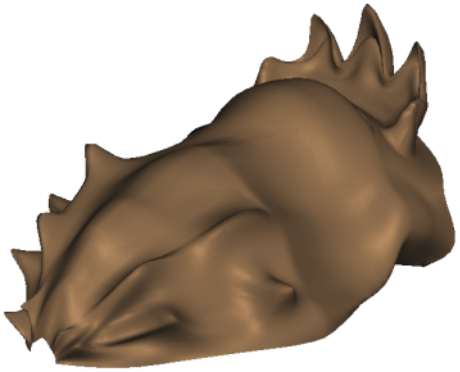
# Section 3 Modeling & Deformations

# Tutorial 4: Surfaces

Burt is an alien creature. He provides an introduction to the various tools for creating and modifying surfaces.

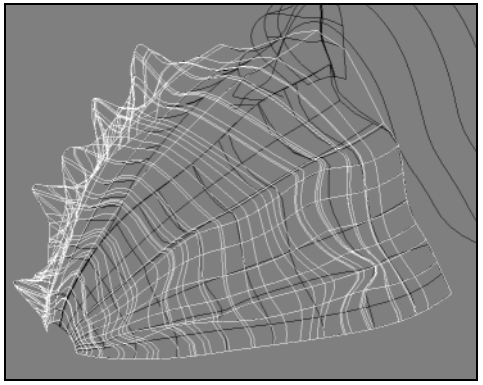
This tutorial shows you how to:

- Create surfaces using Curve Net, Loft, Merge, and Fillet.
- Modify surfaces using Clean, Extend to Curve, and Stitch.
- Use layers to organize your scene.
- Create seamless NURBS surface meshes.

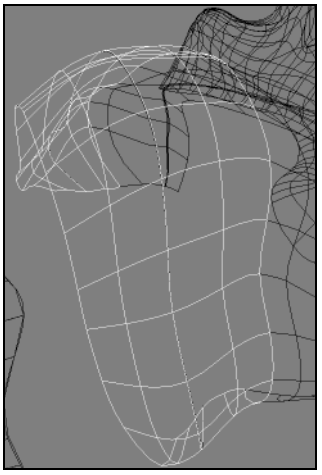


# Overview

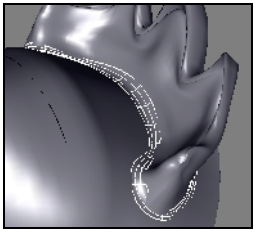
**1 Create Burt's head using Curve Net, Clean, and Merge.**



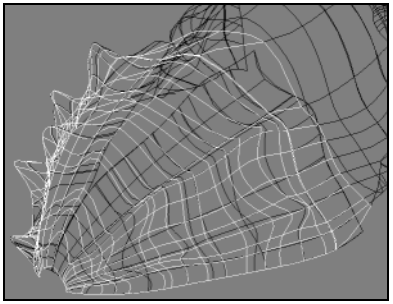
**2 Create Burt's middle using Loft and Extend to Curve.**



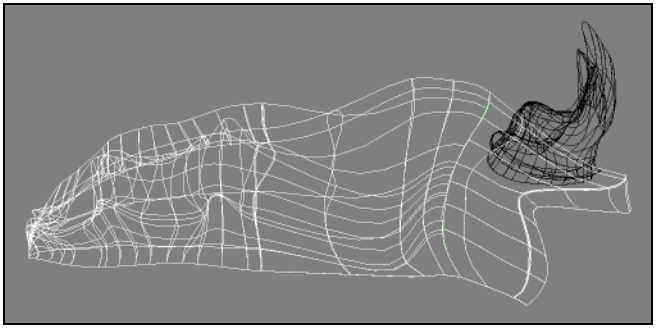
**3 Create a fillet between Burt's back and spines.**



**4 Stitch Burt's head to his middle.**



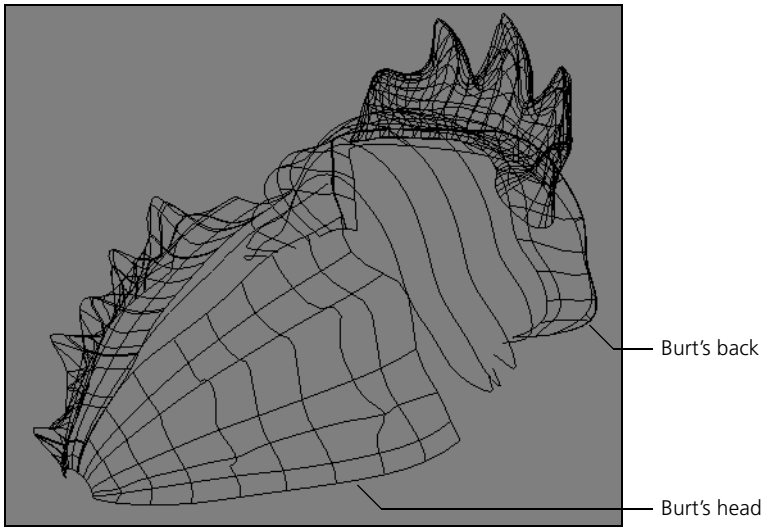
**5 Assemble a single surface out of Burt's head, middle, and back.**



## Creating a Surface with Curve Net

You'll use the Curve Net tool to create a surface section of Burt's head. Curve Net creates a surface from two sets of curves: a set in U and a set in V.

1. Open the **burt** scene from the tutorial database: **<install directory>\content\tutorial_project\Scenes**. The Burt model (shown below) comprises the alien's head and back.

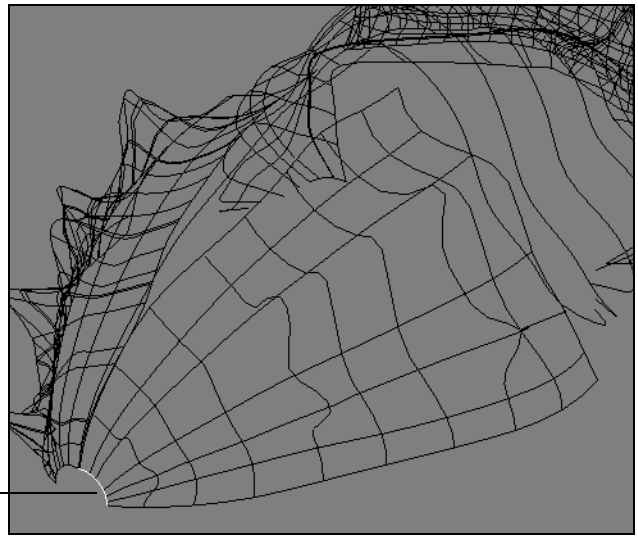



Pressing the **g** key hides or displays the grid for the viewport that currently has focus. This is the one under the mouse pointer if **Give Windows Focus When The Mouse Enters Them** is on in the **General** page of **File** > **User Preferences**.

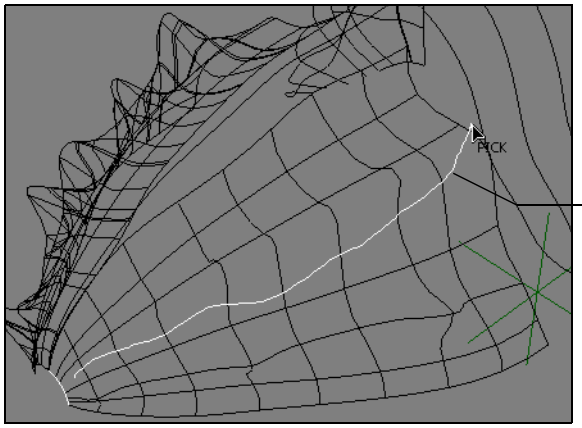
2. Choose **Selection** > **Free Form Tool** from the Selection panel in the main command area. The Free Form selection tool lets you select scene elements by drawing a line across them.
3. Select the first curve at the tip of the head. You may need to zoom or dolly in a little first:
   - To track from side to side or up and down, hold down the **z** key while dragging the left mouse button.
   - To zoom in, hold down the **z** key while middle-clicking.
   - To zoom out, hold down the **z** key while right-clicking.

- To dolly in or out, hold down the **p** key while dragging left or right. Use the left, middle, and right mouse buttons to dolly at a slow, medium, or fast speed respectively.
- To orbit in the perspective view, hold down the **o** key while dragging. Use the left mouse button to orbit freely, the middle mouse button to orbit horizontally, and the right mouse button to orbit vertically.

If you pressed and released the **z**, **p**, or **o** key too quickly, you may have activated the zoom, dolly, or orbit tool in sticky mode instead of supra mode. If this is the case, press and release the space bar to return to the selection tool before selecting the first curve.

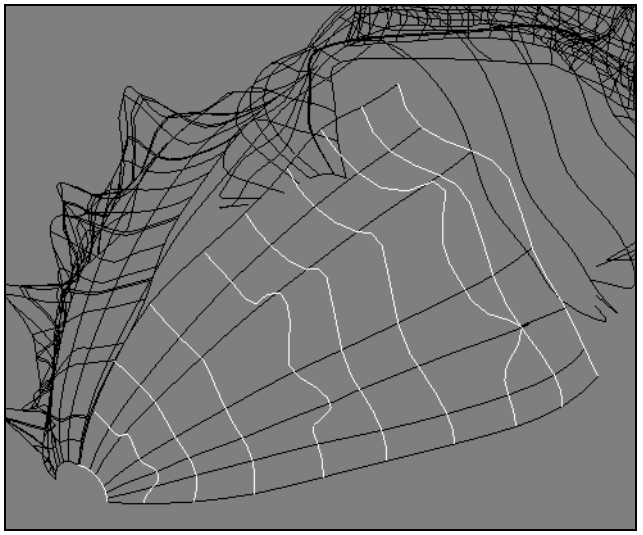



4. From the Model toolbar, choose **Create > Surface > Curve Net**. The status bar at the bottom of the screen prompts you to pick curves.
5. Draw a freeform line (with the left mouse button) to select all the other curves that cross the head lengthwise, starting with the curve next to the first one.



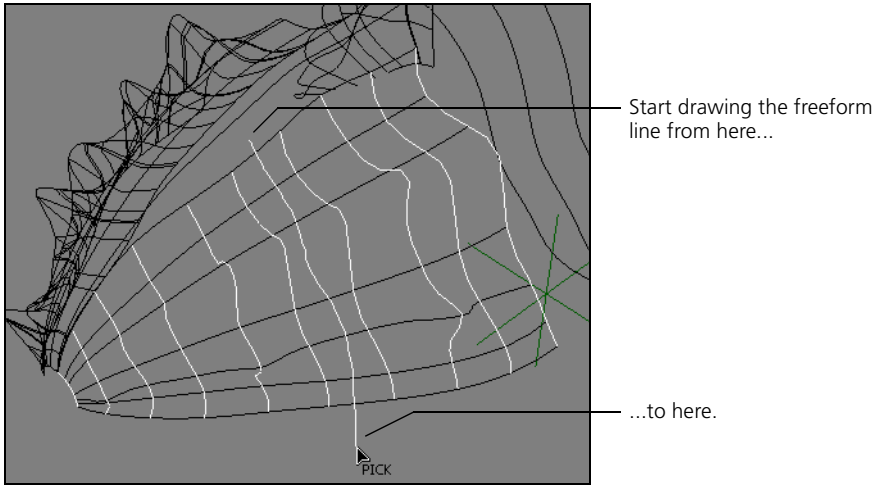

The white line shows the freeform path used to select the curves.

If you make a mistake, Ctrl+click to “unpick” the last curve. Repeat to “unpick” successive curves.

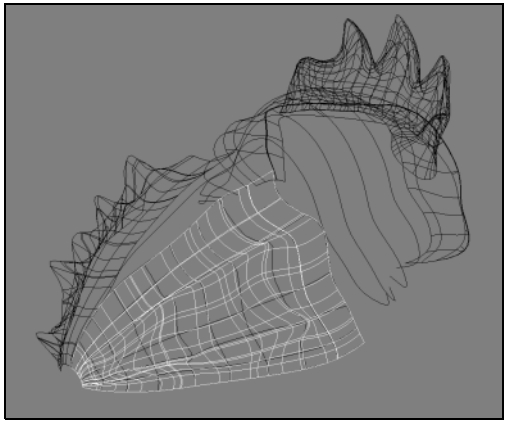


6. Right-click when you have finished picking the first set of curves and are ready to select the next set of curves.

7. With the left mouse button, draw a line across the head section to select all the curves in the other direction beginning with the curve near the center of the head (don't pick the curve in the center just yet).



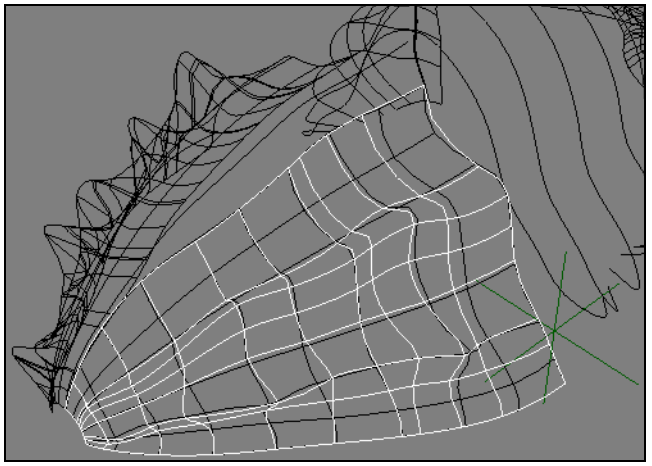

8. Right-click to finish the surface. A new surface is created with the default name **surfmsh** and the Curve Net property editor opens. Leave the values at their defaults.



## Cleaning a Surface

Because the resulting surface may have too many points, the Clean tool allows you to control the number of points on your surface.

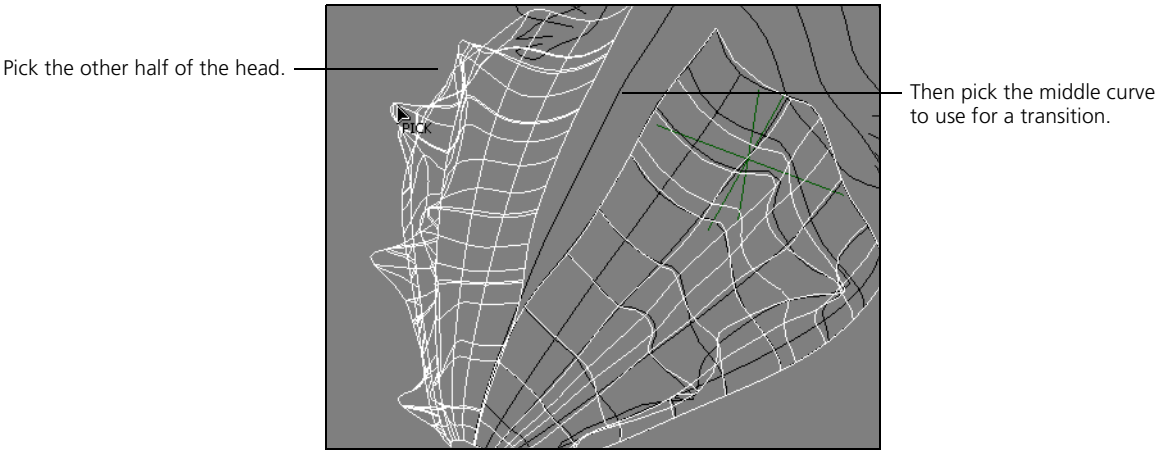
9. With the half-head surface (**surfmsh**) still selected, choose **Modify** > **Surface** > **Clean** from the Model toolbar.
10. Make sure that both **Clean in U** and **V** are on, and modify the **Tolerance** values until the balance between the amount of detail and the number of subdivisions (the "heaviness" of the geometry) is to your liking.



## Merging Surfaces

The next task is to merge the new surface to the other half of the head, using a curve for the transition. The Merge Surfaces tool creates a new surface that spans the two original ones.

11. With the new surface (**surfmsh**) still selected, choose **Create** > **Surface** > **Merge** from the Model toolbar.
12. Pick the other half of the head by clicking on it.




You can cancel any mode or tool you are using by pressing the Esc key.

13. Pick the middle curve to act as an intermediate between the two surfaces. A new surface is created with the default name **surfmsh1**, and the Merge Surfaces property editor opens.

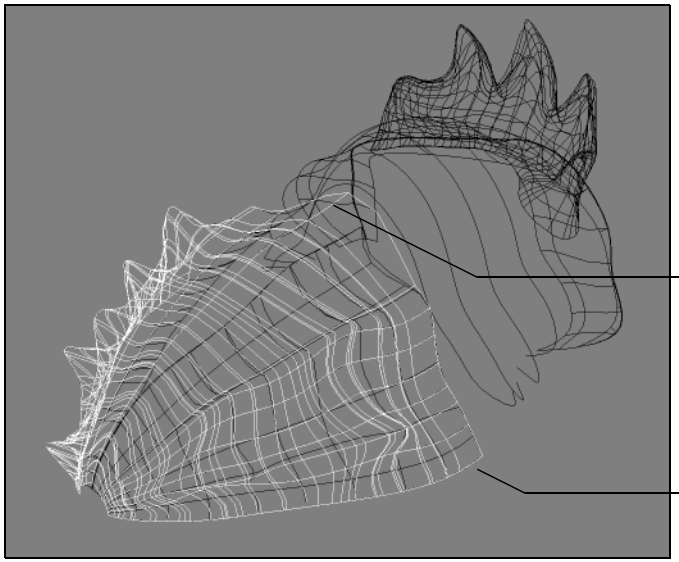
    At first, the new surface may look completely different from what you intended because the wrong boundaries were merged—you will fix this in the next step.

14. In the Merge Surfaces property editor, set the boundaries so that both surfaces merge together via the middle curve.

    - The options on the **Boundaries** page determine which boundaries of the original two input surfaces are merged. For this example, set **Surface 1** to **Min U Boundary** (**Red**) and **Surface 2** to **Min V Boundary** (**Green**). Red and green refer to the boundary colors that are displayed if **Show** > **Boundaries** is on in a viewport (**View** > **Boundaries** for all viewports) and the input surface is selected.

    - On the **Shape** page, set **Seam** to **Curve**. This specifies that the seam between the two input surfaces is the curve you picked earlier.

All of the viewport functions (Pan, Orbit, Shaded mode, etc.) are available while a property editor is open and being modified.

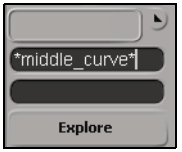



15. Click on the **Clean** tab and adjust the **Tolerance**. When cleaning a surface, try to obtain the least number of subdivisions (resolution) without losing the original shape.

16. Close the property editor.

## Relational Modeling

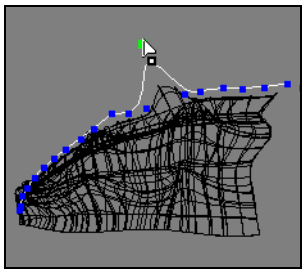
Relational modeling allows you to modify a surface using the curves from which the surface was built. For example, you can create a glass by revolving a profile and relational modeling allows you to edit the glass by moving points on the original profile curve. Relational modeling is SOFTIMAGE|XSI's default behavior.




17. Enter ***middle_curve*** in the Selection text box in the Selection panel of the main command area and press Enter. This selects any scene element with the string **middle_curve** in its name. In this example, it selects the **creature_demo_middle_curve** curve—the one in the middle of the head.

18. Move the mouse pointer over the perspective view and press F12 to enlarge it to full screen. Press **f** to frame the selected curve.

19. Use the **m** key to interactively move points on the curve, or tag points with the **t** key and translate them. Notice how the merged head changes shape in response to the new shape of the curve. Notice also that you can select the merged head and move points on it—these changes are preserved when you reselect the curve and move points on it.



## Freezing the Operator Stack

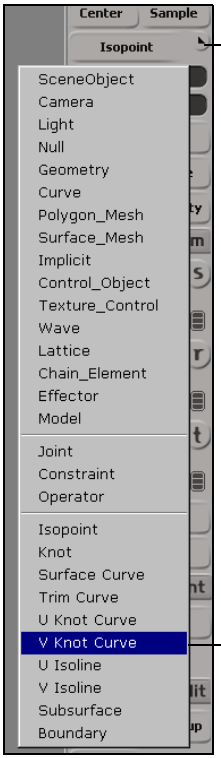
Freezing the operator stack collapses the surface's construction history and breaks the modeling relation with the input curves. It is as if the surface was imported "as is" directly into SOFTIMAGE|XSI.

20. Select the merged head surface that you just created (**surfmsh1**).
21. From the Edit panel of the main command area, choose **Edit** > **Freeze Operator Stack**. This removes the operator history of the surface, including the modeling relations.
22. Try selecting the curve in the middle of the head and moving points again. Notice how the merged head surface is no longer affected by changes to the curve.

## Lofting

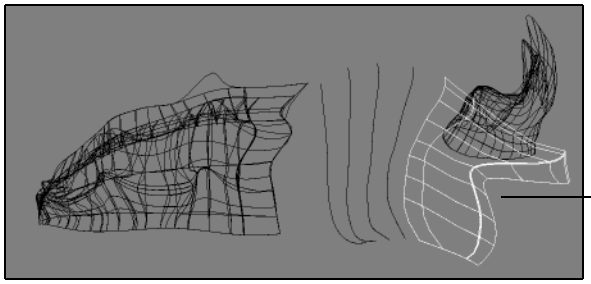
Loft is a new tool that allows you to create a surface out of a series of curves, similar to the Skin function of SOFTIMAGE|3D. However, with the Loft command you don't need the same number of points on each curve. You can also loft directly from a curve, isoline, boundary, or knot curve to another curve, isoline, boundary, or knot curve.

23. With the mouse pointer over the viewport, press F12 again to return to the four-port view.
24. Select the back of the body (**skin245_1**).




Click the triangle to choose a selection filter.

Choose the V Knot Curve filter.

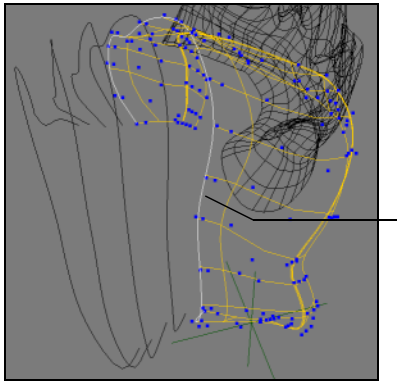


Burt's back

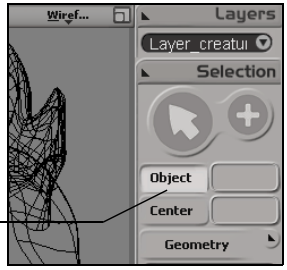
25. Choose **V Knot Curve** from the Selection Filter list (above the Selection text box in the Selection panel of the main command area).

A knot curve is a curve defined by knots on a surface. By default these curves are not visible but can still be selected with the knot curve selection filter.

26. Select the first V knot curve on the back surface.

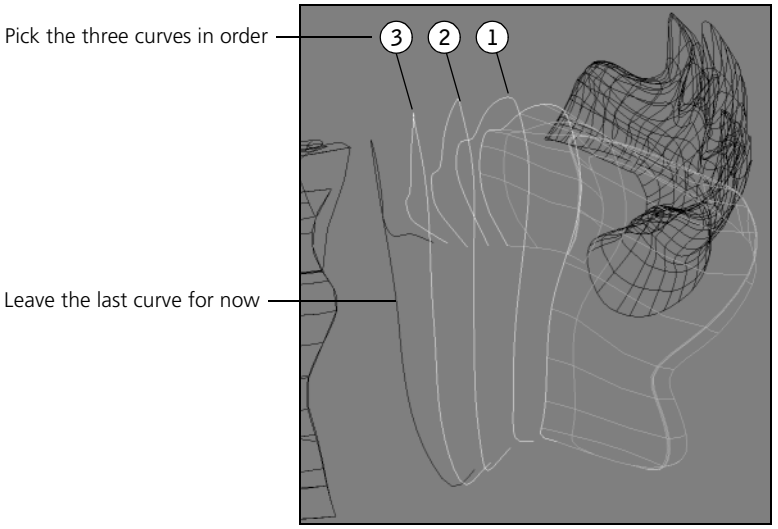


Selected V knot curve in white.

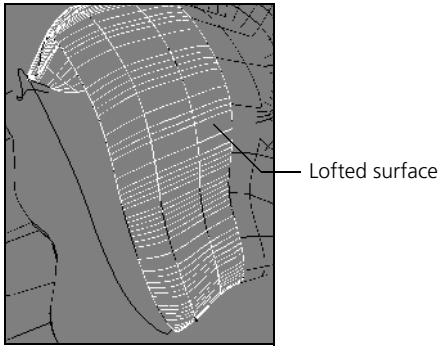



Object selection filter button

27. From the Model toolbar, choose **Create** > **Surface** > **Loft**.
28. Switch back to the **Object** selection filter before picking curves. Alternatively, you can choose **Curve** from the Selection Filter list but don't forget to change it when you want to select something else.
29. Pick the first three of the free-floating curves in order, as in the illustration below. Don't pick the last curve—you'll add it in a later step.




30. Right-click to finish picking. A new surface is created (by default named **surfmsh2**) and the Loft property editor opens. Leave the values at their defaults.



## Using the Extend-to-Curve Command

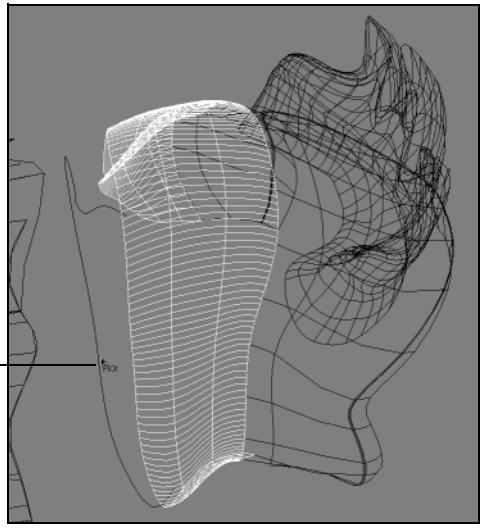
You can now attach the last curve to Burt's back. As the name implies, the Extend to Curve tool extends the selected surface to include a chosen curve.

31. Make sure that only the lofted section (**surfmsh2**) is selected.

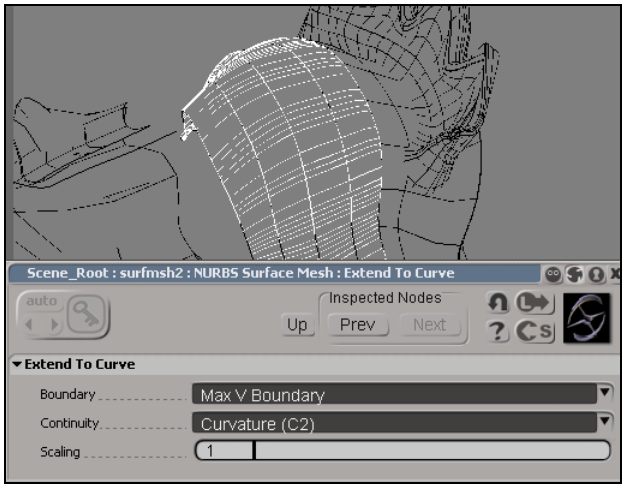
You can check the Selection text box to validate your selection—if you have more than one selection, it displays MULTI (nn).

If more than one object is selected, you can use the Selection button in the Selection panel of the main command area to refine the selection. Click the Selection button, and a pop-up explorer appears listing all selected elements. Click on an individual element to select only it.

32. Choose **Modify** > **Surface** > **Extend to Curve** from the Model toolbar.
33. Pick the last remaining curve to join it to the alien's back surface. The Extend to Curve property editor opens.




34. If the wrong boundary was extended to the curve, set **Boundary** to **Max V Boundary**.

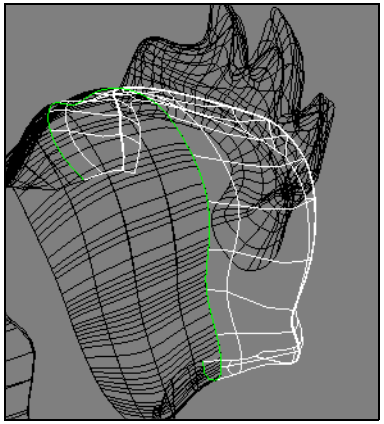



35. Clean the surface by choosing **Modify** > **Surface** > **Clean** (as described on page 53).

## Filleting Intersections

The Fillet Intersection tool lets you create a seamless surface between two intersecting objects; in this case, the back and the spikes.

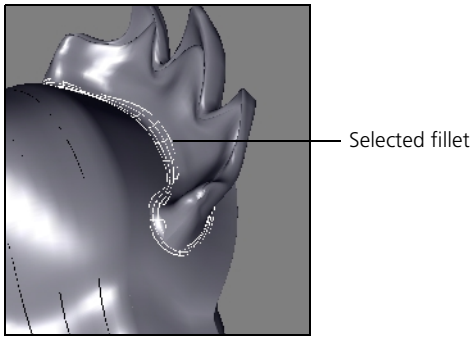
36. Select the back section (**skin245_1**)once again.



37. Choose **Create** > **Surface** > **Fillet Intersection** from the Model toolbar.
38. Pick the spiked surface on the model's back with a left click. The fillet is created (default name **surfmsh3**) and the Fillet Intersection property editor opens.

To tweak your fillet parameters, you may find it helpful to change your viewport display to Shaded view.

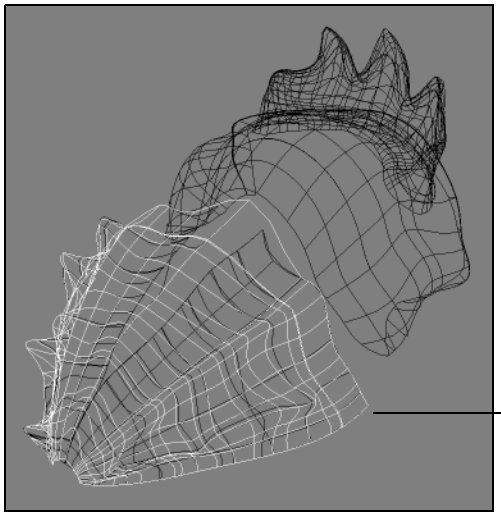
39. Adjust the parameters in the Fillet Intersection property editor. You can set the number of U and V subdivisions as well as the radius of the fillet. If you set the radius too high for the intersecting objects, it becomes impossible to calculate the fillet and the result is a degenerate surface.



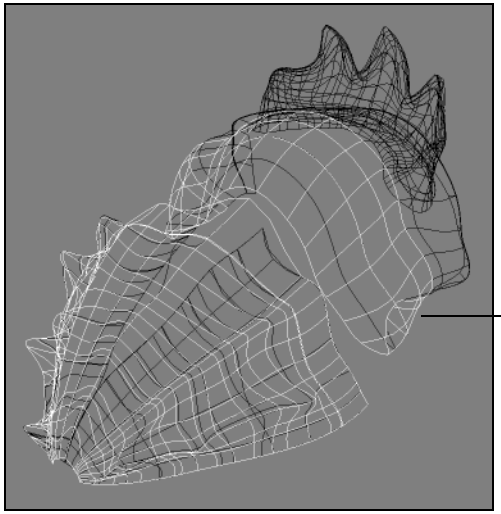

## Stitching Surfaces Together

The Stitch tool seamlessly "stitches" surfaces together, unlike a Merge which creates a new object. This tool brings several surfaces' boundaries together without creating new knot curves.

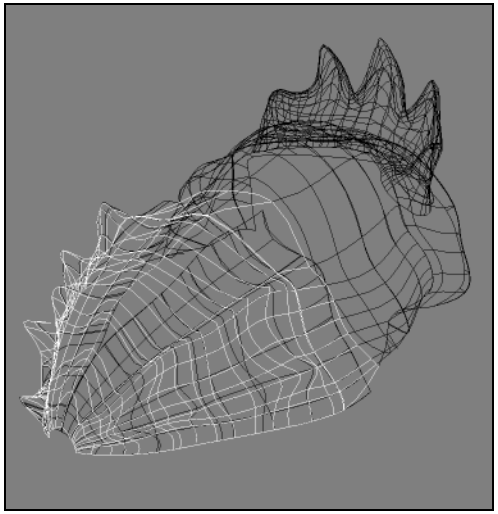
40. Select the head surface of the alien (**surfmsh1**).




41. Choose **Modify** > **Surface** > **Stitch** from the Model toolbar.
42. Select the middle surface (**surfmsh2**) previously created with the Loft tool.



43. Right-click to complete the stitch operation and open the Stitch property editor.
44. In all likelihood, the wrong boundaries were stitched together. For this example, set both options on the **Boundaries** page to **Max V Boundary**—the back boundary of the head is deformed to the front boundary of the middle surface. You can also adjust the **Tolerance** on the **Clean** page to reduce the number of isolines. Close the Stitch property editor.



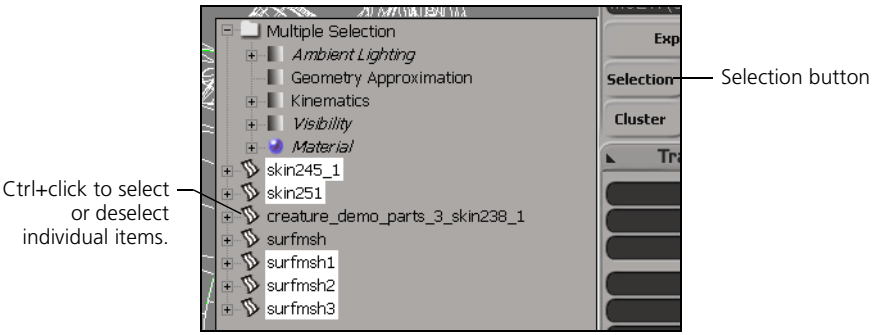
45. Translate the head a little using the **v** key to see how the surfaces remain stitched together. Undo the translation with **Ctrl+z**.

## Organizing the Scene with Layers

As objects, surfaces, and curves start to accumulate, your scene can start to get heavy. You can use layers to hide some elements and display only others.

46. Select the merged head (**surfmsh1**), lofted middle (**surfmsh2**), back (**skin245_1**), spines (**skin251**), and fillet (**surfmsh3**). To do this quickly:
    - Choose **Selection** > **Rectangle Tool** from the Selection panel of the main command area.
    - Choose **Surface_Mesh** from the Selection Filter list. This prevents you from selecting any curves or other types of objects.
    - Click and drag to define a rectangle around Burt. If **Selection** > **Select Single Object in Region** is on, hold down the Shift key while dragging.

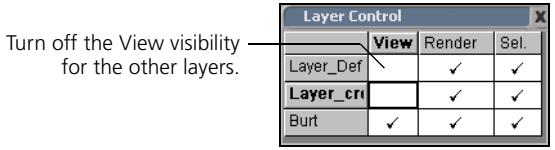
- Click the Selection button on the Selection panel, then Ctrl+click on individual items to refine the selection list.




47. From the Layers panel in the main command area, choose **Layers** > **New Layer** and name your new layer. This creates a new layer including the active selection.

48. Choose **Layers** > **Layer Control** (or press the **6** key). The Layer Control box opens. From here you define the visibility, renderability, and selectability of each layer.

49. Turn off **View** for all layers except the layer you just created. Close the Layer Control box.

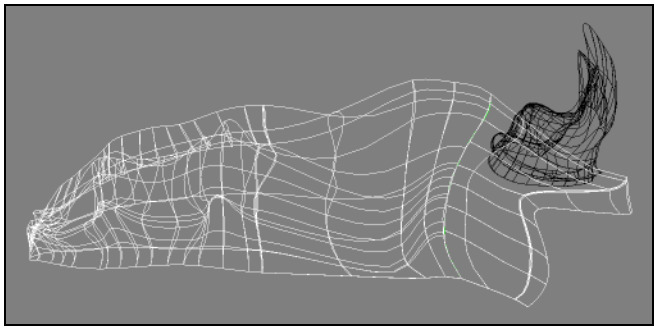



Use the ‘ key (under the Esc key on most keyboards) to quickly close the active property editor.

## Assembling Seamless Surfaces

The Assemble tool creates a new surface mesh based on the selected surfaces. A surface mesh is a single 3D object composed of multiple subsurfaces. Although the result may look like a single surface, you can still pick the individual component subsurfaces.

50. Select Burt’s back, middle, and head surfaces.



51. Choose **Create** > **Surf Mesh** > **Assemble** from the Model toolbar. The Assemble NurbsMesh dialog box opens.
52. Accept the default values and click OK.
53. The Surface Mesh property editor opens. Give the new surface mesh a name if desired, then close the property editor.
54. From the Selection Filter list, choose **Subsurface**.
55. Select the alien's head and translate it a few units away. Notice how there is no continuity between the subsurfaces yet.

### Using Surface Continuity Manager

56. Turn the **Object** selection filter back on in the Selection panel of main command area. The assembled surface mesh is automatically selected.
57. Choose **Create** > **Surf Mesh** > **Continuity Manager** and leave the options at their default values.
58. From the Selection Filter list, choose **Subsurface** again. The last component selected, in this case the head, is automatically selected again.
59. Translate the head a few more units away to see how the continuity between surfaces is maintained.

## Conclusion

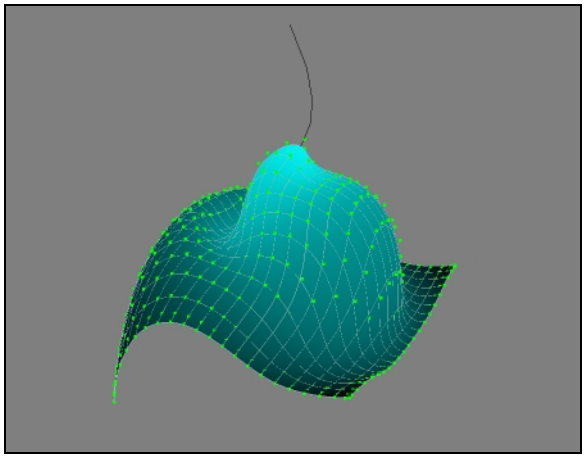
This tutorial introduced you to a few of the modeling tools available in SOFTIMAGE|XSI. There are many other tools for creating and modifying surfaces, including Revolution, Extrusion, Birail, and so on. See the *Modeling & Deformations* guide for more information.

# Tutorial 5: Weight Maps

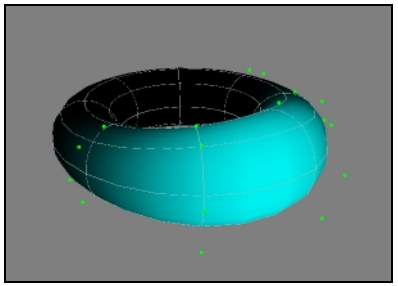
Weight maps are used to precisely distribute the intensities of constraints and/or deformations. You can also use them to define the envelope weighting of a character.

This tutorial shows you how to:

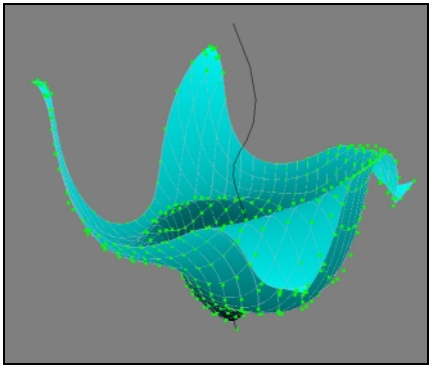
- Use weight maps to modulate the amplitude of deformations.
- Paint weight values on maps.
- Work with Push, Curve, and Spine deformations.
- Assign envelope weights by painting.



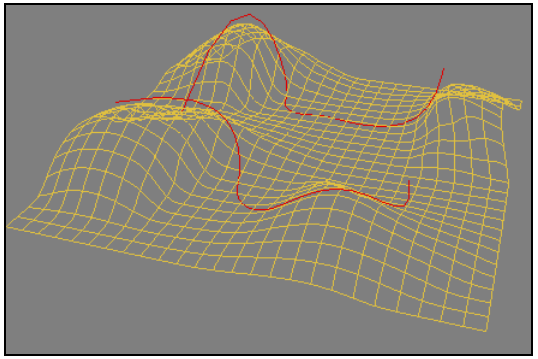
# Overview



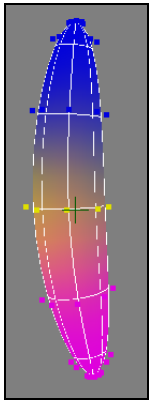
1 **Paint a push deformation.**



2 **Control a deformation by curve with a weight map.**



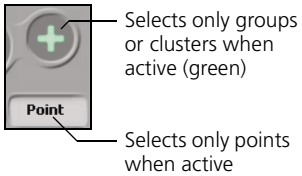
3 **Modulate a deformation by spine.**



4 **Paint envelope weights.**

## Using Weight Maps and Deforms

Weight maps allow you to modify the amplitude and zone of influence of an effect. In this example, you will apply a weight map to a torus and deform it with a Push operator. You will then adjust the deformation by painting on the weight map with the General Attribute Painting (GAP) tool.




1. Start with a new scene by choosing **File** > **New Scene** or pressing Ctrl+n.
2. Create a torus by choosing **Get** > **Primitive** > **Surface** > **Torus** from the Model toolbar. Leave the size and geometry at their default values.
3. Tag all the points on one half of the torus by pressing and holding **t** while dragging the pointer over them. The weight map will be applied to the tagged points. If you don't tag points first, the weight map will be applied to all the points on the object.
4. Choose **Get** > **Property** > **Weight Map**. This defines a cluster with the selected points and creates a weight map on it. The Weight Map property editor opens.

   To see the weight map in a viewport, make sure that the display type is set to Constant and that **Show** > **Weight Maps** is on. These options are set automatically when you paint on a weight map.
5. From the Model toolbar, choose **Modify** > **Deform** > **Push**. The Push property editor opens. Because the weight map was selected when you applied the deformation, it is already connected to the **Amplitude** parameter, as indicated by the red connection icon.

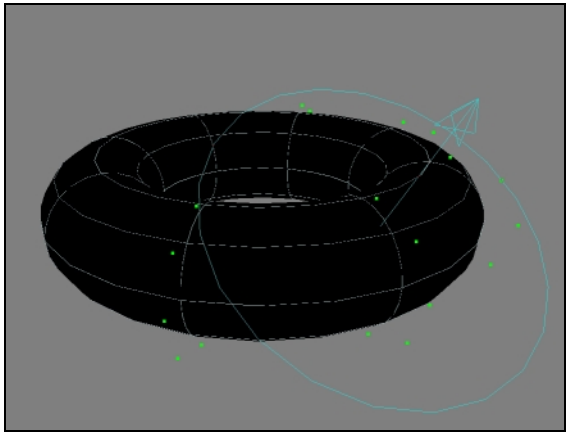
You can double-click on the title bar to minimize or expand the property editor if it's in your way.

6. Set the **Amplitude** to 2. This pushes the cluster points 2 units in the normal direction.
7. On the Weight Map Generator page, experiment with the various parameters. For example, you can select various predefined weight maps such as linear and radial ramps for the **Weight Map Type**.
8. Before proceeding to the next section, create a blank weight map for painting your own strokes: set **Weight Map Type** to **Constant**, **Base Weight** to 0, **Weight Value Range - Maximum** to 1, and **Minimum** to –1. Close the Weight Map property editor.

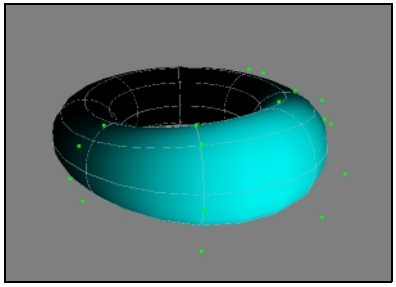
## Painting Weights

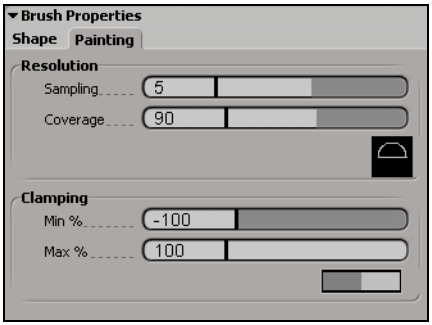
The Paint tool allows you to paint attributes, such as envelope weights, deformation weights, etc., onto an object's surface.

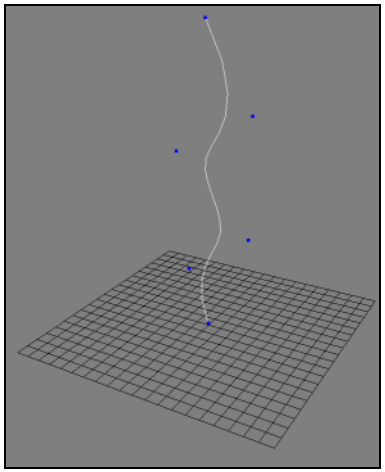
9. With the weight map still selected, press the **w** key to activate the Paint tool.



10. With the mouse pointer over the Camera viewport, press F12 to enlarge it to full screen.
11. Middle-click and drag to adjust the brush radius.You can also "bump" the brush radius up or down in increments using the up and down arrow keys.
12. Experiment with a few strokes on the surface. The left mouse button paints positive values and the right mouse button erases or paints negative values.





Press Ctrl+w to display the Brush property editor. The **Coverage** parameter on the Painting page defines how the brush wraps around a three-dimensional object. This allows you to paint on back and front of an object at the same time. The Coverage icon provides a visual representation of how much the brush wraps around the object, like a cross-section.

13. Click **Object** on the Selection panel to return to Object selection mode, then deselect all by clicking in empty space.
14. Select the torus.
15. Click the **Selection** button (beneath the Explore button in the Selection panel) in the main command area. A pop-up explorer opens.
16. Double-click on the **Push Op** icon. The Push property editor opens.
17. Experiment with different **Amplitude** values.

Even though the Amplitude slider stops at 2, you can push the value higher by manually typing any value.

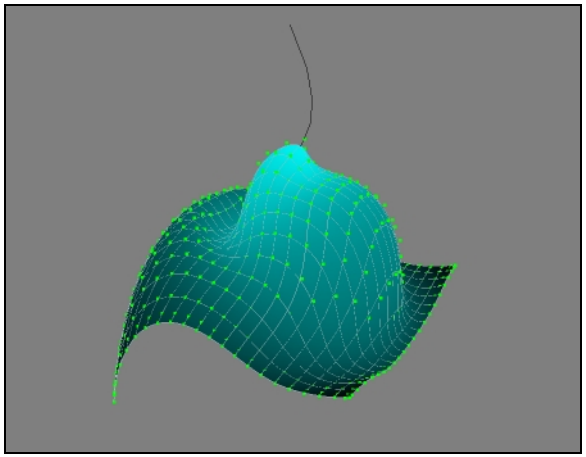
## Weight Maps and Deformation by Curve

1. Press Ctrl+n to start a new scene.
2. Choose **Get** > **Primitive**, then press the **s** key followed by **g** to create a surface grid. Increase the subdivisions to get a higher resolution—the more subdivisions there are, the smoother the deformation will be.
3. Choose **Create** > **Curve** > **Draw CV NURBS** from the Model toolbar.
4. Draw a curved line in the Front view. Create your first point near the grid and the rest in an upward direction (otherwise you may need to invert the curve later on). Make sure the curve is over the grid as shown in the illustration. Press Esc or the space bar to return to the Selection tool when you have finished drawing the curve.

Create a simple, almost linear curve. If the curve's bends are too sharp, the resulting deformation may end up being too distorted for a clear example. You can adjust the curve later on because relational modeling is on by default.

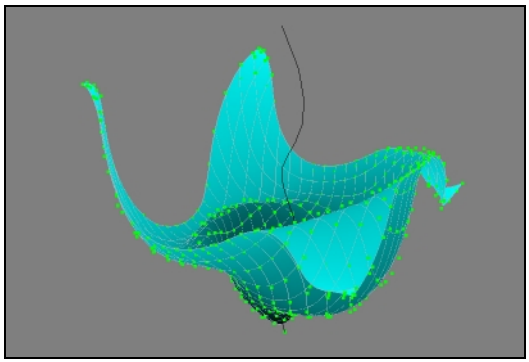
5. Select the grid.
6. Choose **Get** > **Property** > **Weight Map**. This creates a cluster comprising all the points on the grid, and creates a weight map on that cluster. The Weight Map property editor opens.

7. To display property maps when the Paint tool is not active, change the Camera viewport (B) from Wireframe to Constant by selecting it from the drop-down menu in the upper-right of the viewport and make sure that **Show** > **Weight Maps** is on.
8. In the Weight Map property editor, select the **Radial XZ** for the **Weight Map Type** and select **Invert Weights**.
9. Choose **Modify** > **Deform** > **by Curve**.
10. Pick the curve you created in previous steps. The Curve Deform property editor opens.
11. In the property editor, increase the **Translation Along Curve** values until the grid is halfway up the curve. Notice how the curve deformation is more pronounced where the weight map has higher values.



Lock Icon

12. Click on the Lock icon of the property editor, then scroll down to the **Weight Map Generator** page and experiment with the various parameters.



You can paint on top of the generic gradients you select—make sure the weight map is selected before painting on it.

Activate the **Mute** on the Curve Deform page so that you can paint on the undeformed shape.

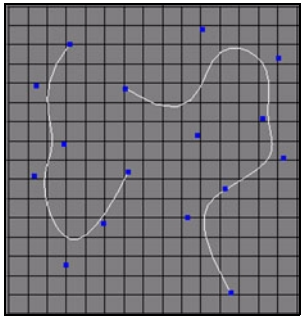
Press Crtl+w to open the Brush property editor.

Move points on the deforming curve to interactively adjust the deformation.

## Weight Maps and Deform by Spine

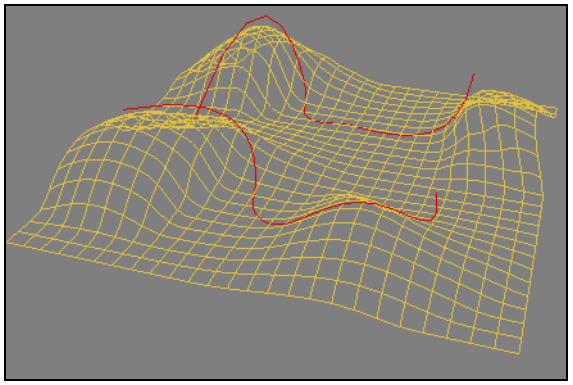
A Spine is basically a curve that can deform a surface.

The Deform by Spine tool automatically creates a weight map—you don't have to create one before using this tool.



1. In a new scene (Ctrl+n), choose **Get** > **Primitive** > **Surface** > **Grid**. Increase the subdivisions to get a higher resolution.
2. Choose **Create** > **Curve** > **Draw CV NURBS** to draw a curve on the grid surface in the Top view. When you have finished drawing the first curve, middle-click on the **Create** > **Curve** button (repeats the previous command) to draw a second curve as shown in the illustration.
3. Press the space bar and select the grid. Choose **Modify** > **Deform** > **by Spine** from the Model toolbar. The status bar prompts you to select curves.

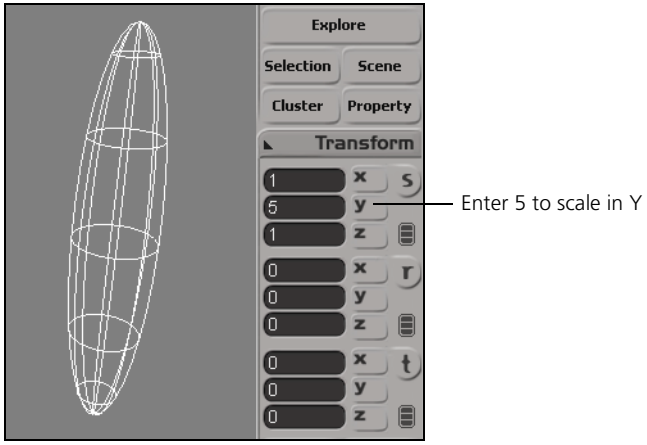
4. Pick the two curves then right-click to terminate the picking session. The Deform by Spine Op property editor opens.
5. Click on the Lock (keyhole) icon to keep this property editor open.
6. Select one of the curves.
7. Press the **m** key, then move points on the curve along the Y axis.

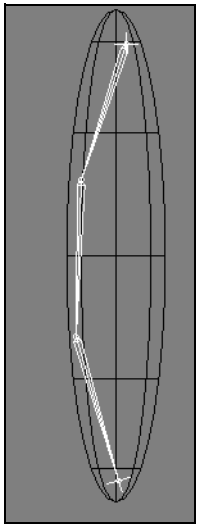


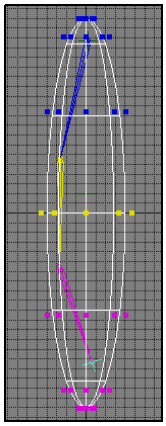
8. On the Deform by Spine Op property editor, adjust the **Falloff Amplitude** profile curve to see the effect:
   - Click to select the curve.
   - Drag the control points to move them. You can also drag the handles to adjust the tension.
   - Press the a key while clicking to add a control point. Press the **d** key while clicking to delete points.
   - Right-click on the profile graph for more options. For example, you can load a preset curve from the **Presets** menu.

## Envelope Weighting

1. In a new scene, choose **Get** > **Primitive** > **Surface** > **Sphere**. Scale it in Y by 5.





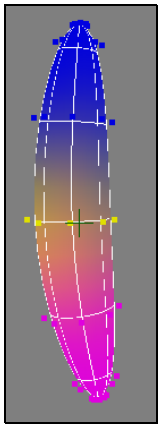
2. From the Model or Animate toolbar, choose **Create** > **Skeleton** > **2D Chain**.
3. In the Front view, draw a three-bone chain inside the sphere by clicking to place the root, joints, and effector. Use the illustration at left for reference. Right-click to finish drawing the chain.
4. Select the sphere.
5. From the Animate toolbar, choose **Deform** > **Envelope** > **Set Envelope**. Click OK to accept the default options.
6. Select the three bones one after the other. Right-click when you have finished picking. The Automatic Envelope Assignment property editor opens—leave the options at their default values.

You can branch-select a chain with the middle mouse button when setting the envelope. You are not obliged to use all the bones in a chain. You can also keep the root and effector excluded from the envelope's deformers.

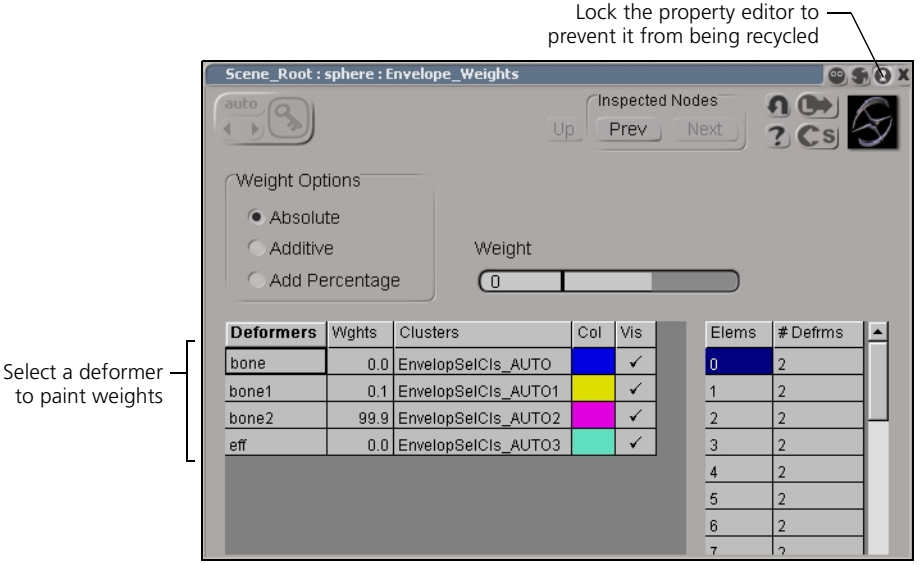
7. Move the effector to see the effect of the automatic envelope weighting.

Points in the envelope are automatically weighted between the deforming bones. The points' colors indicate how they are weighted between the deformers.

8. Select the sphere and frame it in the Camera view by pressing the **a** key.



9. Press F12 with the pointer in the camera viewport.
10. Press the **w** key to activate the Paint tool. The Display mode temporarily switches to Constant and **Show** > **Weight Maps** is turned on.
11. Drag left or right with the middle mouse button to adjust the brush diameter, or press Ctrl+w and experiment with other brush properties.
12. Press Ctrl+e to open the Envelope Weights editor.




13. Click on the Lock (keyhole) icon at the top of the property editor and leave the editor open.
14. Select a bone under Deformers, then click and drag to use the Paint tool to weight points toward the corresponding bone. Use the **o** supra key to orbit as you work.
15. Continue painting weights for different bones until you are satisfied with the weighting. When you have finished painting, press Esc or the space bar to return to the Selection tool.
16. Move the effector to see the effect of your new weight map.

## Conclusion

Deformations are a powerful and flexible way to control the shape of objects, and weight maps give you even more control. You can paint weights onto maps to create exactly the shape you need.

See the *Modeling & Deformations* guide for more information.

# Section 4 **Animating**

# Tutorial 6: Constraints, Expressions & Particles

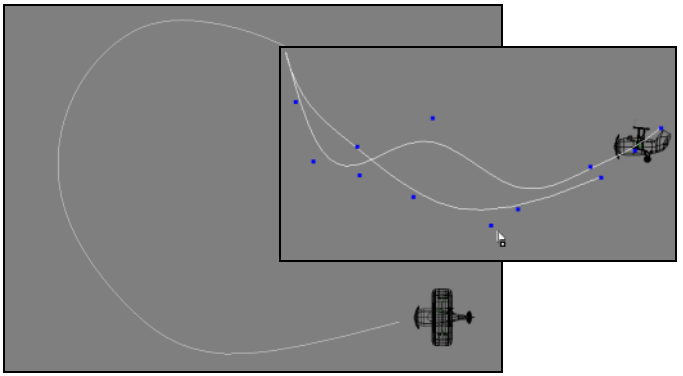
Constraints, expressions, and particles are three ways of animating in SOFTIMAGE|XSI without directly manipulating an animated object. Constraints and expressions allow you to base one object's motion on another object's shape or behavior, while you use particles to create effects such as smoke and sparks by animating certain parameters over time.

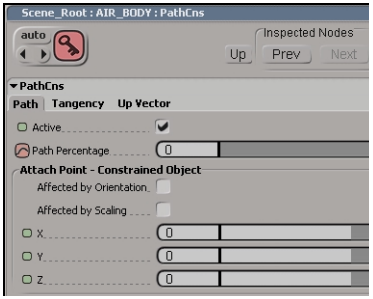
This tutorial shows you how to:

- Animate an airplane along a path.
- Create a second airplane and animate it using expressions to guide its behavior as it travels along the path.
- Create a smoke effect for both airplanes.

# Overview

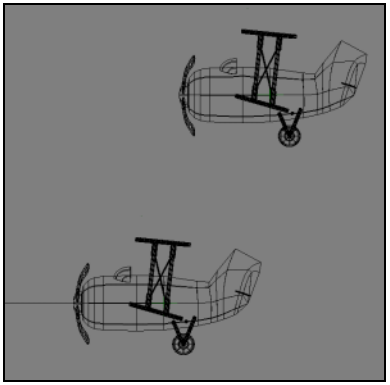
**1 Create and modify the path.**



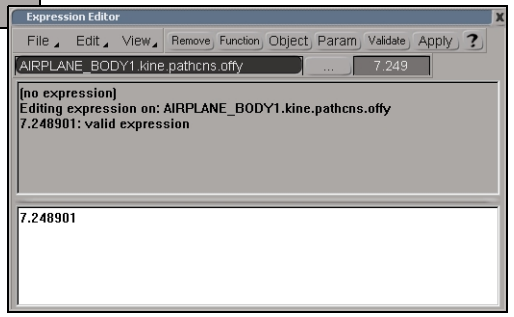



Animate the plane on the path and its tangency and roll.

**2 Duplicate the airplane.**

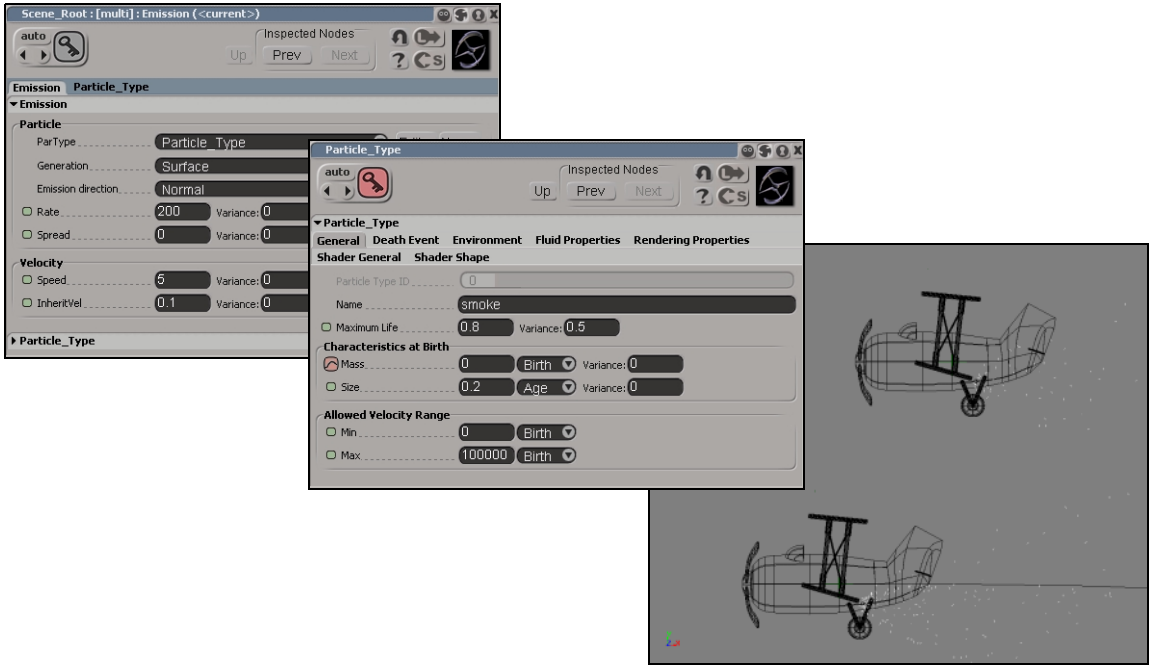



Use compensation to offset the duplicate airplane.




Create an expression to control the plane's behavior as it travels along the path.

**3 Create the particle animation.**

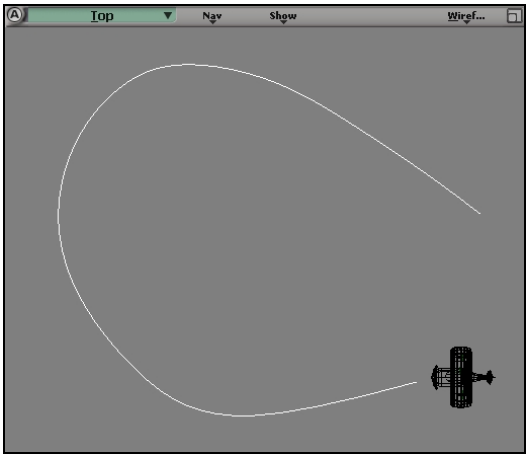


## Animating on a Path

In this example, you will animate an airplane along a path and keep its nose pointing forward by constraining it to the tangency of the curve. You will also duplicate the airplane, preserving its path and tangency constraint, and use compensation to offset the duplicate from the original.

1. Load the AIRPLANE scene from the tutorial database:
<**install directory**>**\content\TUTORIAL_PROJECT\Scenes**.

2. In viewport A (Top), zoom out until there is enough space in front of the airplane to draw its animation path.




*Draw a curve to be used for the path animation*

This curve will be the path which will "drive" the airplane.

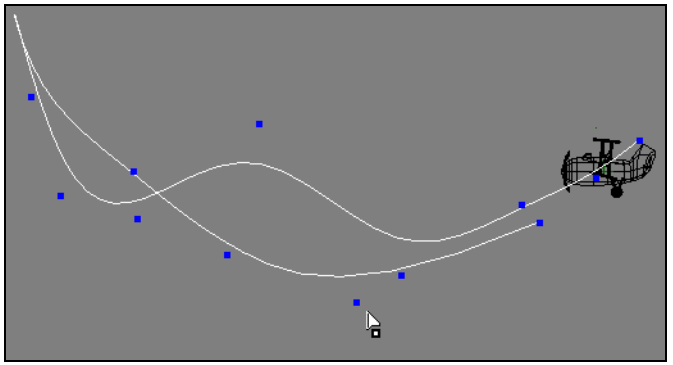
3. In either the Model or Animate toolbar, choose **Create** > **Curve** > **Draw CV NURBS**. Draw a path for the airplane to follow in the Top view, as shown in the illustration at left.

*Modify the curve in Y to animate the airplane altitude*

4. In the Front view, press the **f** key to frame the curve.

5. Press the **m** key to switch to Move Point mode.

6. Move some points along the Y axis to set the airplane's change in altitude.



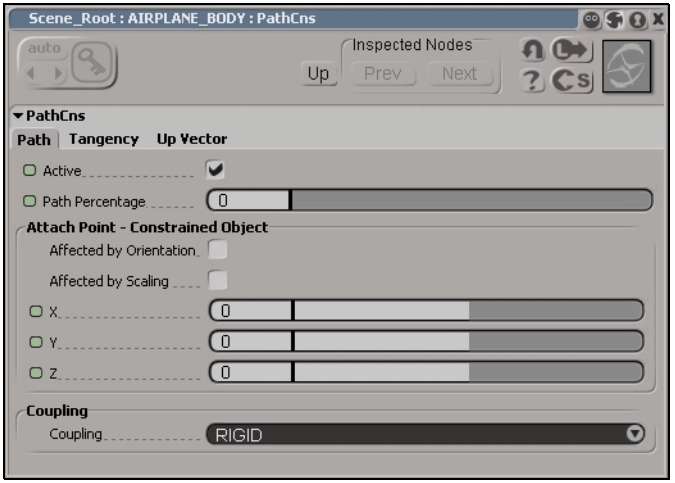
As an alternative to move-point mode, you can use the **t** supra key to tag (select) points on the curve and the **v** supra key to translate them.

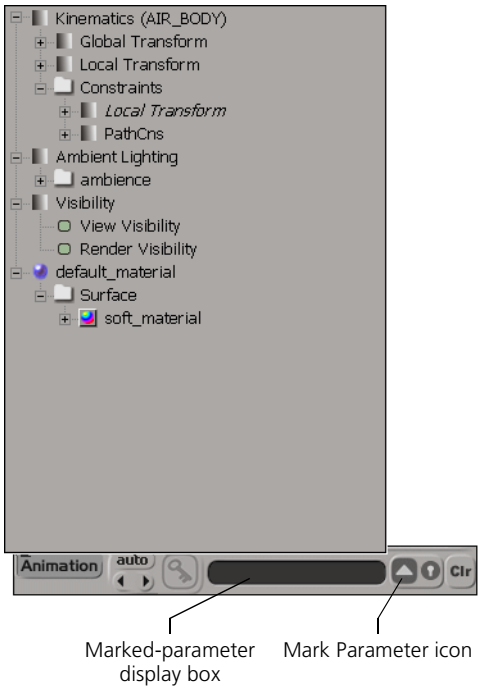
7. Press the space bar to return to select mode.

8. In the Top view, right-click the airplane to select the entire hierarchy.
9. Choose **Constraint** > **Curve** (**Path**) from the Constraint panel in the main command area.
10. Pick the curve.

    The PathCns property editor appears and displays the **Path Percentage** control. This is where you animate the constraint according to its progression along the path.

    The **Tangency** and **Up Vector** tabs give you direct access to related constraints.






## Keyframing Marked Parameters

In the next few steps, you will mark and animate the **Path Percentage** and **Up Vector Roll** constraints. Marking lets you create keyframes for specific parameters, as opposed to keyframing all parameters in a property editor.

When you click the Mark Parameter icon (the arrow) in the Animation panel at the bottom of the main SOFTIMAGE|XSI window, it displays a list of parameters that can be directly marked from the selected elements. If a single parameter is marked, its name is displayed in the Animation panel.

11. Change the end frame to 200 in the End Frame box at the right end of the timeline.


12. In the PathCns property editor, select the **Path** tab, then mark **Path Percentage** by clicking on its name. The name is highlighted in yellow.
13. On the **Up Vector** page, select the **Active** option and mark the **Roll** parameter by Ctrl+clicking on its name.

Ctrl+clicking a parameter name toggles a parameter's marking without affecting other marked parameters.

14. At frame 1, make sure that **Path Percentage** on the Path page is at 0 and **Roll** on the Up Vector page is at –360. Click the keyframe icon at the top of the property editor to set a key on those two marked parameters.
15. Move to frame 200 in the timeline. Change the **Path Percentage** to 100% and **Roll** to 360. Click the keyframe icon again.
16. Click the **Tangency** tab and select the **Active** option to activate the tangency constraint. Change the **Axis to Align** on **X** to –1. Close the PathCns property editor.
17. In viewport B, switch to User view and Ctrl+click the curve. Press Shift+f to frame the airplane and the curve in the viewport.

As an alternative to **Constraint** > **Curve** (**Path**), you can use the **Create** > **Path** > **Set Path** command. This prompts you for the start and end frames for quick path settings.

## Duplicating the Object with Its Constraint

18. Ctrl+click the curve again so that only the airplane is selected, then press Ctrl+d to duplicate it. This duplicates the object and its constraints but since the duplicated airplane is in the same position as the original, there is no visible change in the 3D views. You can see the duplicated airplane in the Explorer view in viewport D.
19. Play the animation: The two planes follow the same path.

## Constraint Compensation

20. Wouldn't it be better if the airplanes were not on top of one another? Click the **Comp.** button in the Constraint panel on the main command area, then click the translate (**t**) button in the Transform panel. Translate the second airplane to another location and deselect the **Comp.** button.

    The compensation tool lets you interactively create an offset to a constraint. You can use compensation to reposition a constrained object or use compensation before applying a constraint and get the offset from its relative position.

Do not animate while in compensation mode because you would be keyframing the offset rather than the path translation.

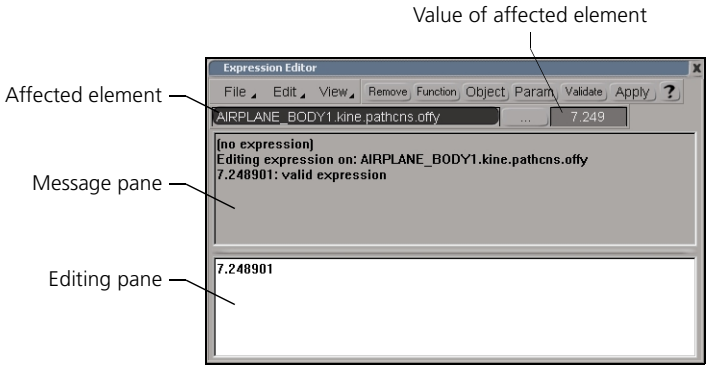
21. Play back the animation. The duplicated airplane remains offset from the original.

You can access the constraint's property editor from the explorer to modify the constraint. Select the object, open the explorer, and use the selection filter option by choosing **Scene** > **Selection** from the explorer menu. Expand the object node, and select **Kinematic** > **Constraints** > **PathCns**.

## Adding Expressions

22. With the duplicated airplane selected, click the **Property** button in the Selection panel.
23. Click the **PathCns** icon (**AIRPLANE_BODY1** > **Kinematics** > **Constraints** > **PathCns**) to open the constraint property editor.
24. Select the Path tab, right-click on the animation icon (the green box) of the **Y** parameter under **Attach Point - Constrained Object**, and choose **Set Expression.** The expression editor opens.

    The expression editor window has two main panes: the message log in the upper pane and the editing pane below it. The affected (target) element is listed in the text box with its current value on the right.




25. Delete the value displayed in the editing pane, then from the expression editor's command bar choose **Function** > **Profiles** > **Sinus** to create an undulating wave.
26. Replace <**period**> (including the angle brackets) with **Fc** (current frame).
27. Replace <**amplitude**> with **5**.
28. Replace [<**offset**>] with **100**. You should now have **sinus( Fc, 5, 100 )**. This causes the second plane to have a varying offset from the path, based on a sine wave.
29. Click **Apply**. The message window logs the validation of the expression on the affected element. Experiment with different values for the period, amplitude, and offset to see how it affects the plane's position.
30. Choose **View** > **Show Graph** (or press Ctrl+g) for a visual representation of the Y-axis values. Close the expression editor and the PathCns property editor.

## Adding Particles

31. Select all the particle emitters on the planes: type `*emit*` in the Current Selection box in the Selection panel and press Enter. The selection box should display "MULTI (4)," indicating that four objects have been selected.
32. In viewport D (Explorer), choose the Selection filter (press **e**) and then click the **Lock** button to lock the selections in the explorer view.
33. In the Animate toolbar, choose **Simulate** > **System** > **Create Particle Cloud**.
34. In the SparksOp property editor's **Simulation** > **Execution State** drop box, select **Interactive**. This causes the simulation to automatically update as you play back the scene.

35. Set **Duration** to 200 and close the editor. This makes the sparks active for the duration of the animation.

## Editing in Multiple Property Editors

36. In the explorer, select the **Emission** node under the AIR_EMITTER_L node by clicking its name.
37. In the Selection panel's Current Selection text box, the selection name should appear as **AIR_EMITTER_L.EmissionProp**. Change the **_L** in the name to **_***, then press Enter. This selects all emitters at once.
38. Press Enter a second time to open a multiple-selection property editor.
39. In the Emission tab, enter `200` for the **Rate** value. This sets the amount of particles emitted per second. Enter `0.1` for the **InheritVel**, which assigns the particles one tenth the velocity of the plane.

## Editing the Particle Type

40. Click the **Edit** button next to **ParType** to edit the particle type.
41. Change the particle name to **smoke**.
42. Change the **Maximum Life** to 0.8 (the maximum lifespan of particles in seconds).
43. Change the Maximum Life **Variance** to 0.5 (maximum variation in the lifespan of particles in seconds).
44. Keyframe the **Mass** to 2 at frame 0, and 0 at frame 200 (mass of particles will slowly dissipate to nothing after 200 frames).
45. In the **Size** drop box, select **Age**. Each particle now scales according to its age.
46. Play back the animation.

## Conclusion

In this exercise, you have used basic path constraints, expressions, and particle animation. Usually, you will need to spend some time customizing these options to achieve a specific type of movement. Experiment with keyframing some of the path constraint options or try different expressions to see how the airplanes are affected, or change some of the particle settings to see how a minor change can drastically affect the look of the airplanes' smoke.

For more information, see the *Animating* guide.

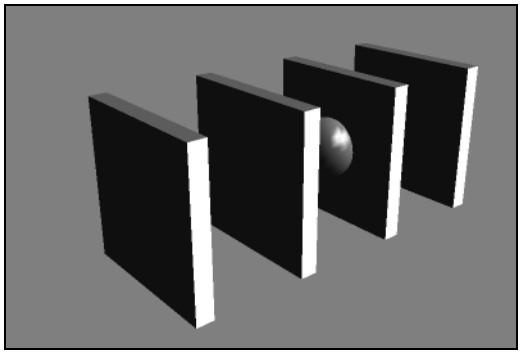
# Tutorial 7: Linked Parameters

Linking parameters is a fast way of creating animation. You use one parameter to drive another, without writing expressions. In this example, you will link the translation of cubes (representing doors) to a sphere's translation so that when the sphere gets close to a cube, the cube slides out of the way.

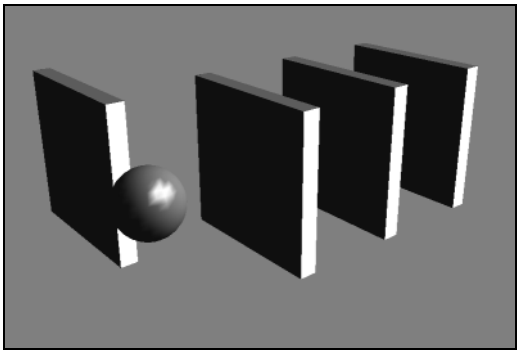
There are two basic steps to creating linked parameters. This tutorial shows you how to:

- Link one parameter to another.
- Set relative values to define the function curve that relates the values of the two parameters.

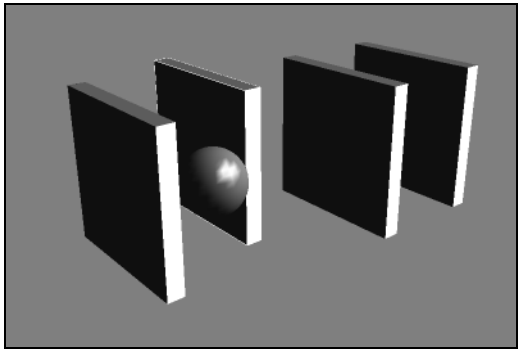
# Overview



1 **Animate a simple scene.**



2 **Link each cube's Z position to the sphere's X position.**

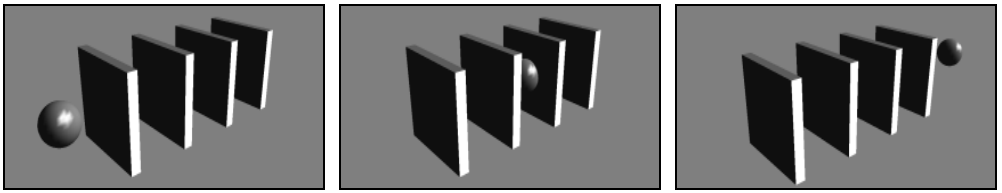


3 **Set relative values so that the cubes move out of the way automatically.**

## Linking Parameters

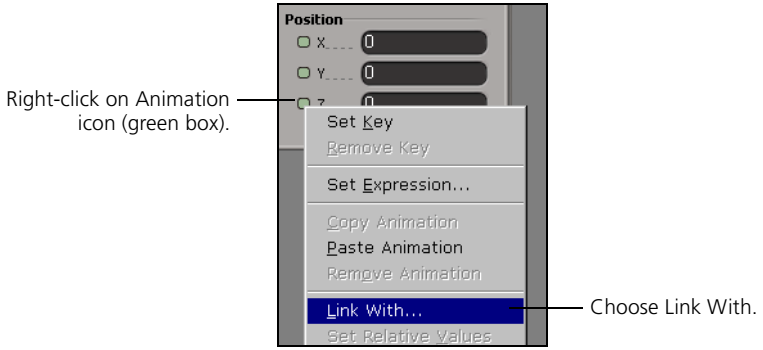
The first thing you will do in this tutorial is to link one parameter to another. This establishes the relationship that makes the value of the first parameter depend on the value of the second. To move the cube out of the way of the sphere in this example, you will link a cube's Z position to the sphere's X position.

1. Create a small scene as shown with some scaled cubes and a sphere arranged along the X axis. Animate the sphere so it runs through the cubes.



2. Once the animation is done, select the first cube hit by the sphere and press the **Property** button in the Selection panel on the main command area.
3. From the list, click the **Kinematics** > **Local Transform** icon.


4. Lock the Local Transformation property editor so it stays open.
5. In the Position controls, right-click the Animation icon (green box) beside the **Position Z** parameter. Choose **Link With** from the menu that appears.




6. In the pop-up explorer that opens, expand the nodes along the **Sphere/Kinematics/Local Transform/Pos** branch and pick the **X** parameter. Make sure you click the parameter's name **X** and not the icon.

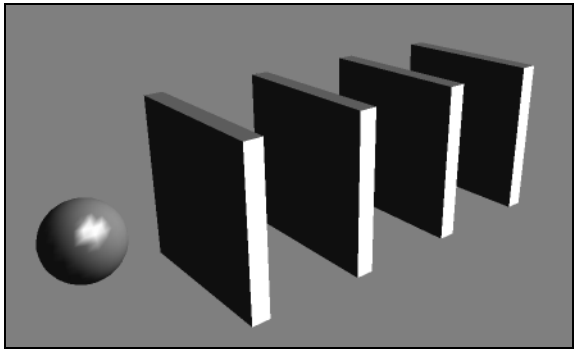
Icon shows that the Z position is linked

The two parameters are now linked but you haven't yet defined the relationship between their values. You will do this next.

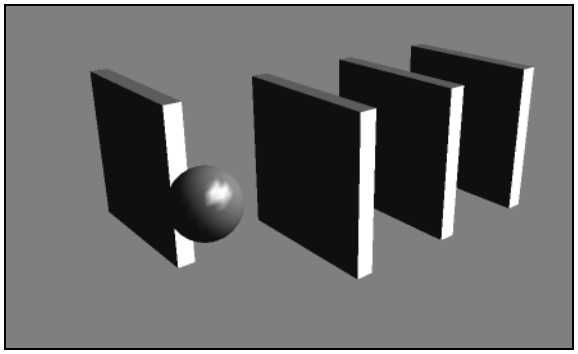
## Setting the Relative Values

When you set relative values, you specify the value of the driven parameter for a particular value of the driving parameter. Relative values are similar to keys, where you specify the value of a parameter at a particular frame. The difference between relative values and keys is that the values are a function of another parameter rather than a function of the current frame.

7. At frame 1, right-click the Animation icon beside the **Position Z** parameter in the cube's **Local Transformation** property editor, then choose **Set Relative Values**.
8. Find the sphere position at which you want the cube to start moving—the sphere is already animated, so move the frame slider until the sphere is just in front of the cube. Right-click the icon beside the **Position Z** parameter and choose **Set Relative Values** again. This relative value ensures that the cube stays in its current position until the sphere is almost about to move through it.

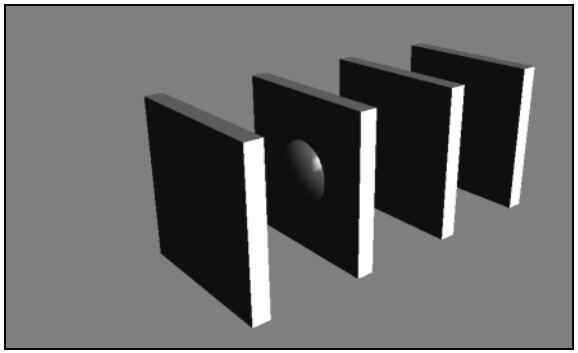


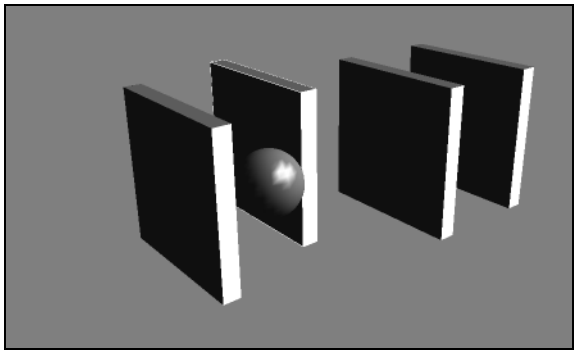
9. Go to a frame where the sphere is in the middle of the cube and translate the cube out of the way of the sphere by setting the **Position Z** value. When the cube is out of the way of the sphere's path, right-click the Animation icon beside **Position Z** and choose **Set Relative Values**.



10. Play back the animation: the cube adjusts its position according to the sphere's position. Move the sphere interactively; note that the cube's position depends on the sphere's position and not the current frame.

    You can continue setting relative values to move the cube back after the sphere has gone by, and repeat the procedure for the other cubes.





You may want to use the animation editor to change the interpolation of the relative function curve to linear. To do this, right-click on the **Position Z** icon in the cube's local transform property editor and select **Animation Editor**. In the animation editor, select the function curve and click on the **Linear Interpolation** button.

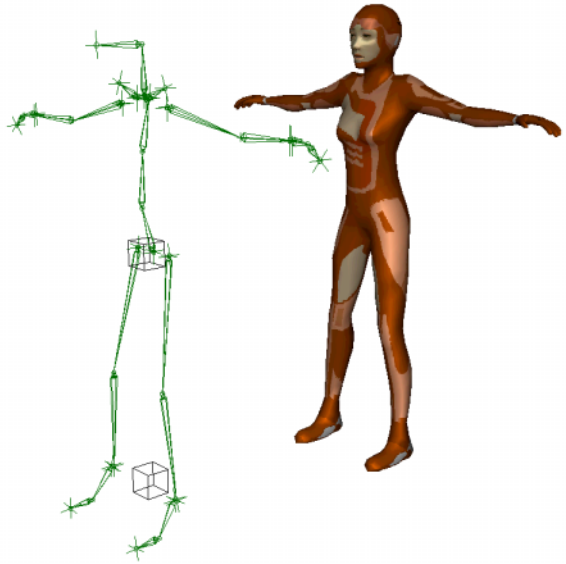
## Conclusion

Linked parameters are a quick way to achieve relationships that would take a while to figure out with expressions. They are especially useful with custom parameters, where you can create a custom control panel to control a rig with sliders.

For more information, see the *Animating* guide.

# Tutorial 8: Skeleton Construction

Skeletons provide an intuitive way to pose and animate your model. They are constructed with articulated chains and objects, which can be animated by keyframing translations, rotations, and scaling.

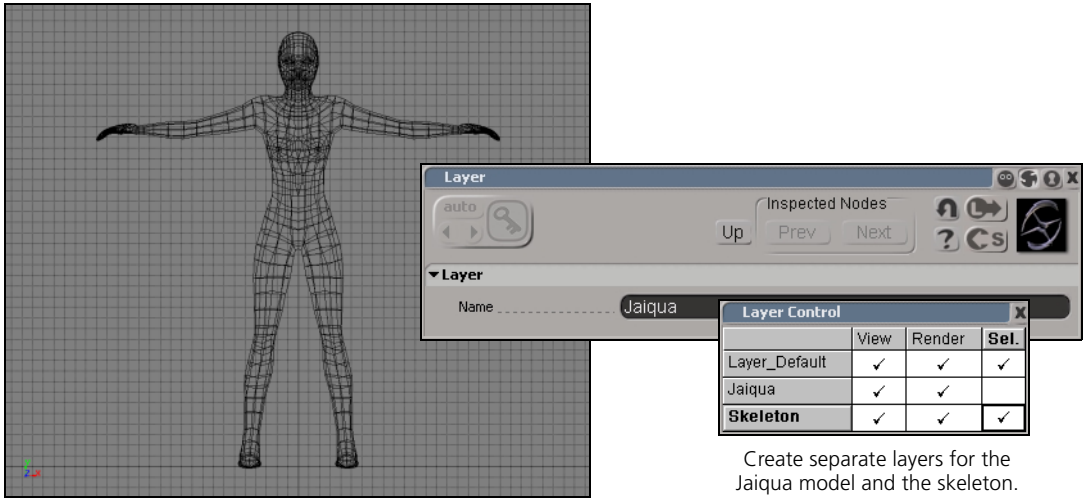


This tutorial shows you how to:

- Create articulated chains.
- Use chains and objects to build a skeleton for a particular model.
- Organize your skeleton's components into a hierarchy.

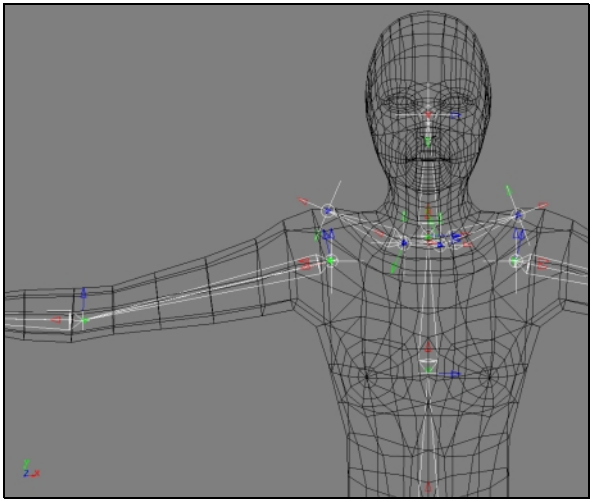
# Overview

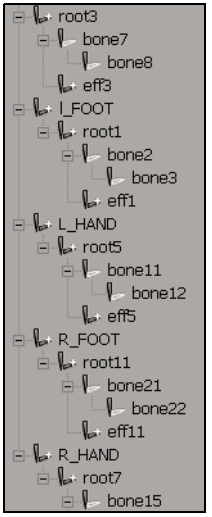
**1 Load the Jaiqua model.**




Create separate layers for the Jaiqua model and the skeleton.

**2 Build the skeleton.**

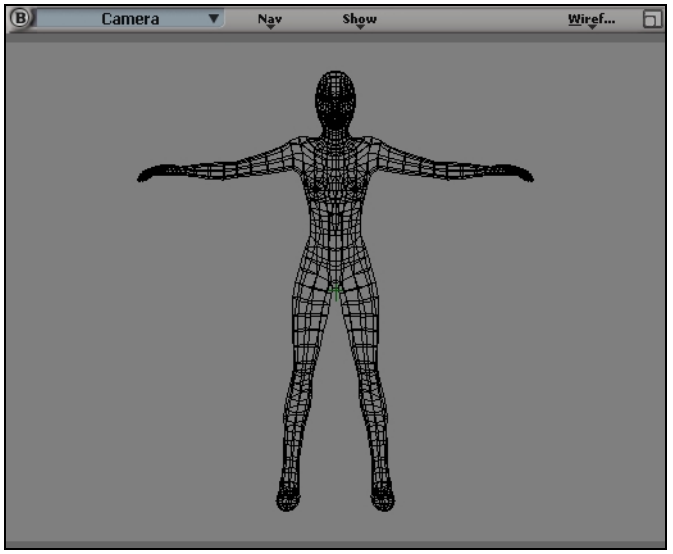





Arrange the skeleton elements into a hierarchy.

## Preparing to Build Your Skeleton

Biped skeletons can be created using an almost endless combination of articulated chains. You will be creating a fairly standard skeleton over the Jaiqua geometry.




It's important to use a reference while building a skeleton because they are usually created with a specific character in mind. Not only must the skeleton be constructed according to the geometry on which it will be used, but you also need a good idea of what this skeleton will be able to do. For example, if the character only bends his back, the skeleton would need only a few bones to achieve this simple motion.

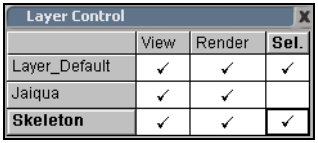
Reference scenes from the tutorial database
**<install directory>\content\tutorial_project\Scenes:**

Jaiqua polygonal character (no inverse kinematics): JAIQUA_POSE.scn
Jaiqua and inverse kinematics (IK) skeleton: JAIQUA_IK.scn

## Layers

Organizing the scene in layers will simplify building the chains in the next procedure. Layers allow you to load the character and keep it on its own layer, letting you quickly access and toggle its selectability and visibility.

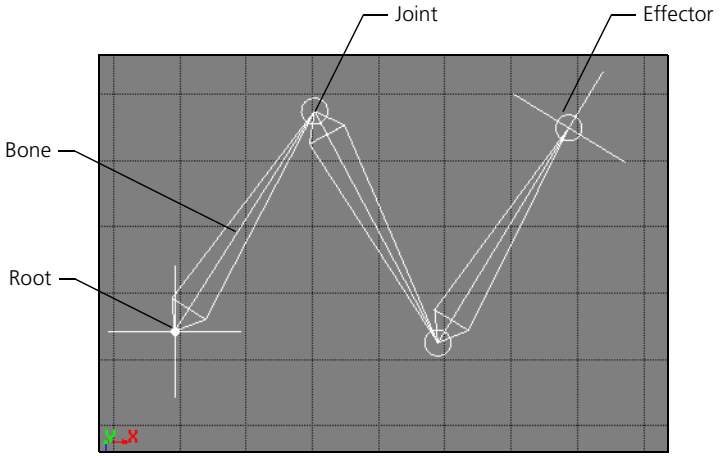
1. Load the JAIQUA_POSE scene from the tutorial database:
   **<install directory>\content\tutorial_project\Scenes**
2. Select the whole character by right-clicking on it.
3. Choose **Layers** > **New Layer** from the main-menu bar and enter **Jaiqua** as the layer name. This creates a separate layer for your model so that it can't be selected while you're creating the skeleton.
4. Deselect the character.
5. Choose **Layers** > **New Layer** from the main-menu bar and enter `Skeleton` as the layer name. This is the layer you will use to create the chains.
6. Choose **Layers** > **Layer Control** and deactivate the selectability (Sel.) of the Jaiqua layer.
7. Select the **Skeleton** layer from the drop-down list under **Layers** menu.



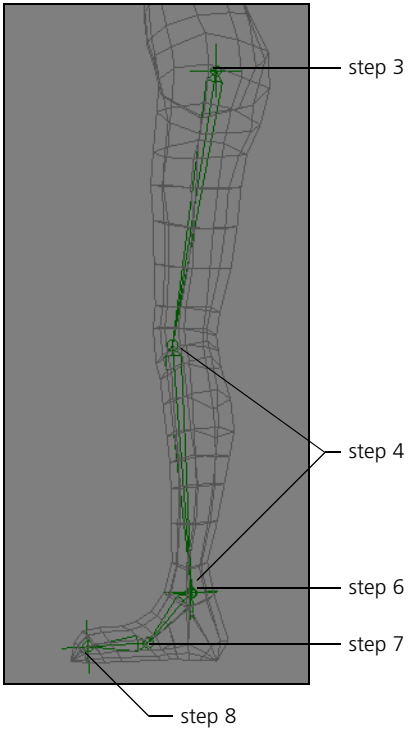

## Drawing the Chains

Draw your chain in the viewport that matches the plane defined by the root and effector of the 2D chain as closely as possible. For example, if you are drawing a chain that will rotate in the global Z axis, draw it in the Front viewport. The preferred angles are set using these original angles, although you can still edit them at any time. Working from a single viewport when drawing a chain gives the most predictable results. Once you draw the chain, close the chain creation tool and tree-select the new chain to position it according to the reference pose.

Chains drawn in the Top, Front, or Right viewports are always created with respect to the center of the world (origin). Although all the chains look fine when they are initially created, some will have to be repositioned after creation in order to fit properly in the Jaiqua model.



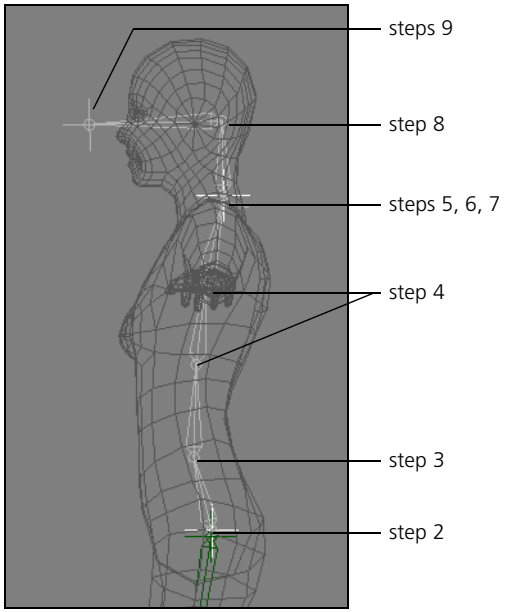

Example of a three-bone chain with its components.




### *Create the leg and foot chains*

When you draw a chain, the first mouse click creates the root and the last click creates the effector if followed by the end chain command (middle-click) or the end chain and exit command (right-click). Middle-clicking allows you to start a new chain immediately without needing to choose the **Create** > **Skeleton** > **Draw 2D Chain** or **Draw 3D Chain** command again.

1. Switch viewport D to the Right view, then frame Jaiqua's legs. If desired, press **g** to toggle the grid on or off.
2. Choose **Create** > **Skeleton** > **2D Chain** from the Animate toolbar.
3. Click to position the first bone's root over the leg as shown in the illustration at left.
4. Click over the knee to create the first bone, then click over the heel to create the second bone.
5. Middle-click to create the effector and end drawing the leg chain.
6. Click over the leg effector to create the foot root.
7. Click to make the foot bone
8. Click to make the toe bone.
9. Middle-click to create the effector and end drawing the foot chain.




### *Create the spine and head chains*

Create a four-bone spine chain, mounted with a two-bone head chain.

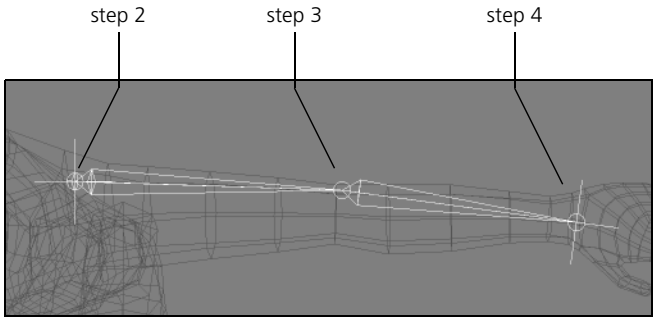
1. Frame Jaiqua's torso in the Right viewport.
2. Click over the root position of the backbone.

If you have exited the chain creation tool, choose **Create** > **Skeleton** > **Draw 2D Chain** from the Animate toolbar to reactivate it.

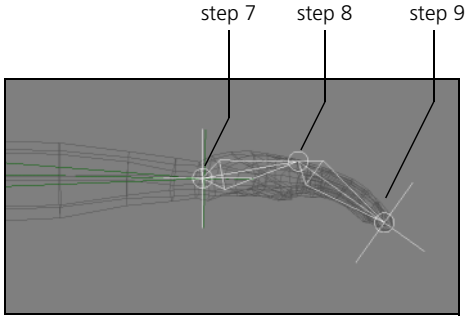
3. Click at the next bone position as shown in the illustration at left.
4. Click at the next two bone positions.
5. Click over the neck.
6. Middle-click to create the effector and end drawing the spine chain.
7. Click over the spine effector to start creating the head chain.
8. Click over the ear position.
9. Click in front of the face to make the last chain. This bone allows you to tilt Jaiqua's head.
10. Middle-click to create the effector and end drawing the head chain.

### *Create the arm and hand chains*

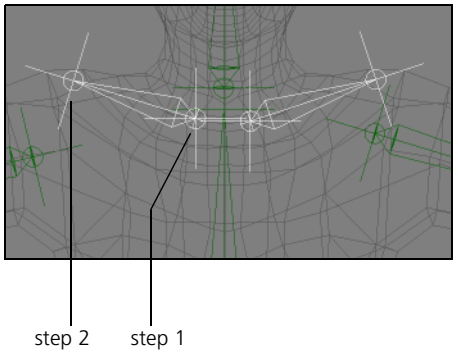
1. Frame one of Jaiqua's arms in the Top viewport.
2. Click over the left shoulder to create the root of the first bone, as show in the following illustration.
3. Click over the elbow to create the first bone.
4. Click over the wrist.




5. Right-click to end drawing the first bone.
6. Press Ctrl+d to duplicate the arm chain.
7. In the **Z** rotation box in the Transform panel of the main command area, enter 180 to rotate the duplicate chain 180° about the global Z axis.
8. Translate the duplicate over Jaiqua's other arm, then select both arm chains and translate them vertically in Y until the roots are positioned at Jaiqua's shoulders.
9. Press **c** to activate rotation mode, and make sure that **Local** transformation mode is on.
10. Select an upper arm bone and rotate it in Y until it is inside Jaiqua's geometry. Repeat for the other upper arm bone, then repeat for the lower arm bones.




11. Frame Jaiqua's hand in the Front viewport.
12. Middle-click on **Create** > **Skeleton** and click over the wrist to start the hand chain, as in the illustration at left.
13. Click over the knuckles to create the first bone.
14. Click over the fingertip position to create the last bone.
15. Middle-click to create the effector and terminate the chain.
16. In the Top view, translate the hand into place.
17. You may need to modify the bone's length: Select the bone that needs to be adjusted and press Enter to open the Chain Bone property page. Change the **Length** if needed.
18. Repeat the previous steps to create the other hand chain.



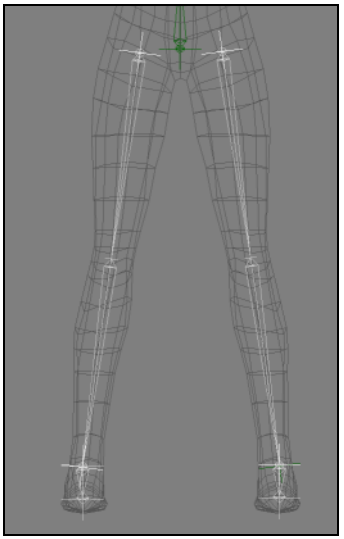

### *Create the shoulder-bone chains*

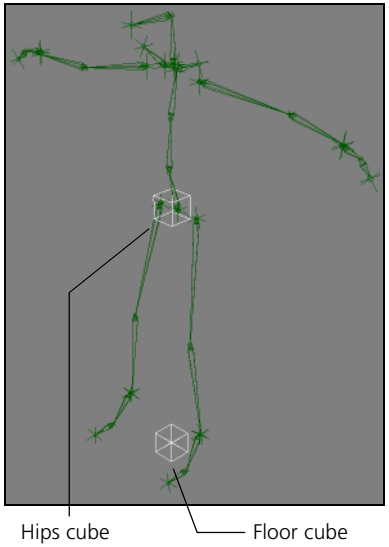
The shoulder-bone chains are single-bone chains.

1. In the Front viewport, click to create the root of the first bone as in the illustration at left.
2. Click at the shoulder to create the bone.
3. Middle-click to create the effector and end drawing the shoulder chain.
4. Create the other shoulder chain by repeating the previous steps.
5. Right-click to create the effector and exit the chain creation tool.
6. In the Right viewport, Shift+select both shoulder chains and translate them into position.

### *Duplicate and position the leg chains*

1. Right-click a bone in the leg chain to select the entire chain.
2. In the Front viewport, translate (press **v**) and rotate (press **c**) the leg chain to the right (over Jaiqua's right leg), as shown in the illustration below.
3. Press Ctrl+d to duplicate the chain.
4. Translate and rotate the new duplicated chain onto the other leg position.
5. Repeat the steps for the feet.






## Hierarchy Relations

Control objects are used as references and will also be used to provide a hierarchy for the skeleton bones and effectors. Changing the hierarchy will allow local keyframing of bones according to these control objects.

Chain roots and nulls would work just as well, but implicit cubes clearly show rotation from any angle and are not rendered.

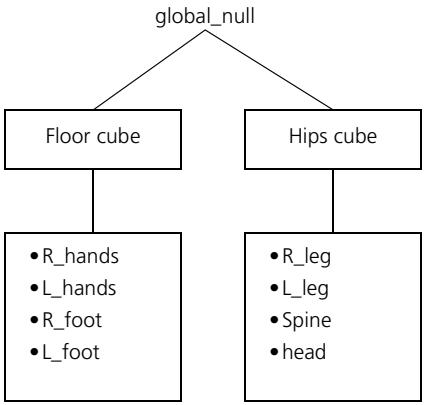
### *Create the hips and floor control objects*

1. Choose **Layers** > **Layer Control** from the main-menu bar and deselect the **View** of the Jaiqua layer. This hides Jaiqua's geometry so it's easier to work with the skeleton and control objects.
2. Close the Layer Control window.
3. Choose **Get** > **Primitive** > **Implicit** > **Cube** from any toolbar. Enter **2** for the implicit cube's **Length** and close the property editor.
4. Press Ctrl+d to duplicate the cube.
5. Position one of the cubes between the feet and slightly above the heels. This is the floor cube.
6. Position the other cube between the backbone and legs. This is the hips cube.

## Building the Hierarchy

Building a hierarchy by defining the parent/child relationship between each object in the skeleton will help define the flexibility of the character so that it can be animated properly.

The following illustration shows the basic relations that you are going to build in the next procedure. The last step will be to make the hips cube and floor cube children of a global null. The effectors will be cut from their respective chains' hierarchies and placed right under the floor cube. The ability to remove an effector from its chain hierarchy is a new feature in SOFTIMAGE|XSI.




*Parent the legs and backbone under the hips cube*

You can build the hierarchy by using the **Parent** button in the Selection panel in the main command area or via the explorer by dragging elements into place.

1. Select the hips cube, click the **Parent** button, and pick one of the leg's roots to make it a child of the cube. You may need the zoom for this.
2. Pick the other leg's root.
3. Pick the backbone's root.
4. Right-click to exit parent mode.

Be careful when using the Parent mode as you need to right-click to turn it off. For example, you cannot temporarily select an object while in Parent mode.

*Parent the shoulders and head under the spine effector*

5. Select the spine effector, click the **Parent** button, and pick a shoulder's root.
6. Pick the other shoulder's root.
7. Pick the head's root.
8. Right-click to exit parent mode.

*Parent the arms under the shoulders*

9. Select a shoulder's effector, click the **Parent** button, and pick the corresponding arm's root.
10. Right-click to exit parent mode.
11. Select the other shoulder's effector, click the **Parent** button, and pick the corresponding arm's root.
12. Right-click to exit parent mode.

*Parent the hands under the arms*

13. Select an arm's effector, click the **Parent** button, and pick the corresponding hand's root.
14. Right-click to exit parent mode.
15. Select the other arm's effector, click the **Parent** button, and pick the corresponding hand's root.
16. Right-click to exit parent mode.

*Parent the feet under the legs*

17. Select a leg's effector, click the **Parent** button, and pick the corresponding foot's root.
18. Right-click to exit parent mode.
19. Select the other leg's effector, click the **Parent** button, and pick the corresponding foot's root.
20. Right-click to exit parent mode.

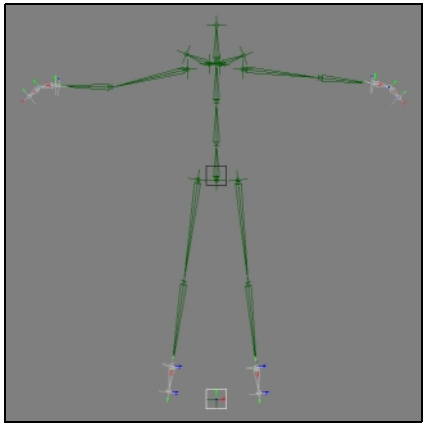
*Parent the leg and arm effectors under the floor cube*

21. Branch-select (middle-click) an arm's effector (the hand should become selected as well).
22. Click the **Parent** button and middle-click the floor cube to make it the parent of the hand. Right-click to exit parent mode.
23. Repeat the previous steps for the other hand and for the two feet as well.

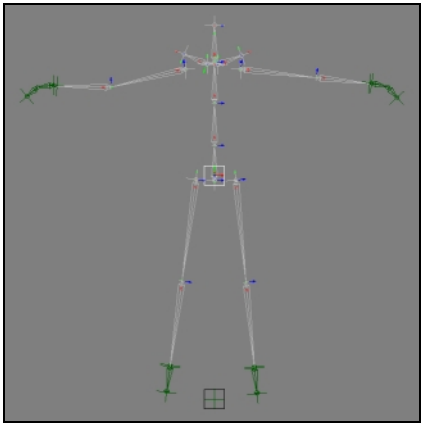
Once you have completed these first parenting steps, you should be able to branch-select the hips cube and the floor cube to select their children as shown in the following illustrations.

*Parent the two cubes under a single null*

24. With nothing selected, click the **Parent** button, then click the two cubes. Right-click to exit parent mode.



Floor cube selected in branch mode



Hips cube selected in branch mode

## Conclusion

You have now completed a simple but typical biped skeleton setup. Save this scene as you will use it later on.

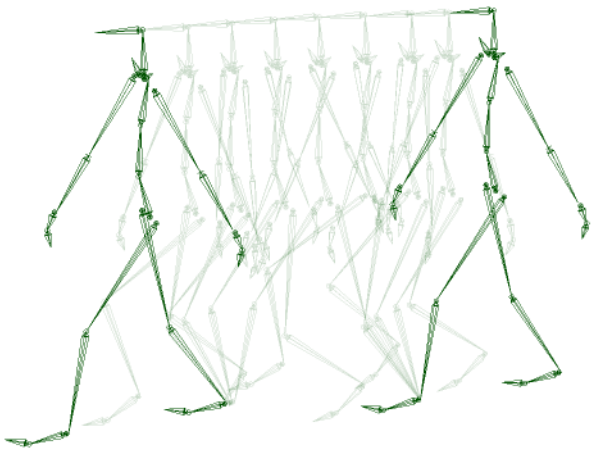
To fully understand the hierarchy you have built, you will need to animate different parts of the skeleton. Try the next tutorial to see how a well-constructed skeleton can fit into the animation process.

For now, if you select the hips cube (in branch mode) and move it around, the feet and hands will move with it. You will need to keyframe these objects (hands, feet, head) in order to really see the local relation between the floor cube, the hips cube, and the chain elements.

For more information, see the *Animating* guide.

# Tutorial 9: A Walk Cycle

A walk cycle is probably the most common character-animation task. It is a segment of a walking sequence that can be repeated as the character travels from point to point. To be cycled properly, all the keyframes defining two steps of a walk are needed (right foot leaves the floor, goes forward, touches the floor; left foot leaves the floor, goes forward, touches the floor).



This tutorial explains the basic principles of a walk cycle. Here you will learn the basic concept of keyframing and marking in SOFTIMAGE|XSI frame by frame, step by step.

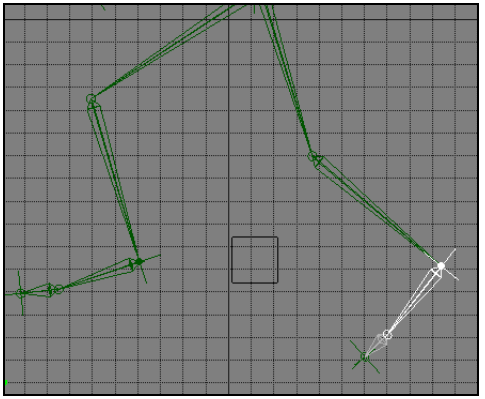
The walk cycle you will be working on is kept to the minimum number of keys needed to create the sequence. These keys will be created on three principal frames (frames 1, 20, and 40).

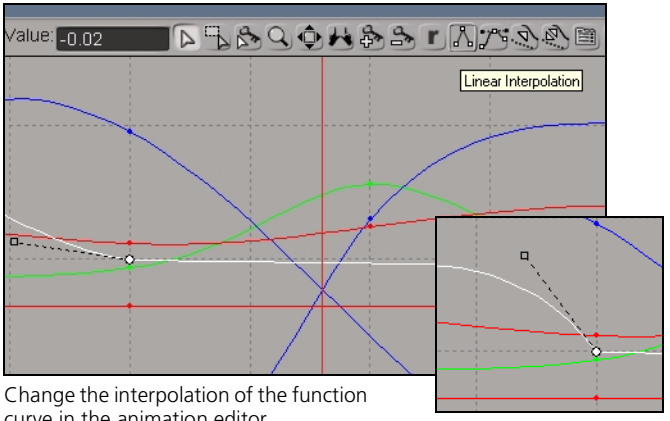
This tutorial shows you how to:

- Animate the translation of the skeleton's feet and hands.
- Give the character's feet a feeling of weight by editing the feet's function curves in the animation editor.
- Rotate the skeleton's hips for added realism.

# Overview

**1 Translate the feet.**

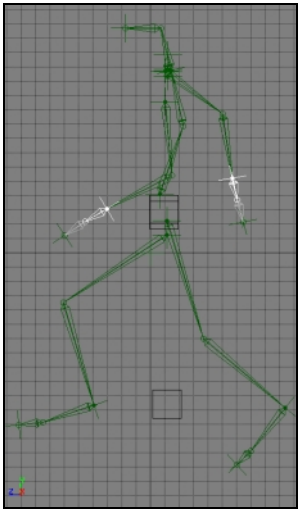




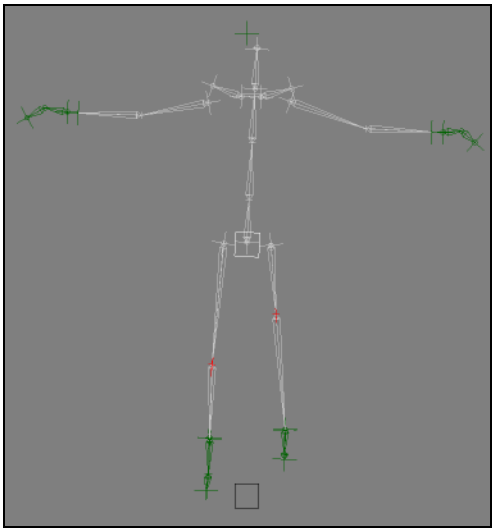

Change the interpolation of the function curve in the animation editor.

Edit the function curve slope.

**2 Translate the hands.**



Rotate the hips for more natural movement.



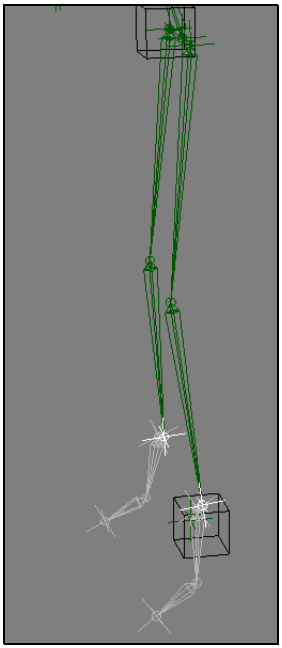
## Starting the Sequence

The first frame sets the start of the sequence, and the last frame is an exact copy of the first in order to have a perfect loop cycle. The middle frame is where a foot's previous motion is repeated on the other leg; that is, the first half is where the first foot goes forward and the second half is where the other foot goes forward).

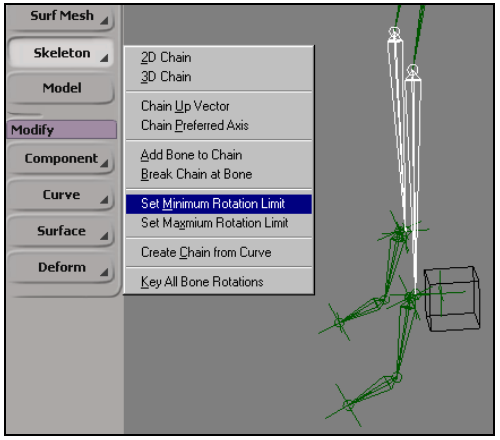
The animation of a character's basic stride (two steps) is created and then repeated as necessary to get the character from one place to another.

## Rotation Limits

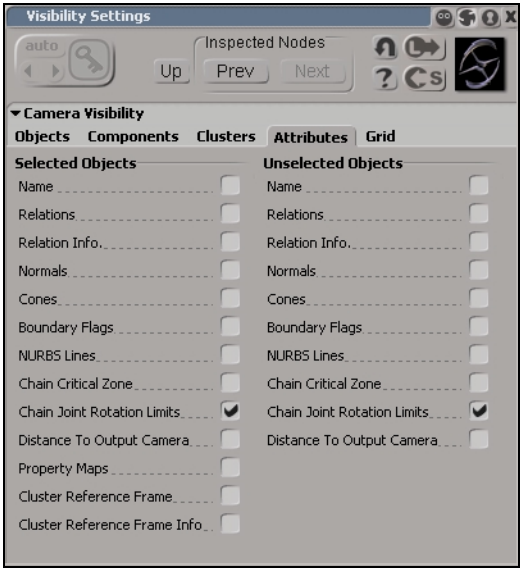
Setting rotation limits ensures that chain bending does not exceed the logical angles that you expect from a typical biped. You will set these rotation limits on Jaiqua's legs.

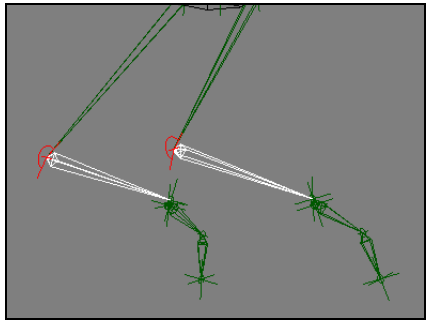


1. Use the skeleton created in the previous tutorial or load the **JAIQUA_IK.scn** scene from the tutorial database: **<install directory>\content\tutorial_project\Scenes**. If desired, turn off the visibility of the layer with Jaiqua's geometry using **Layers** > **Layer Control**.
2. Select both feet by middle-clicking a leg's effector, then holding down the Shift key while middle-clicking the other leg's effector.
3. Translate the effector hierarchies vertically until the knees are almost straight as in the illustration on the left.
4. Select the second bone of each leg and choose **Create** > **Skeleton** > **Set Minimum Rotation Limit** from either the Model or Animate toolbar.



5. To make the rotation limits visible, choose **Show** > **Visibility Options** from the viewport's title bar. In the property editor, click the **Attributes** tab and select **Chain Joint Rotation Limits**.



6. Select both feet again and position them so that the knees are bent as in the illustration at left.
7. Select the bones again and choose **Create** > **Skeleton** > **Set Maximum Rotation Limit**.

You can also change these limits by selecting a bone and pressing Enter to open its property editor. The rotation limits can be found on the Kinematic Joint > Rotation Limits page.

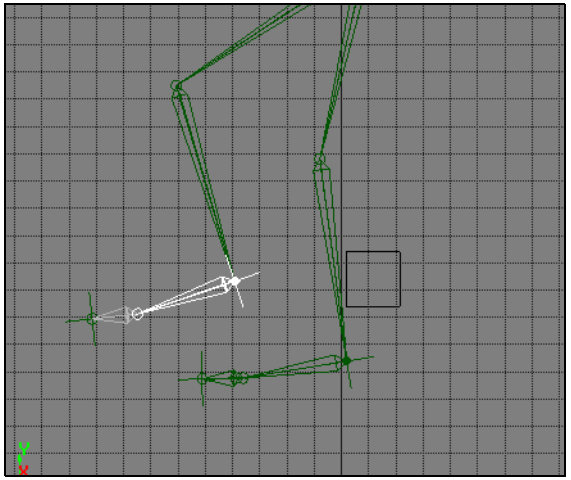
## Animating Foot Translation

You will set all keyframes using the following basic procedure:

1. Go to a specific frame.
2. Position the object to be keyed (for example, a hand or foot).
3. Make sure that local translations are marked.
4. Set the key by clicking the keyframe icon (the key) in the Animation panel at the bottom of the screen or by pressing **k**.

*Set keys on frame 1*

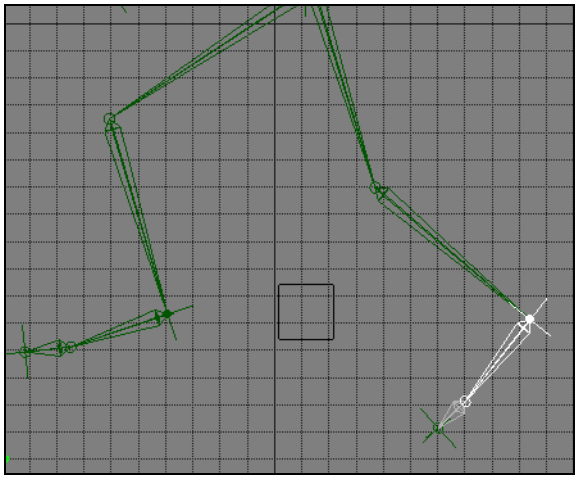
1. Go to frame 1.
2. Branch-select a leg effector to select it and its child foot chain.
3. In the Right viewport, press **v** and move the foot up and forward as in the following illustration.



Position of the first foot at frame 1.

4. Make sure that the Marked Parameter box in the bottom right corner displays **kine.local.pos** (local translation) and press **k** to keyframe the local position of this foot at frame 1. You can also click the keyframe icon in the Animation panel.
5. Select the other foot in branch mode.

6. In the Right viewport, translate the foot as shown in the following illustration.

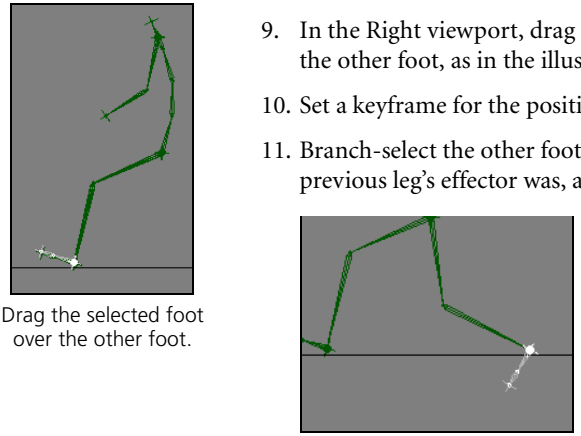


The final position of both feet at frame 1.

7. Keyframe the position of this foot at frame 1.

### *Set keys on frame 20*

8. Go to frame 20. Now you will reverse the position of the feet.
9. In the Right viewport, drag the selected foot to approximately the same parallel position as the other foot, as in the illustration on the left.
10. Set a keyframe for the position of this foot (frame 20).
11. Branch-select the other foot. In the Right viewport, position it approximately where the previous leg's effector was, as in the following illustration.



Drag the selected foot over the other foot.



12. Keyframe the foot's position.

### *Set keys on frame 40*

In these next steps, you will create a full cycle by copying the animation keys from frame 1 to frame 40.

13. Go to frame 1.
14. With one foot branch selected, right-click on frame 40 in the timeline. A green line appears in the timeline.

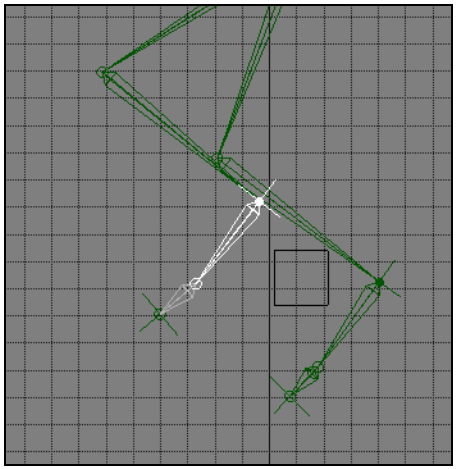
Right-clicking and dragging the playback cursor changes frames without changing the values of animated objects. This allows you to copy keys from different frames.

15. Keyframe the position from frame 1 at frame 40.
16. Branch-select the other foot.
17. Go to frame 1.
18. Right-click on frame 40.
19. Keyframe the position from frame 1 at frame 40.
20. In the End Frame text box on the timeline, set the last frame of the animation to 39 and play back the animation of Jaiqua walking. By playing to frame 39, you avoid repeating the identical frames at 1 and 40.

### *Add keys to the walk cycle*

Since you have keyed the translation on mainly one axis, the character seems to shuffle her feet. To remedy this, you will first fix the foot that translates forward at frame 10, by making it translate up before it moves toward the floor.

1. Select one of the feet by middle-clicking the leg effector. Play the animation and make sure you have selected the foot that moves forward at frame 10. If you have the wrong foot selected, select the other one. You can double-check that you have the correct foot by scrubbing the timeline around frame 10.
2. Go to frame 10 and move the foot up and forward as in the following illustration.



The foot going forward is translated up and keyed at frame 10.

3. Keyframe the foot's local position.
4. Select the other foot.
5. Go to frame 30 and move the foot up and forward as you did in the previous steps.
6. Keyframe the foot's local position.

## Editing the Animation Function Curve

You can easily add some weight to the walk by changing the interpolation shape of the translation function curves.

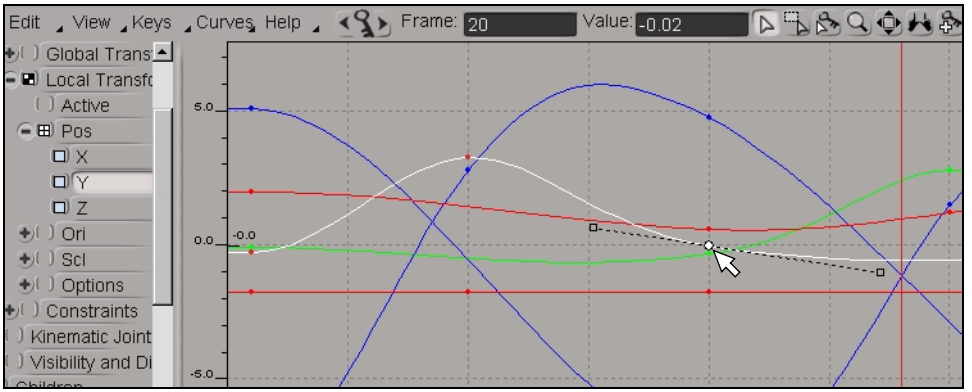
7. Branch-select both feet using the Shift key to make multiple selections. Make sure that you include the leg's effectors in the branch selections.
8. Choose **Animation** > **Animation Editor** from the Animation panel at the bottom of the window. The animation editor displays all animated parameters by default.

You can also open the animation editor by pressing **0** (zero) on the top of the keyboard (not the numeric keypad).

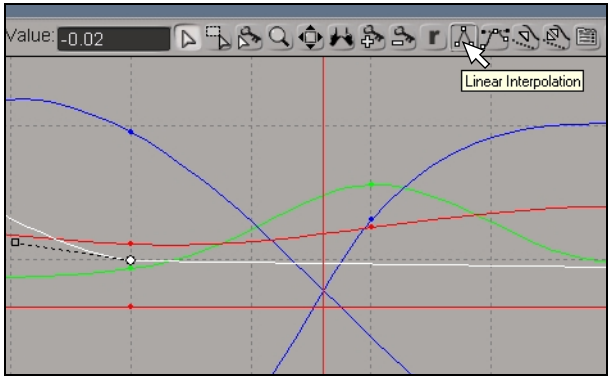
*Set the interpolation type of the feet's Y translation*

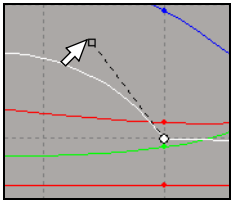
9. In the animation editor, select the Y translation of one of the feet. You can find the Y translation in the property tree pane on the left, under Kinematics > Local Translation > Pos > Y.

10. Select the key where the foot touches the floor (frame 20) by clicking on the key point.




11. Click the Linear Interpolation button in the toolbar to make the interpolation linear after that key. This causes the foot to move at a constant speed between this keyframe and the next.

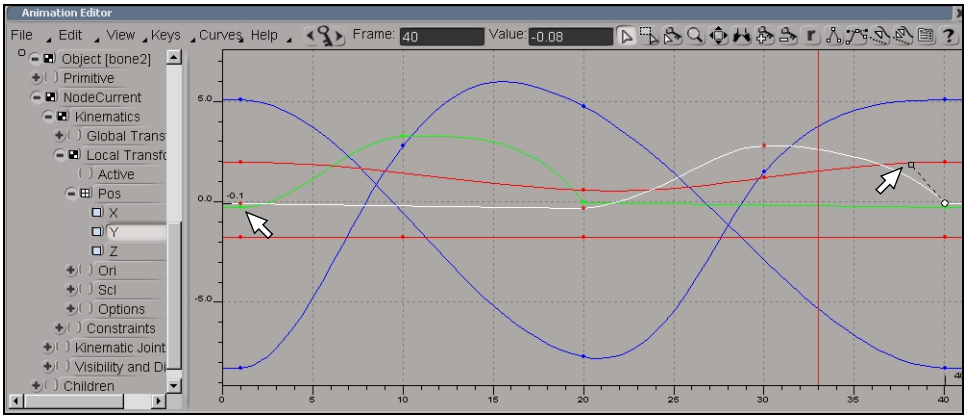


Drag the slope handle to change the curve's shape. The longer the handle, the sharper the curve.

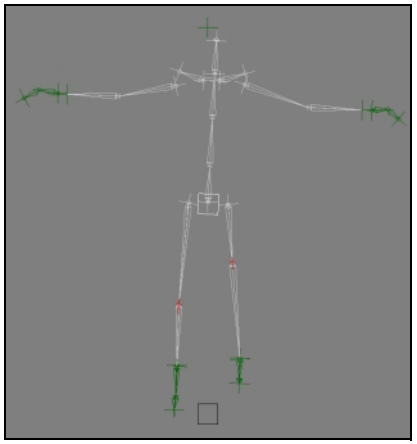
12. Drag the key's tangent handle to get the curve shape as shown in the illustration on the left. This shape makes the foot translate faster when it gets close to the floor position and stop more abruptly.

13. Select the other foot's Y translation.
14. Select the first keyframe (at frame 1) and click the Linear Interpolation button.




15. Adjust the handle of the last keyframe (frame 40) to set the interpolation as you did for the other foot.
16. Close the animation editor and play back the animation.

### *Add hip rotation*



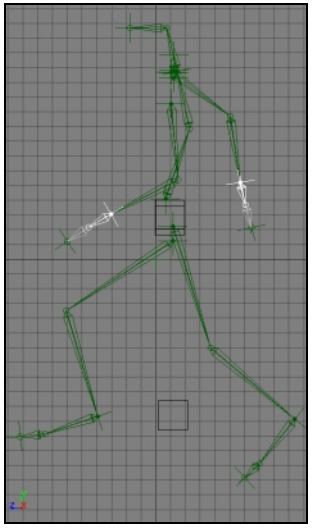
Adding hip rotation will make the walk look more natural and help get rid of the static feel of the animation so far.

17. Go to frame 1.
18. Branch-select the hips cube.
19. Press **c** to enter rotation mode. Make sure you are in local rotation mode.
20. Set the cube's rotation to -3 on the Z axis, as shown in the illustration on the left, and save a key.
21. Set the end frame of the animation back to 40.
22. Go to frame 40 and set another keyframe with the same value. You don't need to right-click on the timeline because there are no other keys set yet.
23. Go to frame 20, set the rotation to 3 on the Z axis, and set another keyframe.

### *Animate the arms*

Animate the arms so that they move in the opposite direction of the feet.

24. Select an upper arm. Rotate the arm locally about the Y axis until it is nearly parallel with the body.
25. Repeat for the other arm.
26. Select one arm effector in branch mode.
27. Go to frame 1.
28. Position the hand opposite to the foot on the same side. For example, if the left foot is in front of the body, the left hand should be behind the body.
29. Make sure that local position parameters are marked and set a keyframe.
30. Repeat for the other hand.

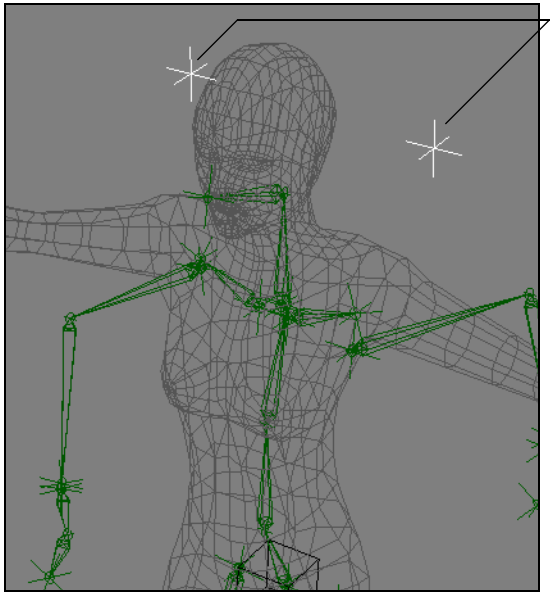


31. Copy both keys to frame 40: select both hands, go to frame 40, and set a keyframe.
32. Set the last frame of the animation to 39.
33. Go to frame 20.
34. Reverse the hand positions one at a time and set keyframes for each.

## Creating an Up-vector Constraint

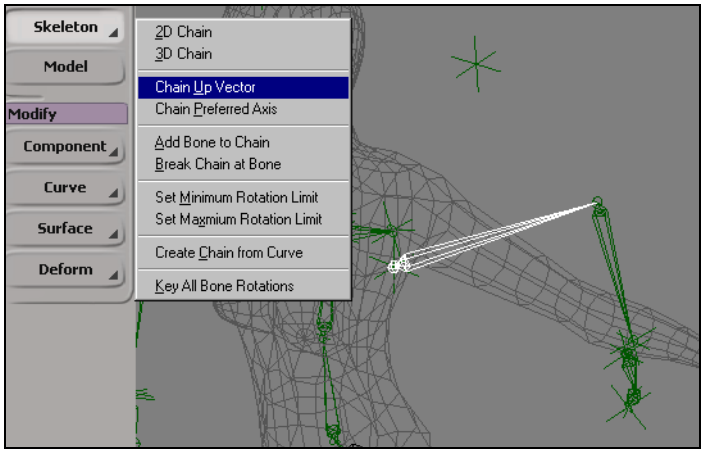
Jaiqua's elbows are probably moving in the wrong direction. You can use up-vector constraints to control the direction of the bones.

1. Choose **Get** > **Primitive** > **Null** from the Animate toolbar to get a null.
2. Position it above and behind Jaiqua, to one side.
3. Duplicate it and position the new null over the other side as in the following illustration.




You don't need to position the null precisely. Once the up vector is attached to the bones, you can readjust the null's position.

4. Select one of the upper bones as in the following illustration, then choose **Create** > **Skeleton** > **Chain Up Vector** and pick the null.




5. Adjust the position of the null in space while playing the animation so that the elbow point moves in a logical direction.

6. Save the scene.

## Conclusion

The walk cycle you have created is quite simple. The goal here was to recreate a basic animation concept using inverse kinematics (IK). When striving for realism, a walk can easily get very complex, with motion-captured walks providing the most realistic animation. Skeletons can be created in a variety of ways for these capture setups.

Very often, animators end up creating walk cycles such as these, then spend time adding more keys to define the motion and model's character.

For more information, see the *Animating* guide.

# Tutorial 10: The Animation Mixer

The walk cycle that you created in a previous tutorial can be animated and mixed with other animations.

The animation mixer works with the same logic that you would find in a non-linear editing system. The animation mixer is a tool that gives you high-level control over animation, by letting you position, cycle, scale, warp, bounce, and mix actions.

Actions are animation segments that you define once and apply as many times as you like. You can create an entire library of actions, like walk cycles or poses, and copy them from one model to another. You can apply them one after the other in any sequence you like so that you can quickly get your character up and running (and jumping, and kicking...). You can also mix actions together, or create compound actions that contain other actions.

In Part I of this tutorial you will:

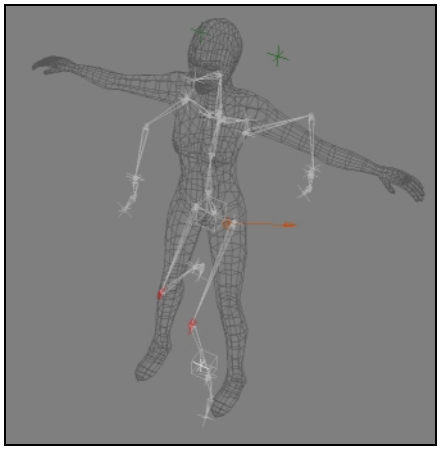
- Create action sources.
- Instantiate action clips.
- Cycle and crop action clips.
- Modify the underlying function curves of the original action source.

In Part II of this tutorial you will:

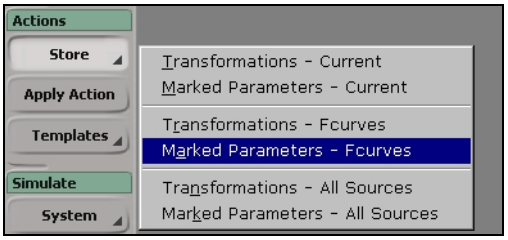
- Import and clean motion-capture function curves.
- Create actions for various kinds of animation.
- Mix between animated actions and static poses.
- Create markers on animation tracks.

# Overview - Part I

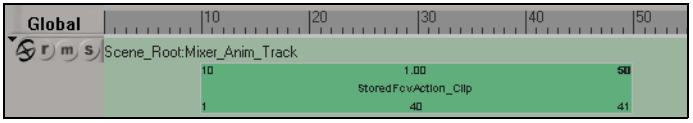
**1** **Start with an animated scene.**



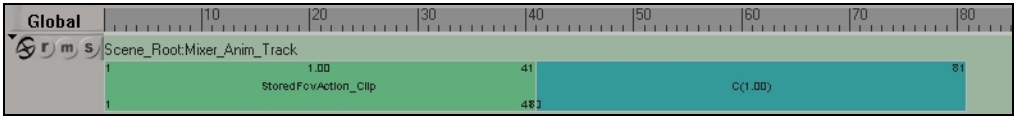
**2** **Store an action source.**



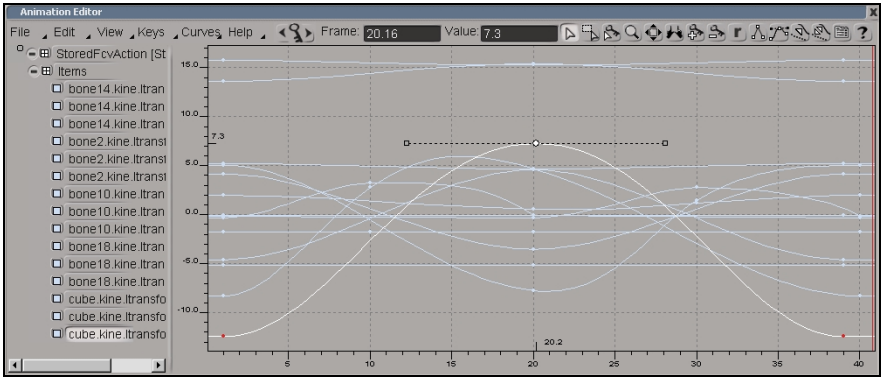

**3** **Instantiate an action clip.**




**4** **Cycle and crop the clip.**



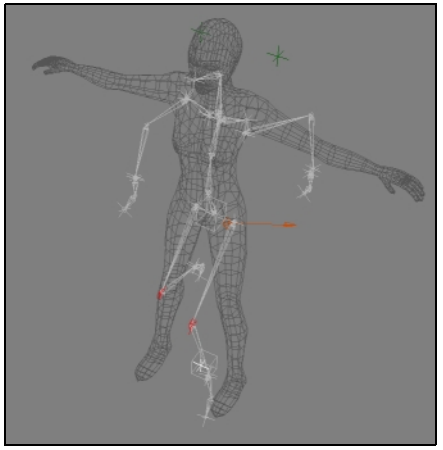

**5** **Modifying the source's function curves.**

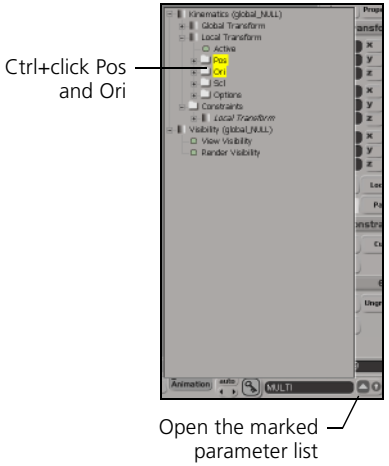


## Storing Actions

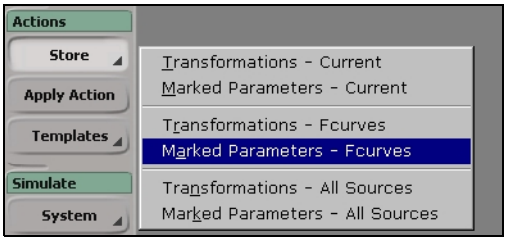
Once you have any kind of animation, you can store this animation in an action source. When storing actions, you can choose to store all transformations or just the marked parameters, thus creating a smaller action file. In this example, you will mark the position and orientation parameters before storing them in an action.

1. Start with the completed walk cycle from the previous tutorial, or open the **JAIQUA_IK_DONE** scene from the tutorial database: **<install directory>\content\tutorial_project\Scenes**
2. Right-click on part of the skeleton to select the entire skeleton hierarchy (not the model) of the walking character.





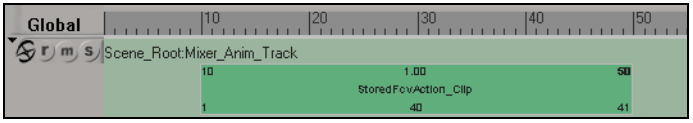

3. Open the marked parameter list in the Animation panel by clicking the triangle button on the bottom right. Expand the Kinematics/Local Transform nodes, then Ctrl+click to mark both **Pos** and **Ori** (position and orientation).
4. From the Animate toolbar, choose **Actions** > **Store** > **Marked Parameters - Fcurves**. Click OK to accept the default values. The animation is stored and is no longer linked to the object.



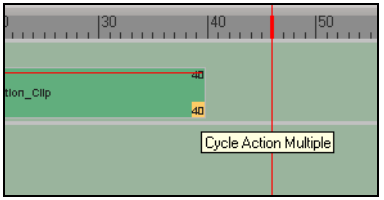

If you need to restore the original animation after creating an action, select the action from the Mixer\Sources\Animation branch of the explorer and then choose **Actions** > **Apply Action** on the Animate toolbar.

## Instantiating Clips

5. Restore your workspace to show all four viewports if necessary, and change viewport C to the animation mixer view.
6. With the skeleton character still selected, press **Update** on the animation mixer's command bar.
7. Right-click on a track and choose **Load Source** > **StoredFcvAction**. The action is loaded as a clip on the track.



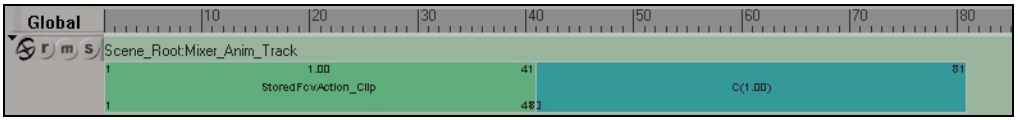

8. Play back the resulting animation.
9. Click and drag the action clip so that it starts at frame 1.
10. Change the animation mixer length by setting the End frame in the animation mixer to 100.
11. Change the main timeline End frame to 100 as well.


## Adding a Cycle to the Action Clip

12. Click on the lower-right corner of the action clip so that an orange square appears. Drag the clip so that it ends just before frame 90.

    The animation did not cycle all the way to frame 90 because dragging creates only cycles that are integral multiples of complete actions. You can create fractions of cycles by setting it in the Time Control property editor of the action clip, which you will display in the next step.



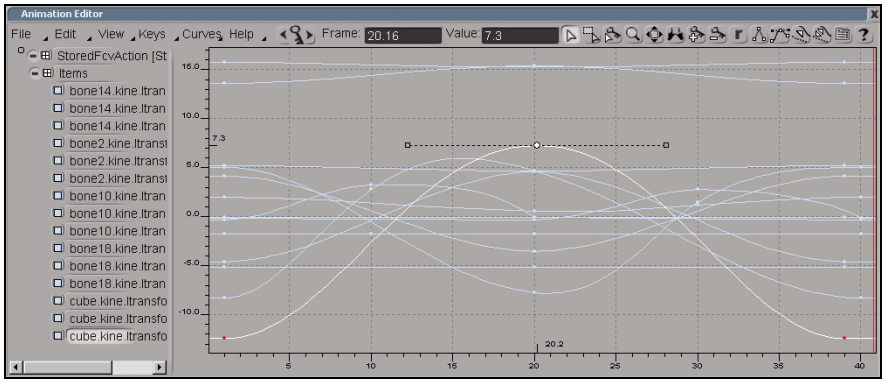

## Cropping the Action Clip

When playing the animation, you can see at the transition frame between the two cycles that you have forgotten to shorten the original animation back to 39 frames. Since frame 40 is a duplicate of frame 1 you have two identical frames beside each other.

13. Right-click on the action clip and choose **Time Control**.
14. Enter `40` in the **Out** value of Source Clipping.
15. Close the property editor. The action now has a cycle length of 39 frames.

## Editing the Original Function Curves

16. Is the rotation of the hip is a bit too much? Or not enough? You can modify this as needed. Right-click on the clip again and choose **Source**.
17. In the Source property editor, click the **Edit Source Data** button to open the animation editor.
18. Select hips_null.kine.ltransfo.rotz from the function curve list.
19. Go to frame 20 by dragging the cursor in the timeline.
20. Select the key on this frame of the selected function curve.




21. Move the key down to decrease the swing, or up to increase it.

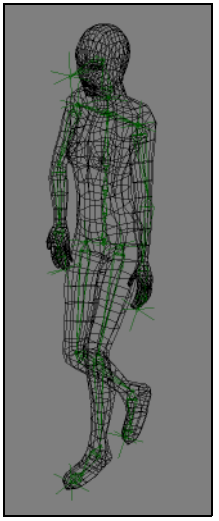
    Look at the character to see it update the new values as you modify them.

22. Play the animation and find the correct hip swing adjustment.

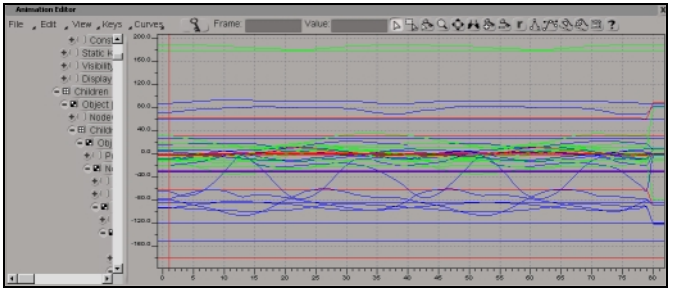
23. Close the animation editor and property editor when done. Play back the final animation.

# Overview - Part II

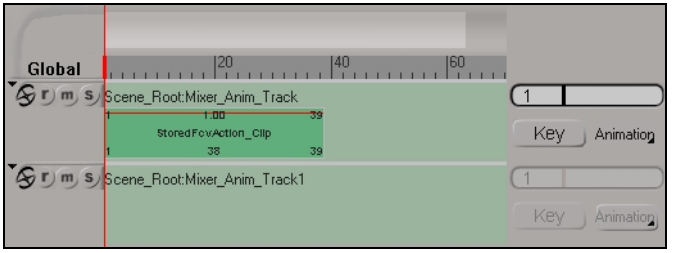
**1** **Start with motion-capture animation.**



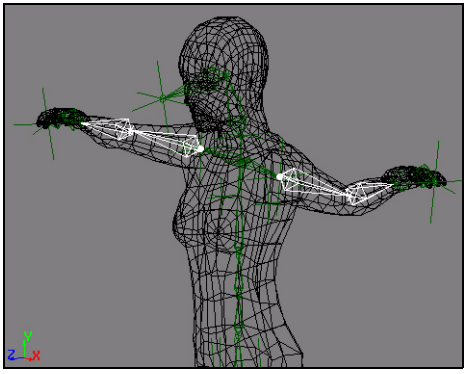
**2** **Clean up the function curves.**



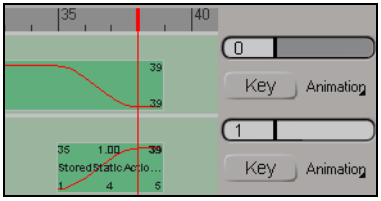
**3** **Create an action.**



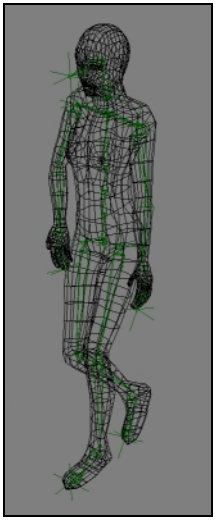

**4** **Store additional poses.**



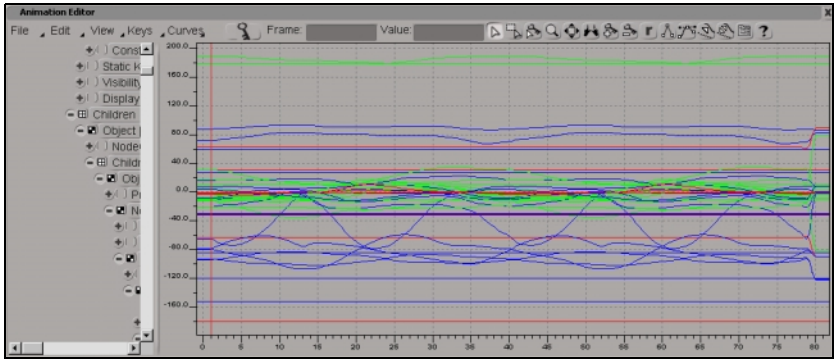
**5** **Mix actions by setting weights.**



## Processing Motion Capture Animation

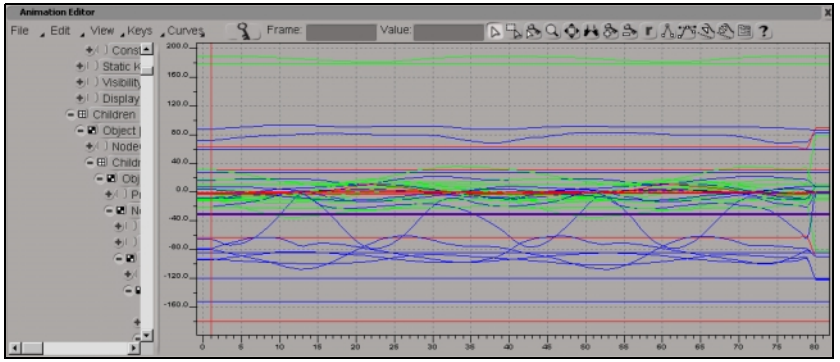


1. Open the **jaiqua_walk** scene from the tutorial database: **<install directory>\content\tutorial_project\Scenes**.
2. Play back the animation. Notice that a full walk cycle is 39 frames long.
3. Select the whole model hierarchy by right-clicking on it.
4. Choose **Animation > Animation Editor** from the Animation panel (bottom of the main window) or change one viewport to **Animation Editor**.




5. In the **View Menu** of the animation editor, make sure that **Animated Parameters** is on.
6. Choose **Edit > Select All Curves**.

7. Click the Curve Editor icon—it's the last one beside the Help icon (question mark).
8. Click the **Curve Processing** tab. This page gives you three options for processing curves:
   - You can **smooth** curves. This works in a similar way to blur on pixels. Smoothing tries to decrease the noise often seen on mocap (motion capture) files.
   - You can **fit** a curve onto the raw values. This reduces the key density of a curve while keeping the same overall curve shape.
   - You can **resample** the curve to add keys at regular time steps while retaining (or not) the existing keys.
9. In this example, you fit a curve to raw values. Enter **3** as the **Fitting Tolerance** value.
10. Click on the **Fit** button and wait for the process to finish.




11. Close the property editor and the animation editor.

## Storing Transformations

Actions can be thought of as encapsulated animations. You can create actions for the transformations or the marked parameters. For either of these options, you can store three types of action:

- The current "static" values (**Current**)
- Those parameters that are animated by keys (**FCurves**)
- Those parameters that are animated in any way, including function curves, expressions, constraints, and so on (**All Sources**)

### Marking Parameters

Creating actions based on marked parameters can increase performance because the actions don't contain unnecessary parameters. The list of parameters included in the action is also easier to manage when working with templates.

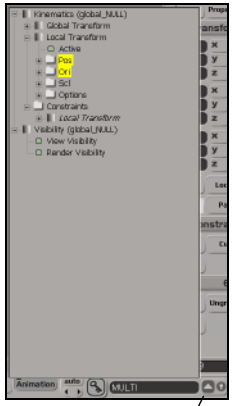
You should always mark local transformations when storing actions. Actions that contain local transformations are easier to work with. Remember that *Global always overrides Local!*

Before you store an action, always check that the marked parameter box shows **foo.local.bar** and *not* **foo.global.bar**.

Marking parameters can be quickly done by selecting an SRT tool. For example, if you press the **x** key to select the scale tool, **kine.local.scl** is automatically marked and appears in the box at the lower right of the main window.

If you click a specific axis on the Transform panel and want only that axis marked, you must be in Parent mode for scaling and translation, or Add mode in rotation. In other modes, all three axes will be marked even if only one axis is selected.

You can also use the animatable parameter list in the Animation panel at the lower-right of the window. Click the upward-pointing triangle to display a list of animatable parameters of the current selection, then click on any parameter to mark it. To add or subtract marked parameters, use the Shift+ or Ctrl+key modifiers. You can also use the Lock and Clr (Clear) icons just beside the Animatable Parameter List icon.




Opens the marked parameter list

12. In an explorer view, expand the jaiqua node, then expand the jaiqua_skeleton node, then middle-click on the COG node to branch-select it.

13. Click the Animatable Parameters icon, and expand the Local Transform node. Shift+click **Pos** and **Ori**.

    The marked-parameters box displays “MULTI” because more than one parameter is marked.

### Storing Animation

14. Choose **Actions** > **Store** > **Marked Parameters - FCurves** from the Animate toolbar. In the Stored Action dialog box, set **Default Out** to 39 (the length of the walk cycle), then click OK.

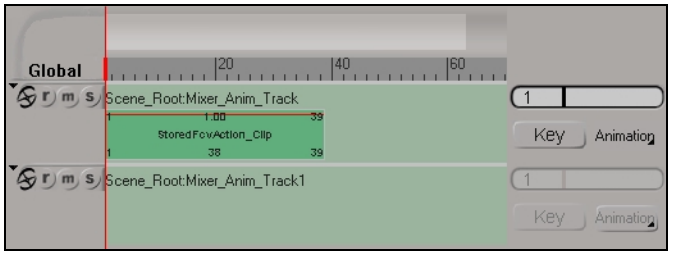
    The animation is disconnected from the model and becomes a source in the action library under the model (Mixer/Sources/Animation).

### Instantiating Clips

15. Open the animation mixer and click **Update**.

16. Position the mouse pointer near the start of the first track, then right-click and choose **Load Source**.

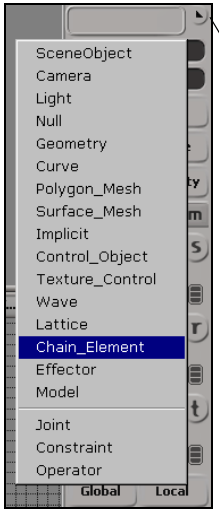
17. Pick the only source available from the menu (**StoredFcvAction**). The action clip is placed on the track and is represented by a green rectangle.




You can change the timespan shown in the animation mixer independently from the first and last frame of the animation. Simply enter new values in the Start and End boxes on the animation mixer command bar. However, remember that frames outside of the start and end frames of the scene will not play back.


## Storing Poses

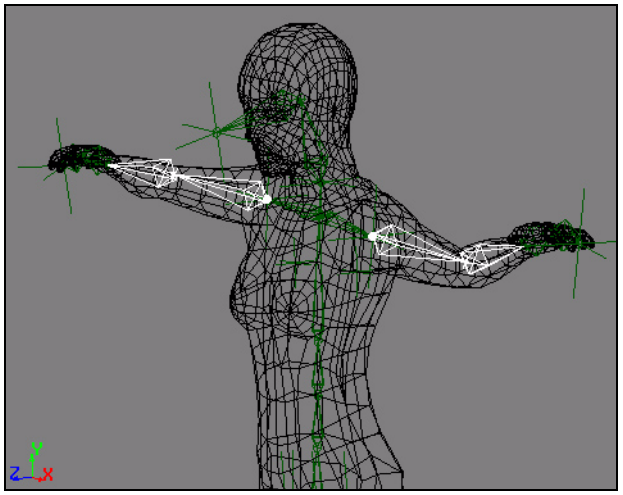
18. Now you will store some static poses. First click the **m** icon on the left of the animation track to temporarily mute the track as shown in the illustration. This prevents the action from updating the skeleton's transformations, such as if you accidentally jog the timeline.



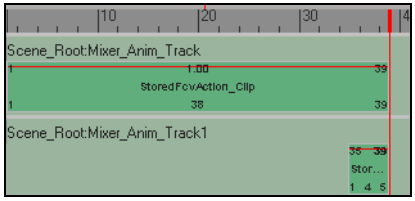

19. Choose **Chain_Element** from the Filter menu of the Selection panel on the main command area. (Click on the small arrow under the Sample button.)

20. To create some static positions, first select one of Jaiqua's upper arms and rotate it locally in Y into a neutral pose. Repeat for the other upper arm.

21. Ctrl+click to select all four arm bones as shown. Notice how easy it is to select only the bones with the **Chain_Element** selection filter on.



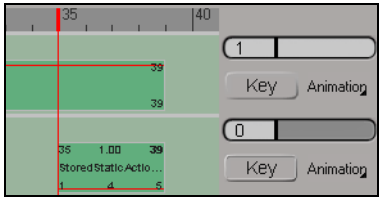
22. With these four bones selected, press the **c** key to mark the local rotations.
23. Save the current rotation values by choosing **Actions** > **Store** > **Marked Parameters - Current**. Click OK to accept the default values.

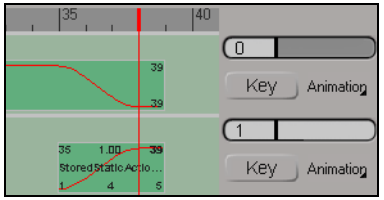



24. Now load this new pose in the animation mixer: right-click in the second track and choose **Load Source** > **StoredStaticAction**. Drag the clip so that it ends at the same time as the walk cycle.
25. Click the **m** icon again to "unmute" the first track, then play back the animation.

    Notice that the arms jump halfway to the neutral pose at the beginning of the second clip. You'll fix that by keying mix weights to blend the two clips in the next section.

## Mixing Clip Weights




26. Activate **View** > **Weight Mixer Panel** (if necessary) to show the weight sliders on the right side of the tracks.

27. Animate the weighting so that the first track fades over the second track as shown in the illustration:

    - Move the timeline to the beginning of the second clip. Click the top **Key** button to set a key on the first action's mix weight at its current value of 1. Set the second action's mix weight to 0 and click its **Key** button.
    - Move the timeline to the beginning of the last frame of the clips. Set the first action's mix weight to 0 and save a key. Set the second action's weight to 1 and save a key.

      If you don't see the weight curves in the mixer, make sure that **View** > **Weight Curves** is on.

## Creating Markers

You can add some markers to help organize the layout, or to set in/out points for loop playback:

28. Crtl+drag to select an area on a track.

29. Right-click in the newly created region and choose **Add Marker**.

30. Right-click again on the marker and choose **Properties**. Rename the marker **"foo."**

31. Right-click on the marker and choose **Set In-Out Loop**.

    Play back the scene—the marked frames play in a loop. To play the whole scene again, turn off the Loop option on the Playback panel.

## Conclusion

The animation mixer is a rich and powerful tool for manipulating and reusing animation in a non-linear and non-destructive way. In the next tutorial, you will see how to use the animation mixer to do shape animation.

For more information, see the *Animating* guide.

# Tutorial 11: Shape Animation

Shape animation is based on clusters and uses the animation mixer. There are several ways to save shapes in SOFTIMAGE|XSI:

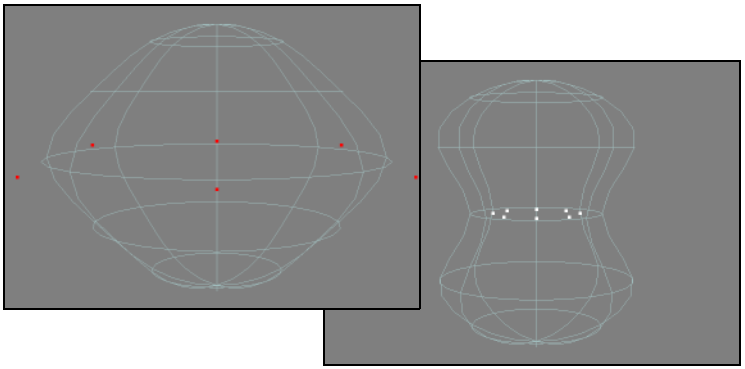
- You can simply store them and retrieve them when needed from the mixer folder.
- You can save a shape key at the current frame using Mixed Weight Mode, where each shape has its own shape track and is weighted against the other shapes.
- You can save shape keys using Transition Mode or Cardinal Transition Mode, where all shapes gets inserted in the animation mixer on the same track with automatic transitions between them.

This tutorial shows you how to:

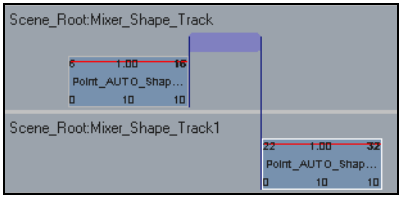
- Store shape sources and instantiate shape clips.
- Create transitions between shape clips.
- Use weights to mix shape clips.
- Create a custom property to control shape weights with sliders.

# Overview

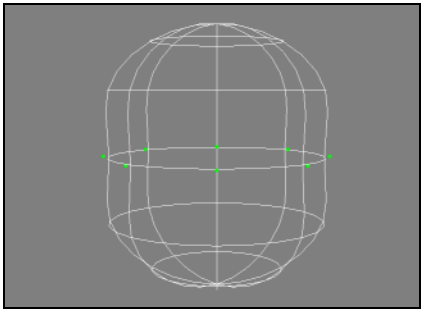
**1** **Create shapes.**



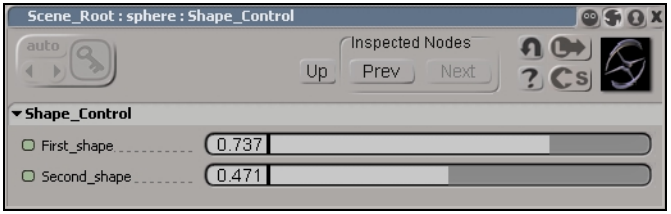
**2** **Create transitions between shapes.**




**3** **Mix shapes by adjusting their weights.**

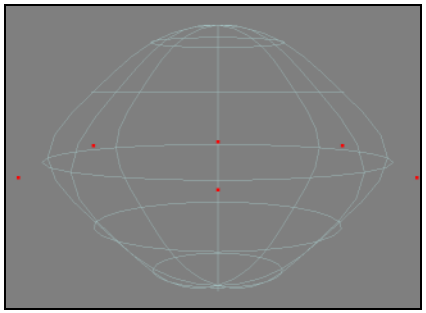


**4** **Control shapes using custom properties.**



## Deforming Points

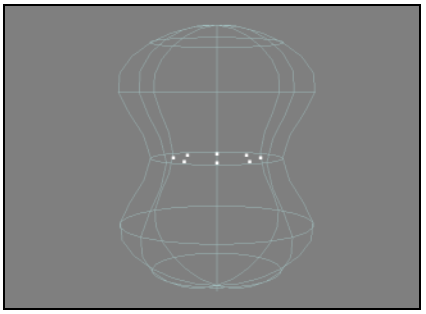
1. Press Ctrl+n to start a new scene.
2. Get a primitive sphere.
3. Tag the middle row of points with the **t** key and scale them with the **x** key.



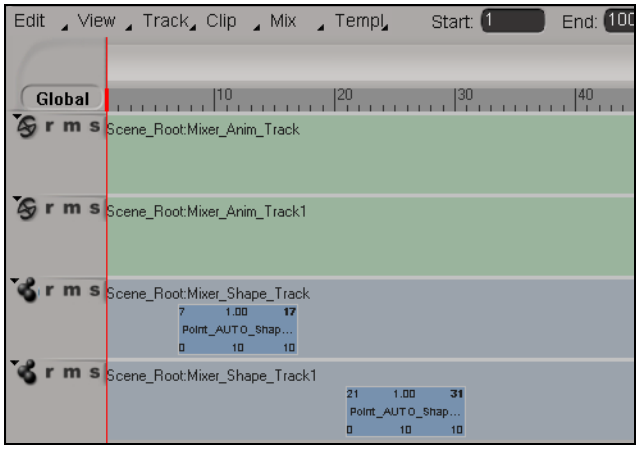
## Storing and Retrieving Shapes

4. On the Animate toolbar, choose **Deform** > **Shape** > **Store Shape Key**. This saves a shape source for later use.


5. Change the shape of the selected points again using the SRT tools.



6. Choose **Deform** > **Shape** > **Store Shape Key** again, or middle-click on **Deform** > **Shape** to repeat the last command.
7. Open the animation mixer and click **Update**.
8. Right-click on a track and choose **Add Track** > **Shape** twice to add two shape tracks.
9. Right-click on a shape track and choose **Load Source** > **Point_AUTO_ShapeAction1**.
10. Open an explorer and make sure that **Show** > **Mixers** is on. Expand the Scene_Root\Mixer\Sources\Shape node.
11. From the explorer, drag and drop the other shape onto the second shape track. For this example, make sure the clips don't overlap on the same frames—you can move a clip along a track by dragging it.

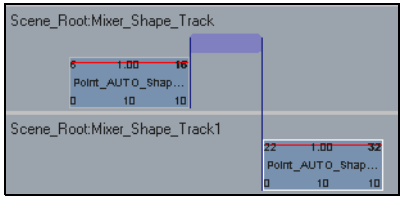



12. Play the animation. Notice how the shape snaps at the beginning of a clip.

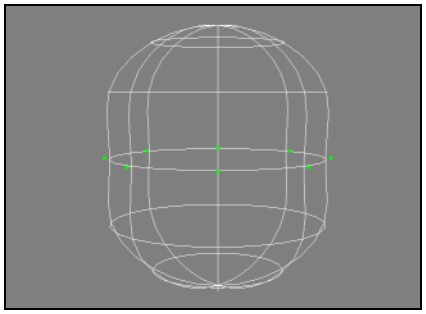
## Creating Transitions

Transitions can be created automatically when you create clips in the mixer. You can also add transitions afterwards individually.

13. Choose **Mix** > **Transition Tool** in the animation editor.
14. Pick the first clip and then the second clip.
15. Right-click to terminate the transition tool.



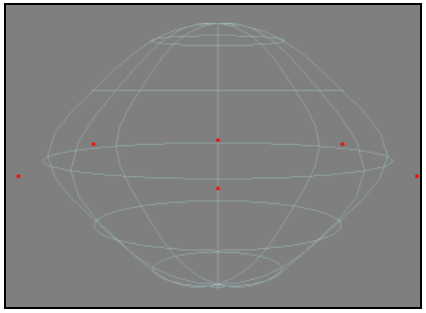

16. Play the animation again. Notice how the two shapes are blended in the transition between the clips.



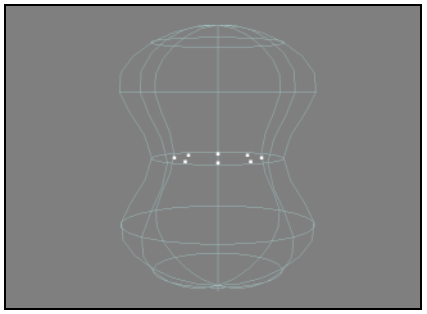
You can also choose **Mix** > **Auto Transition**. When auto transition is on, a transition is automatically created for each new clip in the mixer.

## Mixed Weight Mode

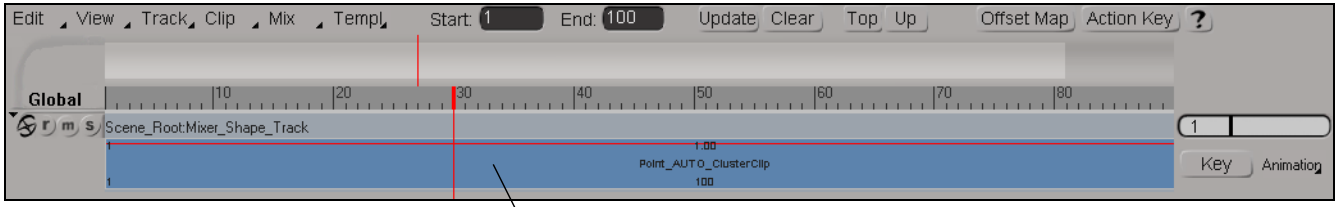
1. Press Crtl+n to start with a new scene.
2. Get a primitive sphere.
3. Select the middle row of points and scale them.



4. On the Animate toolbar, make sure that **Deform** > **Shape** > **Mixed Weight Mode** is on.
5. Go to frame 1.
6. Choose **Deform** > **Shape** > **Save Shape Key**. This stores a shape source, creates a shape clip, and sets a key with a value of 1 for the clip's weight at the current frame.
7. Change to frame 30, then modify the shape of the selected points again using the SRT tools.

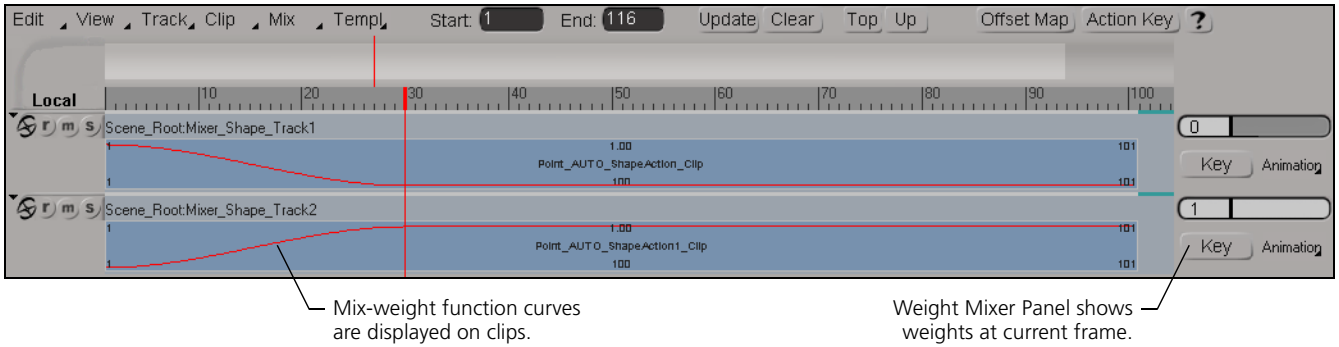


8. Choose **Deform** > **Shape** > **Save Shape Key** again.
9. Repeat as many times as you want, but two shapes are enough for this example.
10. Play back the animation.
11. Open the animation mixer and click **Update**.
12. Double-click on the compound clip to expand it. Note that each shape key is a clip on a separate track within the compound clip.

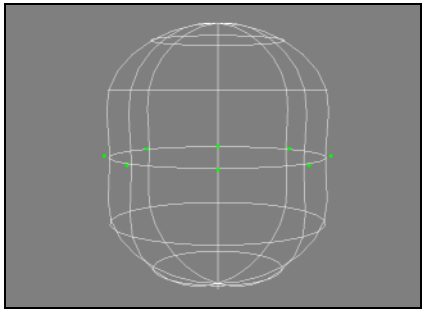



Double-click on the compound clip to open it.

13. Make sure that **View** > **Weight Curves** and **View** > **Weight Mixer Panel** are on. The weight mixer panel shows the value of each weight at the current frame.




14. Adjust the values in the weight mixer and click the **Key** button to set keys on the weight curves. Play back the animation to see the effect of the modified weights.



## Controlling Weights with Expressions

The shape weights can be keyed, but they can also be driven by an expression just like in Softimage *3D*. Remember those "virtual control panels," where objects control the shapes?

15. For each track in the mixer, delete all weight animation by choosing **Animation** > **Remove Animation** in the weight mixer panel. You will replace the function curves by a simple expression.

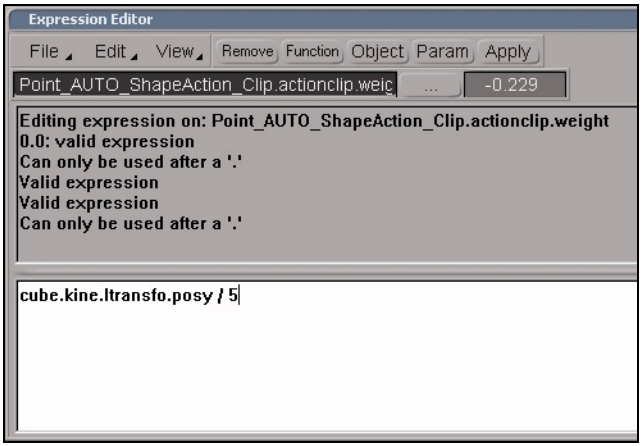
16. Get an implicit cube and scale it down to a manageable size.

17. On the first track's weight mixer panel, choose **Animation** > **Set Expression**.

18. In the expression editor, delete the default expression and type:

    `cube.`

19. With the text insertion point immediately after the period, press F12.
20. Choose **kine.** (for kinematics).
21. Press F12 and choose **global.** (for global transformation).
22. Press F12 and choose **posy** (for the Y position).
23. Divide everything by 5 (type **/ 5** at the end of the expression as in the illustration).




24. Click **Apply** and translate the cube on Y. Notice how the cube's position controls the weight of the shape key.

If nothing happens when you translate the cube on Y, it may be because all other shapes are weighted to 0. If this is the case, change the weighting of at least one of the shape weights to 1.

## Controlling Shape Weights via a Custom Property Set

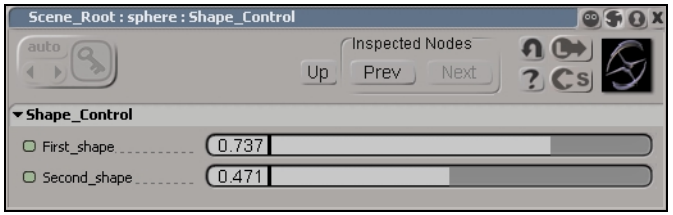
Instead of using an object to drive shape weights, you will replace it and create a property page with sliders.

25. Select the sphere with the weighted shapes.
26. On the Animate toolbar, choose **Create** > **Parameter** > **New Custom Parameter Set**. Enter `Shape_Control` as the name and click OK.

27. Click the **Property** button on the Selection panel in the main command area, and select the Shape_Control property.
28. Press Enter to open its property editor. The property editor displays an empty page. Lock the property editor so that it doesn't get recycled (click its lock icon).
29. Now add parameters: first, make sure that the **Shape Control** node is still selected in the explorer.
30. Choose **Create** > **Parameter** > **New Custom Parameter** and name this parameter **First shape**. Leave the other settings at their defaults. Notice that the parameter is added to the Shape Control page.

Do not use underscores (_)or special characters (#!%$#@) when naming custom parameters.

31. Choose **Create** > **Parameter** > **Custom Parameter** again and name this parameter **Second shape**. Again, leave the other settings at their defaults.
32. On the weight mixer panel of the animation mixer, choose **Animation** > **Expression Editor** (since there is already an expression on this parameter) on the first shape weight track.
33. Select the expression you had selected here and delete it.
34. With the text cursor in the white empty area of the expression editor, click the **Object** button and select **Sphere/Shape Control/First_shape**.
35. Click **Apply**.
36. Drag the first slider on the Shape Control property set and watch the shape change.
37. Repeat the previous steps to connect the second shape to the second slider. You can create more shapes and add more sliders if you like.




## Conclusion

You can create and control shapes in several different ways, using transitions, mixing weights, and so on. Custom parameters are a powerful tool for for controlling shapes and other things.

# Tutorial 12: Low-Res/High-Res Models

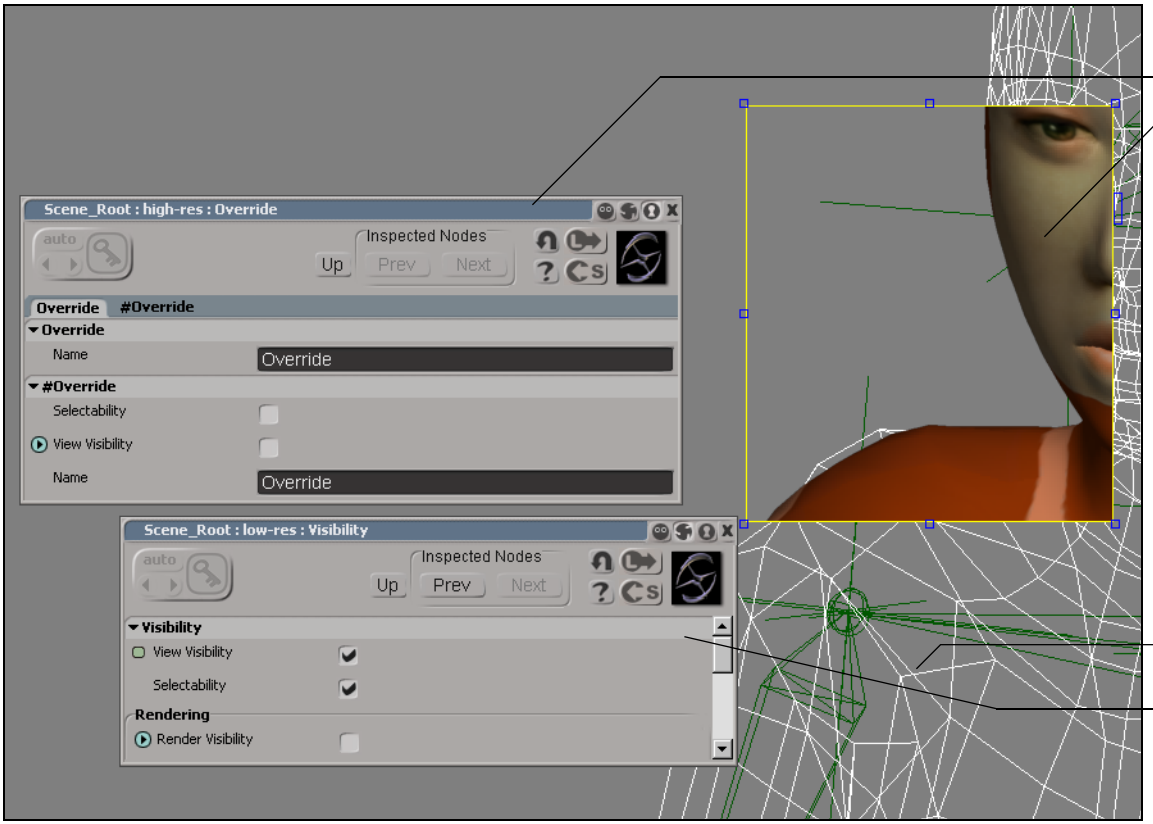
In the following short tutorial, you will set up visibility properties for two characters. One of the characters is a low-resolution polygonal model with its rendering visibility turned off so that it is visible only in the viewports. The other character is a high-resolution version of the same model—it is renderable, but not visible from the viewports. Both models are weighted to the same inverse kinematic (IK) chain.

If you are animating with complex characters, this technique is especially beneficial because hidden objects are not loaded into memory. This technique allows for a fast interaction with the IK because its uses a simpler or more generic character, and renders the high-resolution one.

This tutorial shows you how to:

- Use low-resolution models to speed up your work, and high-resolution ones to render.
- Use overrides to change property settings without destroying the original data.

# Overview




**High-Res Model**
Visibility is overridden on the high-res model and is only visible when rendered.

**Low-Res Model**
The low-polygon model is visible in the viewports but its rendering visibility is turned off.

## Creating an Override on the High-Resolution Character

1. Open the **RESOLUTION** scene from the tutorial database:
   **<install directory>\content\tutorial_project\Scenes**
2. In the explorer, expand the **jaiqua** node and select the **high-res** polygon mesh model.
3. Press the **h** key. This hides the object by creating an override for the visibility options.
4. With the pointer in the explorer view, press the **e** key to show the selection and all its nodes.
5. In the explorer, click on the Override icon to open its property editor.
6. Turn **Render Visibility** on and leave **View Visibility** off.

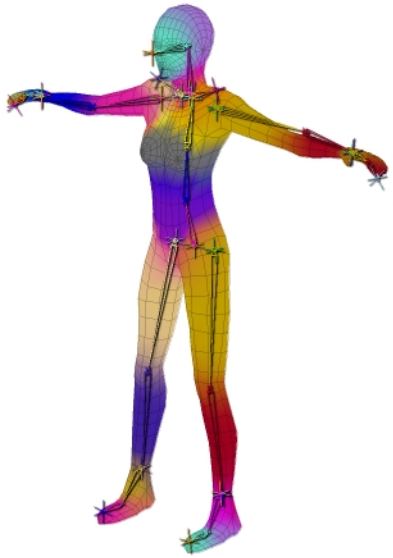
## Creating an Override on the Low-Resolution Character

7. Select the **low-res** polygon mesh model.
8. Press the **h** key again to create another visibility override.
9. In the explorer, click on the low-res model's Override icon to open its property editor.
10. This time, turn **View Visibility** on and leave **Render Visibility** off.
11. Draw a render region to see the results. The high-res model is rendered in the region, but only the low-res model is visible in the rest of the viewport.

### For Information, Please See....

The *Animating* guide.

# Tutorial 13: Envelope Weighting



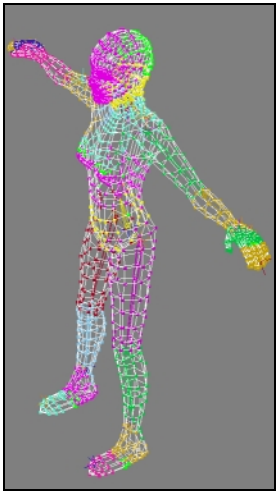
By attaching a surface or polygon mesh object as an envelope to a skeleton, the object deforms automatically as the skeleton moves. You can adjust the envelope weighting of individual points on the object to control its deformation more accurately.

This tutorial shows you how to:

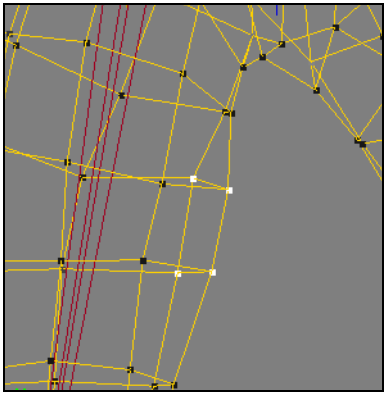
- Make a polygon mesh envelope.
- Reassign points locally to different bones.
- Adjust envelope weights manually.

# Overview

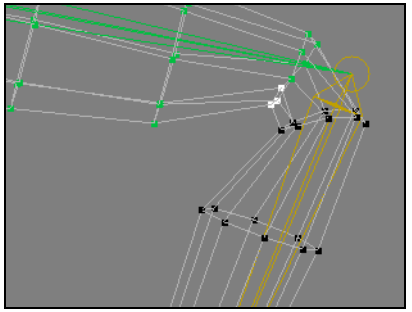
1 **Attach the envelope to the skeleton.**



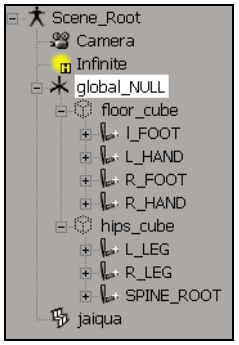
2 **Reassign points to other deformers.**



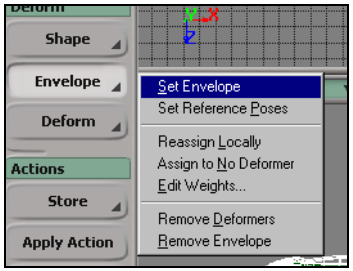
3 **Change the weighting of individual points.**



## Attaching the Envelope

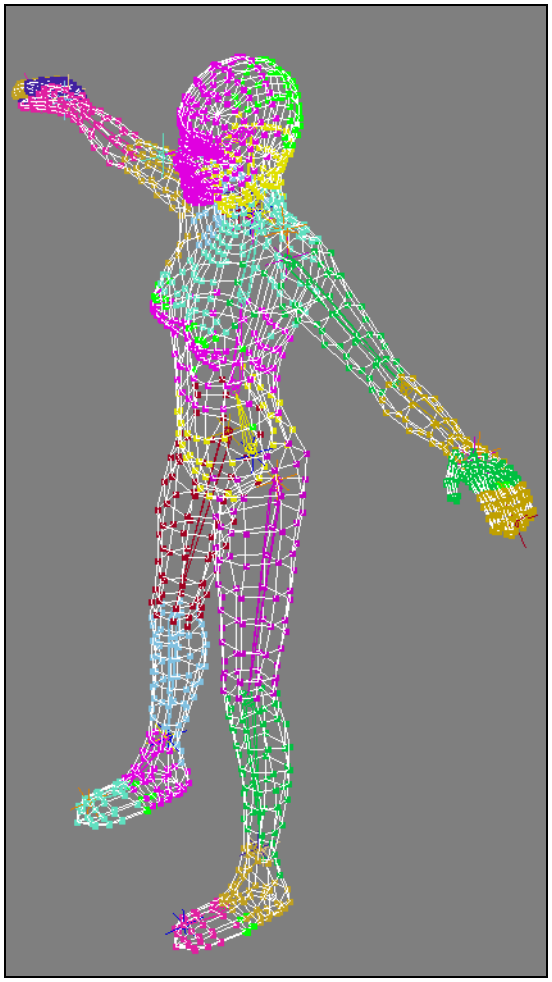



1. Load the **JAIQUA_IK** scene from the tutorial database:
   **<install directory>\content\tutorial_project\Scenes**
2. Open an explorer and expand the global_NULL node, then expand the floor_cube and hips_cube nodes.
3. Select the character's geometry (the **jaiqua** node).
4. Choose **Envelope** > **Set Envelope** from the Animate toolbar. In the Envelope dialog box, click OK to accept the default values.

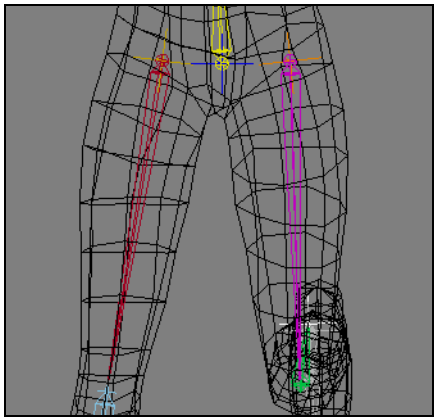


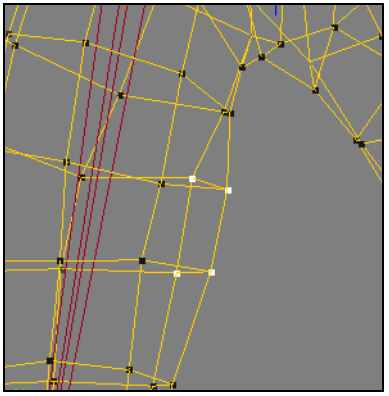

5. In the explorer, branch-pick (middle-click) the SPINE_ROOT node, the two FOOT nodes, the two HAND nodes, and the two LEG nodes.
6. Right-click in any 3D view to end the picking session. The Automatic Envelope Assignment property editor opens.


7. Accept the default automatic envelope settings and close the property editor.



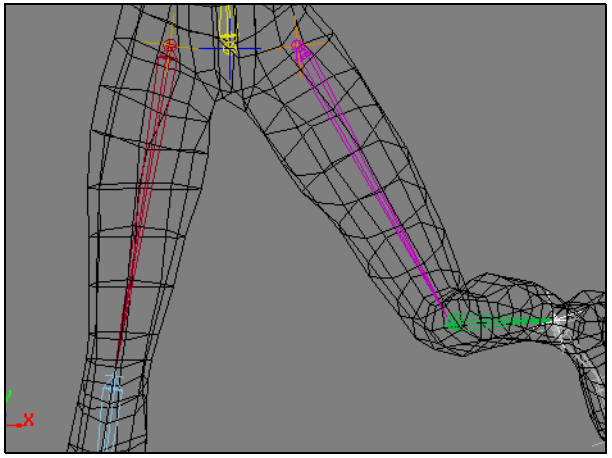
Cluster colors shows the different bone relation assignments of the envelope weights. To see them in a viewport, make sure that **Show** > **Clusters** is on.





## Adjusting the Weights in Extreme Poses

8. Branch-select one of the FOOT nodes in the explorer.
9. Translate the foot away from the other to pose the character in a split position as shown in the following illustration.

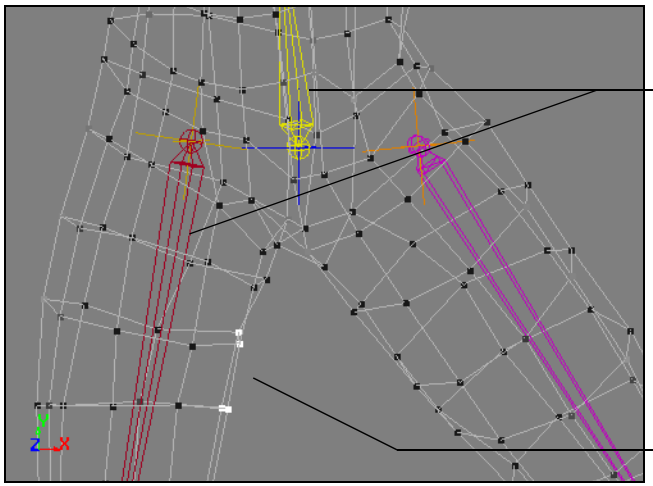


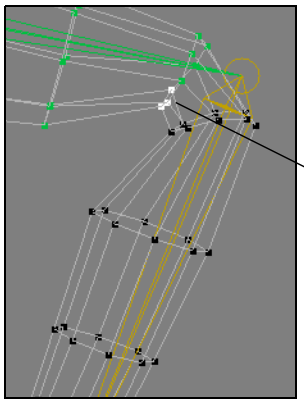
If only the foot is translated, press Ctrl+z to undo and branch-select from the leg effector (FOOT node) and not the foot effector (eff1 node). Use the explorer to make it easier to select.

Look at the points on the inner part of the thigh. When you translated the leg, some of the points on the other leg deformed as well. You will now fix this problem.

10. Tag the points that were not weighted as you expected and choose **Envelope** > **Reassign Locally** from the Animate toolbar.

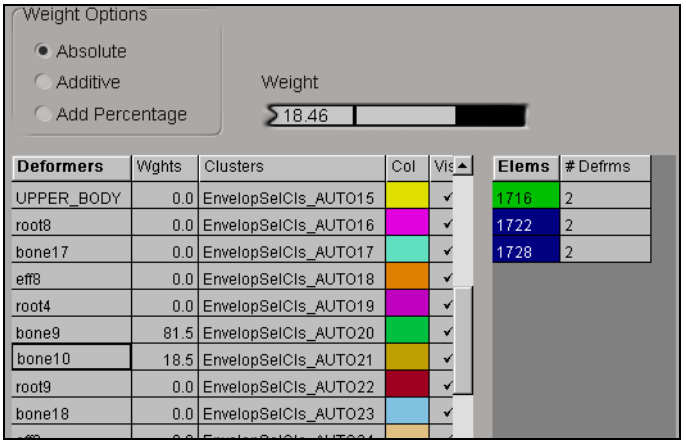
11. Pick the thigh bone and the first spine bone to distribute the envelope weighting of the points between those bones.



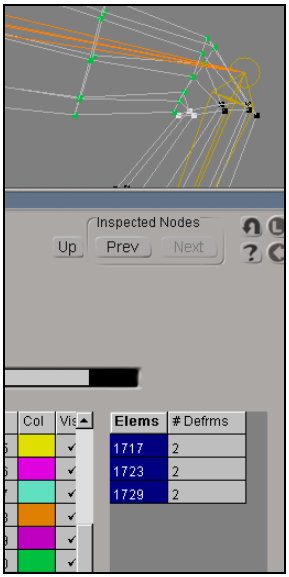

### Adjusting Weights on the Elbow

12. Branch-select one of the hands.
13. Translate the hand so the elbow bends to a normal rotation limit.
14. Tag the points that need to be reweighted properly. (Don't forget to untag the points on the thigh.)
15. Choose **Envelope** > **Edit Weights**. The tagged points are listed in the cluster list.




16. Shift-select all the points from the list under **Elems**.
17. Click on the **bone10** name under **Deformers** to make it active.
18. Drag the **Weight** slider to 18.

The weight is interactively adjusted to the other bones.




19. Keep this property editor open by clicking on its Lock icon. In the viewport, untag all points and tag only those as shown on the left. The weight property editor updates with current selection.
20. Shift-select all the tagged points from the list.
21. Find the upper arm bone by selecting different bones in the list (active bones are displayed in orange in the viewports).
22. Adjust the weights until you are satisfied with the results.

## Conclusion

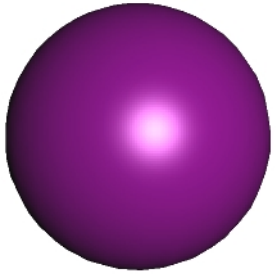
Using the various enveloping tools, you can easily apply an envelope to a skeleton, adjust the assignment, and modify the weights of individual points.

For more information, see the *Animating* guide.

# Section 5 **Rendering**

# Tutorial 14: Applying Materials

Applying materials, surfaces, and textures is the most fundamental step in achieving the look you want for your scene.



This tutorial covers the most-used surface attributes: materials, texture, volume, bump/displacement maps, volume shaders, and output shaders. By completing these steps you will get a good overview of the possibilities offered from the open architecture of the render tree, texture projections, groups, and layers.

The Material node in SOFTIMAGE|XSI doesn't have quite the same function as it does in SOFTIMAGE|3D. What were called *material* shaders are now named *surface* shaders. The Material node acts like a container for all of the possible shaders that can be applied to an object, much like the full-screen Material Editor in SOFTIMAGE|3D.

This tutorial shows you how to:

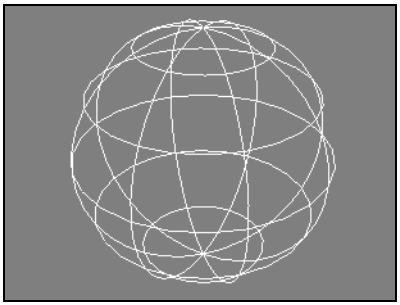
- Apply a material and surface shader.
- Create a local material.
- Edit a surface's properties.

## Creating a Simple Scene

You start by creating a sphere on a floor and applying a basic material to it. With this simple setup, you will also look at material sharing and propagation.

1. Choose **Get** > **Primitive** > **Surface** > **Sphere**. Accept the default values and close the sphere property editor.

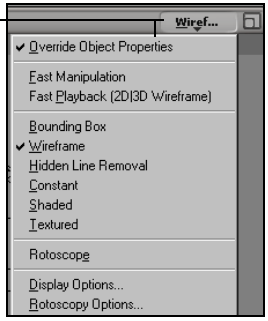
If you want to hide the viewport grid in the Camera view, simply position the cursor over the viewport and press **g**.

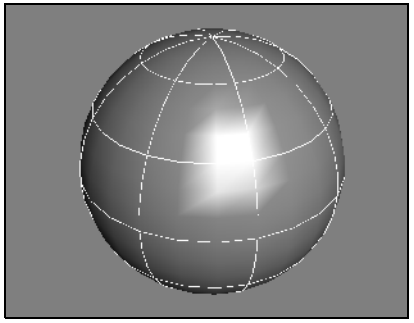


2. Choose **Get** > **Primitive** > **Surface** > **Grid**. Click the Grid tab and change the grid **UV Length** to the maximum allowed by the sliders (50) and close the grid property editor.
3. Position the grid under the sphere as though the sphere is on the floor.

   To do this, press the **v** key (translation supra key) and drag the grid below the sphere. Press **v** again to turn translation mode off.
4. Select **Shaded** from the viewport display menu in the Camera (perspective) view.

Viewport Display menu



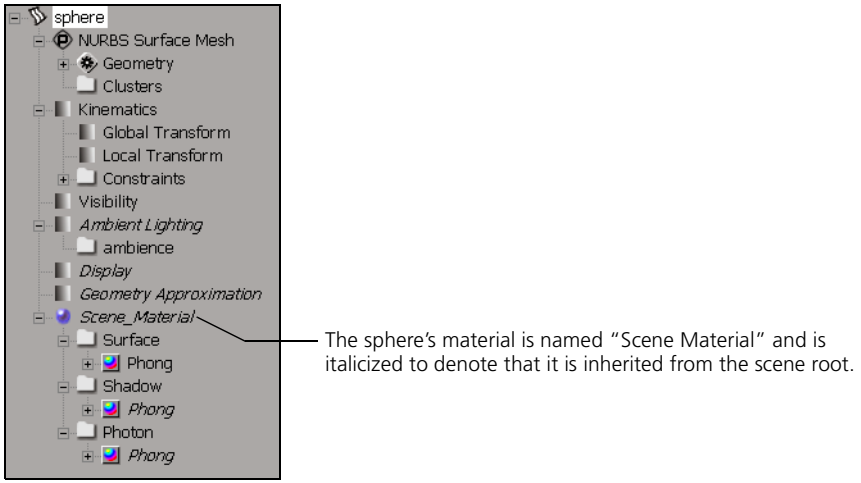
The wireframe of the selected object is still visible by default in Shaded display. If you wish to hide the wireframe or display the other side's wireframe, choose **Display Options** from the viewport display menu and select **Wireframe on Top** or **Transparency Shading**.

5. Position the camera to frame the sphere in the scene. To do this, place the mouse cursor over the perspective view and press **s** to activate the Combination tool that lets you orbit, track and dolly at once. Use the left mouse button to track, the middle to dolly, and the right to orbit.

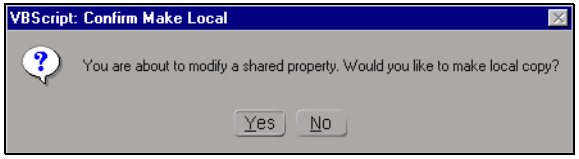
## Editing Shared Materials

The scene itself also has a material. This is what is called the Default Material and it is inherited by any object with no defined local material.

6. Select the sphere and click the **Property** button in the Selection panel of the main command area. In the explorer view that pops up, you see shared material nodes shown in italics.




7. Click the Phong node icon situated under *Scene_Material* > Surface to open its property editor. This is a common material that both the sphere and the grid have inherited from the scene. When a message box appears asking if you want to create a local copy, click No. This maintains the link to the inherited property, allowing you to edit the source's property.



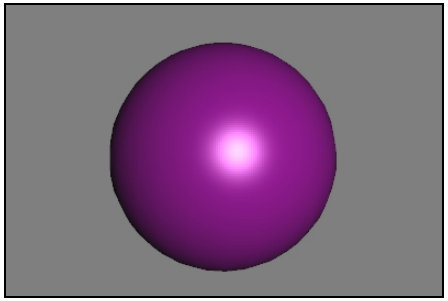
8. Modify the **Diffuse** color and see how it affects both objects in the shaded perspective view.

## Creating a Local Material

You can create a local material by repeating the steps you just did and choosing Yes in the message box when prompted to create a local material.

You can also do the following to create a local material:

9. Select the sphere and choose **Get** > **Material** > **Phong**. The Phong material's property page opens automatically.
10. In the Material:Phong property editor, modify the **Diffuse** color. The new material updates in the shaded perspective view. Close the property editor.



Once you apply a material, you can get a look at how it will be rendered by drawing a render region. Press **q** and drag a rectangle around the sphere. Middle-click to hide the region or press **Esc** to deactivate the region mode.

11. Choose **File** > **Save** to save this scene for the following tutorial.

## Conclusion

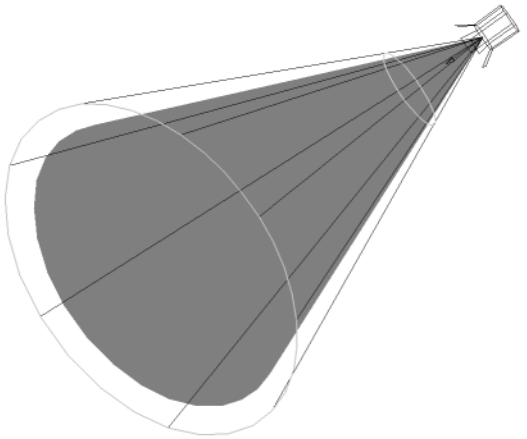
You have just applied a material (surface shader) to a simple geometric object. These are the basic steps for applying any shaders to objects in your scene.

Next you'll learn how to create, manipulate, and edit light properties in your scene.

For more information, see the *Shaders, Lights & Cameras* and *Rendering* guides.

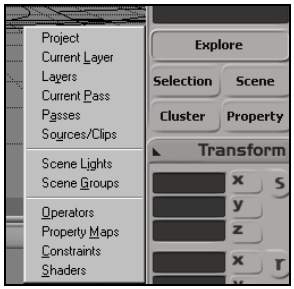
# Tutorial 15: Working with Lights

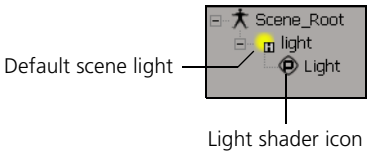
Lighting can really make or break a scene. This tutorial introduces you to the basics of working with lights. You will use the sphere and grid scene you created in the previous tutorial.



This tutorial shows you how to:

- Work with the default scene light.
- Create a new light.
- Manipulate and edit light properties.




## Editing the Default Light

Every new scene is supplied with a default light. You will edit this light.

1. Click the **Explore** button from the Selection panel of the main command area and choose **Lights** to open an explorer view of all the scene's lights. In this case, there is only the default scene light.

   The default light is hidden but renderable (hidden and renderable objects are depicted with a solid "H" in their explorer icon). Because it is an infinite light, it is sometimes difficult to see how something will really look using its default settings. You will change these settings later.

2. In the explorer, the default scene light icon is expanded to expose the light shader icon. Click the light shader icon to display the soft_light properties in the light property editor.

3. In the perspective view, draw a render region (press **q** and drag the mouse) around the sphere.

Did you know that you can speed up rendering in the render region? Hold the cursor over the upper-right rectangle on the region's border so that a slider appears. Drag the slider's widget down to lower the region's sampling and speed up rendering.

4. Modify the soft_light's color by dragging the RGB sliders. The region renders the effect of the soft_light color change on the sphere's color. Close the property editor.

In the next steps you will add a new light, but first you will delete the default light so that you can work with a single light source.

5. Click the **Explore** button from the Selection panel of the main command area and choose **Lights**. Select the Light (default scene light) by clicking its name.

6. Press the Delete key to delete the scene's default light. When the light is deleted, the scene goes dark and the region renders black.

## Adding a New Light

7. Add a new light to the scene by choosing **Get** > **Light** > **Spot.** The spotlight property editor appears—leave it open for now.

8. Press **v** and translate the spotlight away from the sphere.

9. If not already done, change the perspective viewport to Shaded display.

10. In the light property editor, try the following values:

| Parameter | Value | Function |
|---|---|---|
| Spread Angle | 20 | Determines the cone spread of the spotlight. |
| Light Falloff | On | Enables light falloff. |
| Start Falloff | 10 | Determines where the light will start to falloff. |
| End Falloff | 50 | Determines where the light ends completely. |
| Shadows Enable | On | Enables shadows. |
| Shadows Umbra | 0.1 | Defines the shadow's opacity or transparency. |

Wait until the render region is finished rendering the altered values. Keep in mind that every value you edit prompts a refresh of the render region.

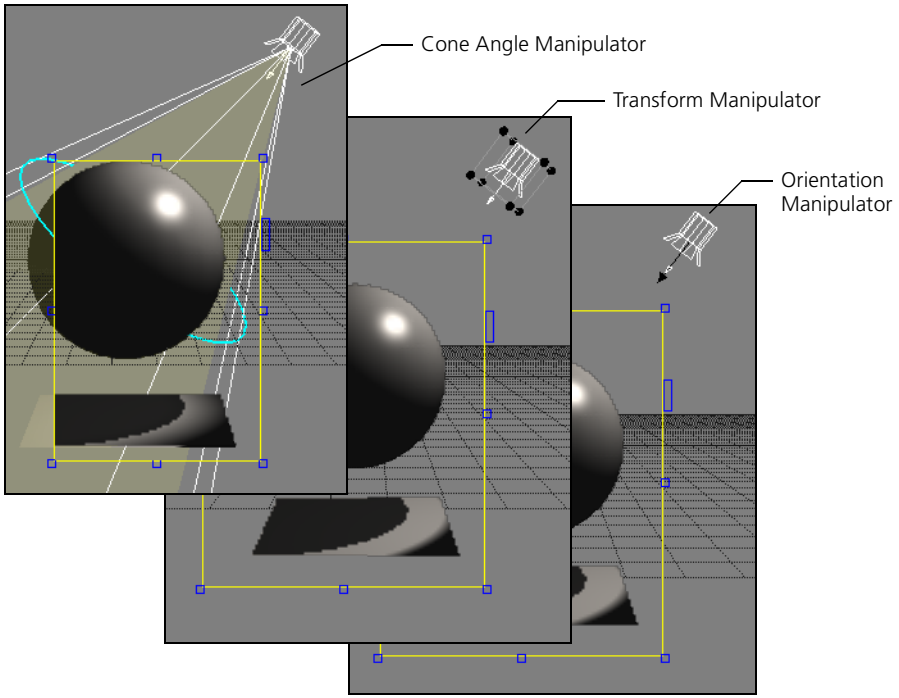
If the render region slows you down too much, you can set it to refresh on demand by choosing **Render** > **Region** > **Auto Refresh** on the Render toolbar to deactivate it. When you are ready to preview your changes, choose **Render** > **Refresh**.

11. Close the light property editor.

## Using the Light Manipulators

There are many ways in which you can manipulate and edit a light's properties: you will use the light's 3D manipulators.

12. From the Selection panel in the main command area, click the **Scene** button to display a list of all your scene's elements. Expand the **Spot_Root** node and select **Spot** (not its icon) to select the spot light.
13. Click anywhere outside the explorer to close it. With the mouse pointer in a viewport, press **a** to frame all the scene's elements.
14. Press **b** to activate the light's manipulators. Press the Tab key to toggle among the light's manipulators: angle, transform, and orientation.

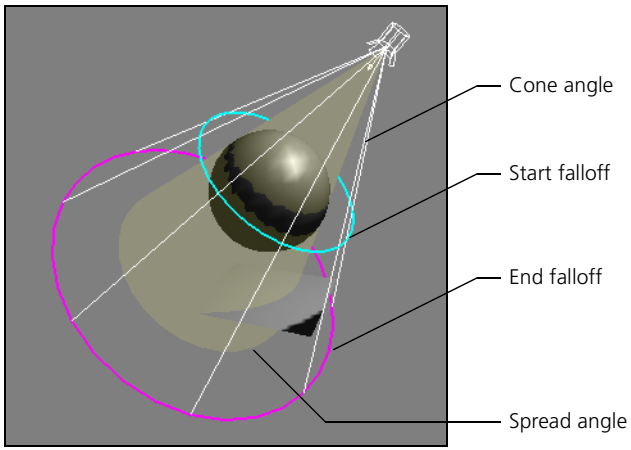



*Changing the falloff, angle, and spread*

15. Click and drag the blue and magenta lines of the spotlight's cone to adjust where the light starts to dim (blue) and where the light ends completely (magenta).

16. Modify the area the light cone covers by dragging any of the cone's white lines.

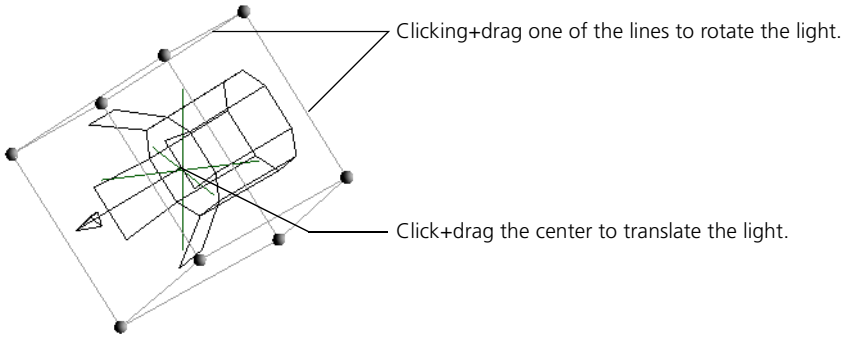
17. Modify how intense the light is within the cone (spread angle) by clicking and dragging the yellow area inside the cone.




### *Moving the light around*

Change the manipulator type by pressing the **Tab** key so that a bounding box with handles on each corner appears. The handles allow you to scale the spotlight, the lines allow you to rotate it about any axis, and dragging the icon's center lets you translate the light.

If you cannot see the center icon in the light, simply click the middle of the light icon and the center icon appears.



Scaling the light manipulators changes only the light's display icon. Scaling of the area of an area light can be done from the Light property editor.

18. Translate the spotlight by dragging the center icon of the bounding box.

### *Changing the orientation*

19. To change the orientation using the orientation manipulator, you must first delete the spotlight's interest to which it is constrained. Simply select the spotlight interest from the viewport or an explorer view (Click the **Scene** button, expand the **Spot_root** node and select the **Spot_interest**) and press **Delete**.

    A light's interest is useful but not essential. A spotlight's cone is always oriented toward the interest: where the interest moves, the spotlight's cone moves as well, but not the light source itself. Deleting a light's interest obliges you to move a spotlight's cone manually rather than manipulate its interest.

Manipulators are available even when an object is hidden. Select the hidden object (from an explorer view, for example) and press the manipulator key (**b**).

Choose **Edit** > **Delete All** from the Edit panel of the main command area to clear the screen for the next tutorial.

## Conclusion

You have created and manipulated a simple light object. You can use a light's property editor and its 3D manipulators to place and edit a light in a scene. Many of the tools you have used will be useful when manipulating several lights or editing their values to create caustic and/or volumic effects.

Now you are ready to apply textures to an object. Don't forget to light them!

For more information, see the *Shaders, Lights & Cameras* and *Rendering* guides.

# Tutorial 16: Giving Life to a Dragonfly

With the help of *Intorqueo-Volaticus*, Softimage's resident dragonfly, you will use SOFTIMAGE|XSI's rendering technology to finish a modeled scene.

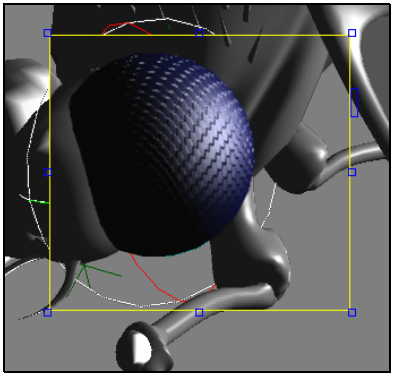
This tutorial shows you how to:

- Work with the default scene light.
- Use local visibility options.
- Apply a material preset.
- Use texture supports.
- Use image clips and sources.
- Understand propagation with groups.
- Build a simple render tree.
- Isolate a texture's alpha channel.
- Create a displacement map.
- Load a background scene.

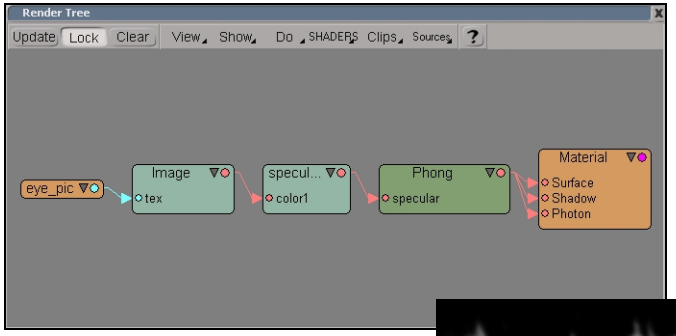
# Overview

**1** **Set parts of the display using the Palette.**

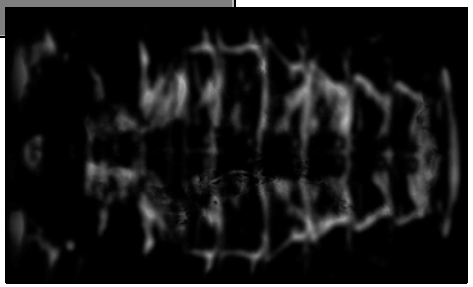

**2** **Apply a texture and texture projection.**



**3** **Use the render tree to apply a texture to a parameter.**




**4** **Use the alpha channel to control the displacement of a texture.**



## Setting Display Properties

Even with a small scene like the one you will load, you can set display attributes for some objects to enhance your workflow and improve the interactivity of your scene. You will create a group that contains all of the small polygonal cones that make up the dragonfly's hairs and set a display option that will show these objects in bounding box form when not selected or far from the camera.

Object name text box

1. Choose **File** > **Open** from the main-menu bar and select the **DRAGONFLY** scene from the tutorial project: **<install directory>\content\TUTORIAL_PROJECT\Scenes**
2. Using a wildcard (*) allows a fast multi-selection based on object names. Type `cone*` (and press Enter) in the object name text box on the Selection panel to select every object whose name starts with "cone". Notice that all the hairs on the dragonfly's back are selected.

### *Group polygon mesh objects*

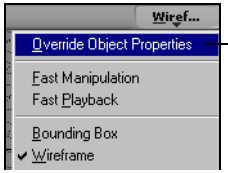
Grouping all of these polygonal objects into a single group allows you to apply a single property to the group rather than applying it to each of the cones, which saves a lot of time.

3. With all the cones still selected, click the **Group** button on the Edit panel of the main command area. Enter `hair_group` in the Group Name text box and close the property editor.

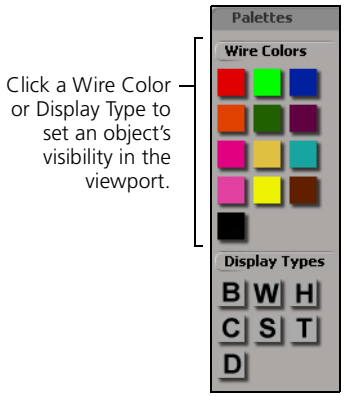
### *Change the display type for the group*

You'll set the display type for all the objects in the hair_group to bounding box when not selected or far from the camera.

4. With the hair_group still selected, choose **Get** > **Property** > **Display** from the Render toolbar.
5. In the Display property editor that appears, select **Bounding Box** for all options under **Unselected, Near** and **Unselected, Far**. Close the property editor.
6. By default, local display properties are overridden by the display option defined for the viewport. Deselect the **Override Object Properties** option in the viewport display menu so you can see the viewing modes you have just set.



The **Override Object Properties** option overrides an object's display mode with what is defined in the viewport display menu (wireframe, shaded etc.).





7. Place your cursor in the perspective window and press the **z** key (Pan and Zoom supra key). Middle-click to zoom in on the hair_group. Press **Esc** to deactivate the Pan and Zoom tool.
8. Deselect the group by clicking (and dragging) in an empty part of the viewport. All the hairs are displayed in bounding box when unselected.
9. Select a single hair. It is displayed in standard wireframe when selected.

*Change the wireframe color*

You can edit the wireframe color of objects displayed in the viewports from the Palette toolbar.

10. Click the Palette button at the bottom of the current toolbar to open the palette options.
11. Click a color from the Wire Colors panel and pick the dragonfly body. It's wireframe is displayed in the chosen color.

    You can left-click to color an object or middle-click to color a branch. You can select as many objects or branches as you wish.
12. Click the right mouse-button when you are finished picking.

## Applying Surface Shaders

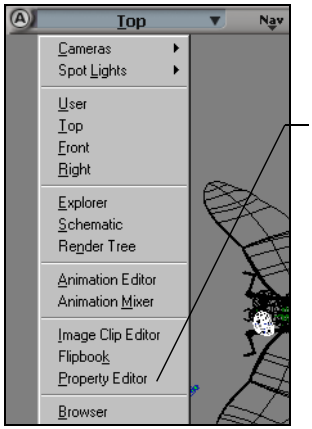
Any modification in a property page can be saved as a preset. This means you can, for example, easily transfer a surface preset to another object. But first, you'll apply a surface shader.

13. In the perspective view, select one of the dragonfly's eyes.
14. With the mouse pointer in a perspective view, frame the selected eye by pressing the **f** key.
15. Draw a render region over the selected eye (press **q** and drag).

To remove a render region from a viewport, press **q** and click anywhere within the viewport.

The selected dragonfly eye (and the rest of the model, for that matter) does not have any material of its own. The material it currently displays is inherited from the default scene material. Start by giving it a local material.

16. Change a viewport to a **Property Editor** view. This makes property editors automatically appear in a viewport rather than floating on the interface. This is a good way to prevent too many windows opening and closing during this tutorial.



17. From the Render toolbar, choose **Get** > **Material** > **Phong** to apply the default surface shader to the eye. The Phong Shader property editor opens in the property editor view in the viewport.

18. Hold the Ctrl key and drag one of the **Ambient color** RGB sliders to dark gray.

Holding the **Ctrl** key while dragging a color slider allows you to drag them all at once.

19. Edit the **Diffuse** color until it is a dark blue.

One method to define an extreme color range is to adjust a color in RGB mode, then switch to HLS and lower the L value (luminosity).

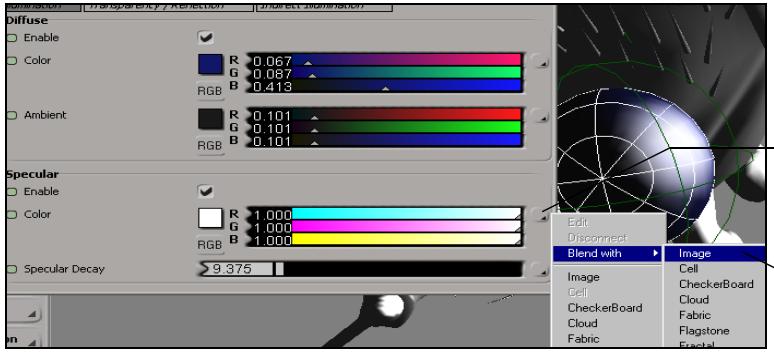
20. Lower the **Specular Decay** to about 12. This washes out the eye with a big white specular highlight, but you will correct it a little later. Leave the property editor open for now.

## Connecting an Image to a Parameter

You can connect almost any type of shader to any parameter. Fractal shaders can be connected to a transparency value, or cloud texture shaders can "drive" a surface's reflection value instead of using a solid color.

You will use a specular map to modify the specular highlights on the eye and add a small bump value to enhance the highlights. The specular map will be added with an image shader. An image shader lets you use any texture or image file as a 2D texture or bump map.

21. From the Phong surface shader property editor, click the connection icon (the little plug) beside the **Specular Color** slider and choose **Blend With** > **Image** from the menu that appears.




Click the connection icon to display a list of possible connections for a particular parameter.

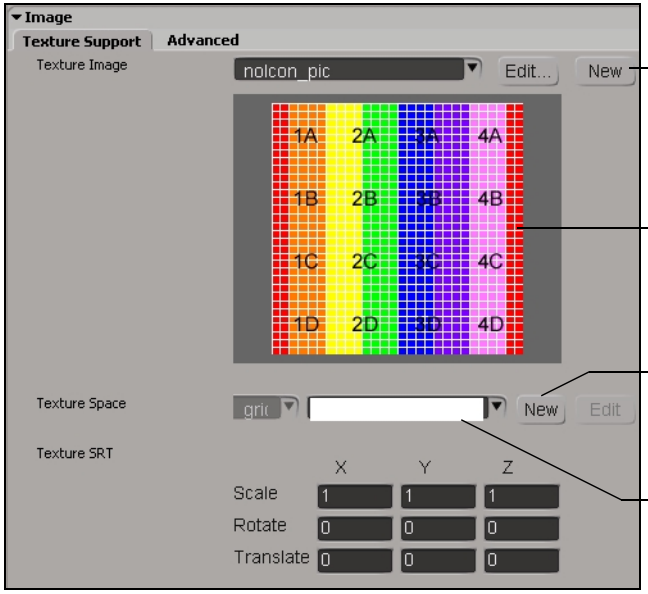
Choose **Blend With > Image**

An image shader is connected to the **Specular Color** parameter via a mixing shader which blends two or more shaders. The mix_8_colors shader gives you greater control over textures and how they are blended. The mixing shader will be discussed later on.

## Applying a New Texture

In SOFTIMAGE|XSI, most textures are applied using the image shader. You can use this shader to define a texture map, bump map, and texture projection.

22. The image shader property page should now be displayed in the property editor. It displays the default **no_icon** texture. Click the **New** button beside the image name and choose **New From File** from the pop-up menu. A browser appears from which you can select an image file.




23. From the browser, navigate to the tutorial_project's Pictures folder and select the **eye.pic** file.

In any browser, you can quickly jump between projects or folders using the Path button in the top right of the browser.

### *Define a texture space*

Texture space defines how a texture is placed on an object. In this example, the eye object does not yet have a defined texture space: that is your next step.

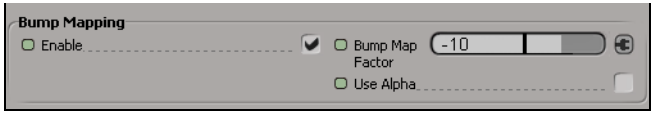
24. Since no texture space has been created, the object does not have any available texture spaces in the menu. On the image shader's property page, click the **New** button beside the **Texture Space** text box and select **Spherical** in the pop-up menu.

    The Spherical texture space is automatically applied to the selected object, meaning that the texture will be "wrapped" around the object as though it were a sphere, which it happens to be in this case.

25. Double-click in the Texture Space text box and rename the texture space to `spherical`.

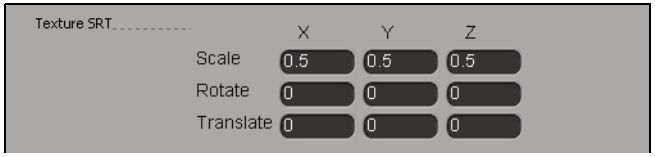
### *Enable Bump mapping*

26. **Enable** the **Bump Mapping** and set the **Bump Map Factor** to –10. A negative value bumps the texture inward, whereas a positive value bumps outward.




### *Scale a texture*

27. In the Texture SRT section (transformations) of the image shader property page, enter `0.5` for the **Scale XYZ** values.



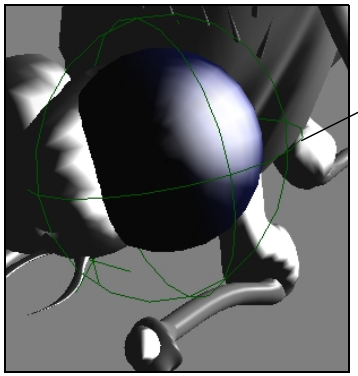

28. Click the **Prev** (previous) button to go back to the Phong surface shader property page.

29. Adjust the **Specular Decay** to a low value, and watch the render region refresh with the updates. You have just created a simple specular map.

## Manipulating the Texture Support

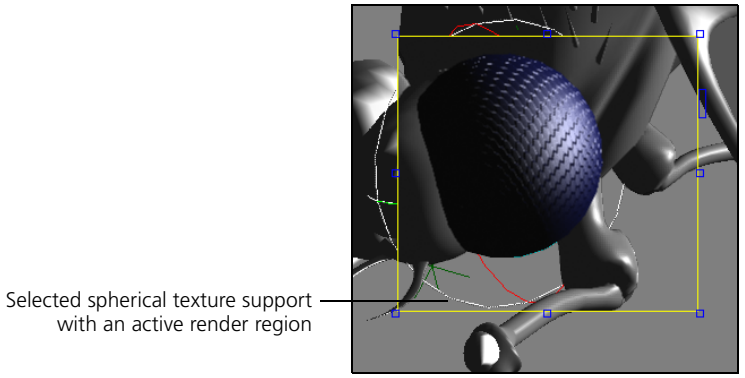
Once you've applied a texture and edited it, you can use the texture support object to interactively manipulate the texture space in a viewport.

30. You should see the green spherical texture support controls around the eye object. If they are not visible, press **j**. If you still cannot see the texture control objects, select **Show** > **Texture Controls** from the viewport.




31. Select the texture support object from any viewport—it becomes white.

32. Press the rotation supra key (**c**) and rotate the texture support object so that the "pinched point" of the spherical map is not visible (as shown below) and turned toward the dragonfly's body.

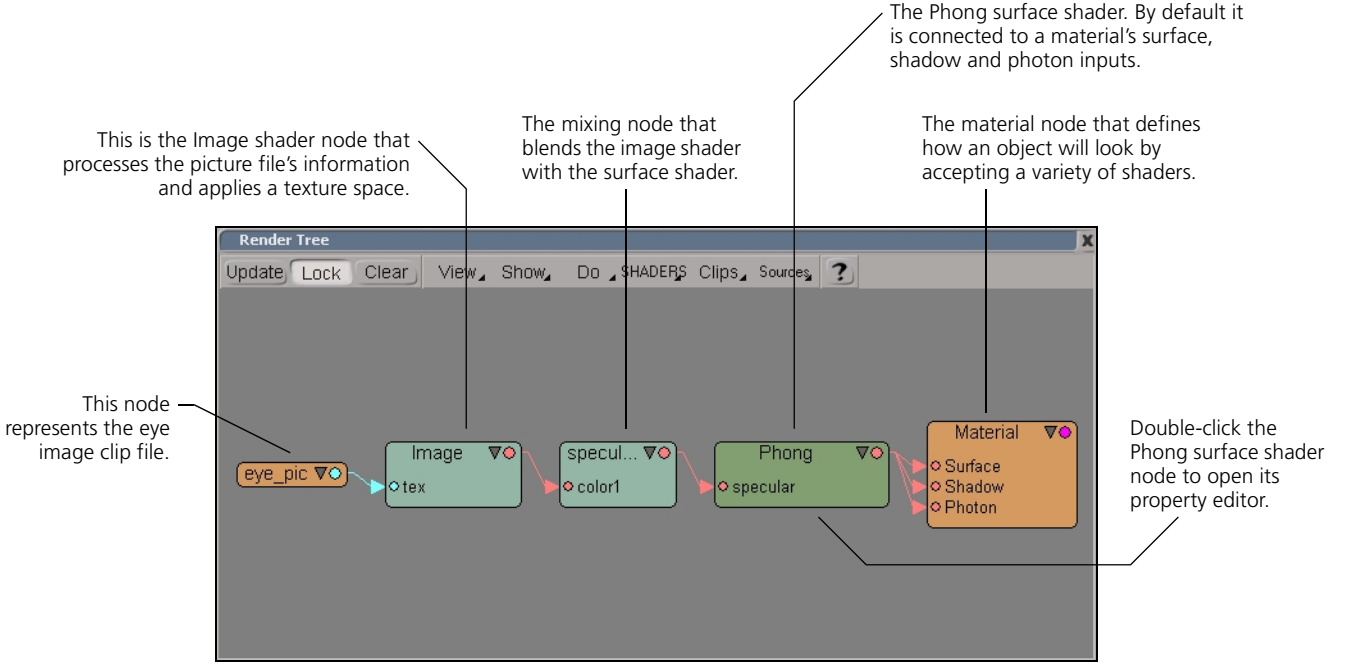


## Viewing Shader Connections in the Render Tree

The render tree gives a graphical representation of shader connections and their relationships, in this case, the texture shader. The selected object's surface shaders are displayed as nodes in the render tree. You use the render tree to create custom render trees by connecting textures and tools to create specialized effects.

33. With the Eye object selected, choose **View** > **Views** > **Render Tree** from the main-menu bar.

    A floating render tree window appears. By default, it shows the selected object materials tree, as in the following illustration. If you select a new object and want to view its render tree, click the **Update** button.



## Saving a Texture Preset

Before saving a texture preset, you may have to make some slight adjustments to ensure that your preset is completely compatible with the object to which you will apply your preset. In this example, you will have to apply a wildcard to the texture space's name and make the eye.pic's path relative so any object you load the texture preset onto will know how and what to apply.

### *Wildcards and texture support names*

When loading a texture preset onto a new object, the texture preset looks for a "spherical" texture support on the new object. If the texture support name does not exist for the object, the default support is used, which is most likely not the one you need. To solve this, you can use a wildcard (*) with the **Texture Space** name.

34. To be certain that the saved Phong preset is fully compatible with other objects, insert a wildcard (*) in the **Texture Space**'s text box of the image shader's property page before saving it.

A wildcard can be placed only at the start or end of the name.

### *Relative paths*

For the preset to know where it must retrieve the eye.pic image file from, you must give it a relative path so it can find the image file.

In the following steps, you will open the property pages of each of the above nodes in the render tree to edit some shaders' settings. You can open a shader simply by double-clicking its node in the render tree.

35. Double-click the Phong surface shader node in the render tree to open its property page.

    Notice how the Specular Color parameter is connected to another shader (done earlier), shown by the connection icon that is red and "plugged in".

36. Click the Specular Color's connection icon to open the Mixer shader (specular_blend) property editor.

37. In the Mixer property editor, click the **Layer 1 Color** connection icon to open the Image texture shader property editor.

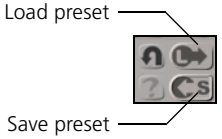
38. In the Image property editor, replace the **Texture Space** by `s*`. This makes the texture accept any projection that has a name that begins with "`s`".

39. Click on the **Edit** button beside **Texture Image** to open the image clip property editor.

40. In the image clip property editor, make sure that the path for the image is relative to the project. In the **File Name** text box, make sure the source file is local.

You can set a variable and use it in your path, but if you start the path with a back slash as in the following path **`\Pictures\eye.pic`**, the Pictures folder is relative to the current project location.

*Save a preset*




41. Click the **Prev** (Previous) button until you open the Phong Surface shader property editor.
42. Click on the save preset icon to save the render tree connected to the Phong shader. Saving a node as a preset saves everything connected to it. In this example, saving the Phong surface shader as a preset saves the Phong attributes, the mixing shader, the image shader, and the eye picture file.
43. Select the **Use Region as Thumbnail** option in the Save dialog box. This uses the current render region as a thumbnail for the preset in the browser. If this option is dimmed, create a render region in a viewport to enable it.
44. Enter `dark_eye_mat` for the preset name and click Ok.

*Create a texture support for the other eye*

Before applying the preset to the other eye, you need to create a texture support for the other eye.

45. Select the dragonfly's other eye.
46. From the Render toolbar, choose **Property** > **Texture Support** > **Create Texture Control**.
47. Select **Spherical** as the **Projection Type** and leave the default **Texture Support** name. Click OK.

*Drag and drop the preset*

On IRIX, you can drag and drop only from SOFTIMAGE|XSI browsers.

48. Open a browser from a viewport or from **View** > **Views** > **Browser** on the main-menu bar.
49. Navigate to where you saved the `dark_eye_mat` preset. Drag and drop the preset from the browser onto the dragonfly's other eye. All settings and connections made to the first eye are now applied to the second one.

When using drag and drop, wait for the cursor to change to a plus sign (+) inside a white square.

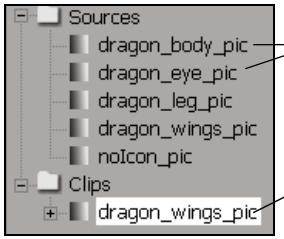
## Loading Image Sources and Clips

When you load an image into SOFTIMAGE|XSI, you create an *image source*. The same source can be used as many times as you wish. An *image clip* is an instance of an image source. You can crop or edit an image clip without affecting its image source, which is a read-only file.

To create an image source, you can drag and drop images directly into any viewport, as you will in the following step. You can also use the **New** image clip button in any image shader or the Clips menu on the Render toolbar. When creating a new clip, you are prompted to browse for an image. When an image is selected, a new source is created for it, allowing you to use a first instance of it.

Use the **Source and Clip** focus in the explorer (**i** shortcut key) for quick access to these files.

50. Open a browser and drag and drop (Shift+select them) the four **dragon_*.pic** files from the **<install directory>/Content/tutorial_project/Pictures** folder into a viewport. The four files are: dragon_body, dragon_eye, dragon_leg, and dragon_wings.




Dropping the images onto the render tree automatically assigns them an image shader and places them in the render tree work area where they are ready to use. Also, a source is created for each image and is available from the Sources and Clips folder in the explorer.

To get a better look at an image source or clip, you can view it from the Flipbook. Change one of the viewports to the **Flipbook** view and choose **Sources** or **Clips** from the command bar and select an image file.

## Adding a Wing Texture (Transparency)

Now you'll multi-select all four wings and map a 2D texture to the Diffuse color. This will be done in a similar way to how you mapped the specular texture to the dragonfly's eye.

51. Type `wing*` in the Current Selection text box on the Selection panel of the main command area. The wildcard (*) selects all objects with the word "wing" in the object name.

### *Define a texture support*

52. From the Render toolbar, choose **Get** > **Property** > **Texture Support** > **Create Texture Control**.

53. In the Create Texture Support dialog box, select the Planar XZ **Projection Type** and enter `WingXZ` as the texture support name. An XZ planar texture support is created for each of the selected wings.

### *Create a group for the wings*

54. With all four wings still selected, click the **Group** button on the Edit panel from the main command area. Name the group as you wish, such as `wings`.

If you multi-select several objects and apply a texture projection, you will create a texture projection for each item in the group. If you were to define a texture projection to the group (single selected), a single texture projection would be created for all the objects.

### *Apply a material and texture*

55. Apply a surface shader to the new wings group by choosing **Get** > **Material** > **Phong.**

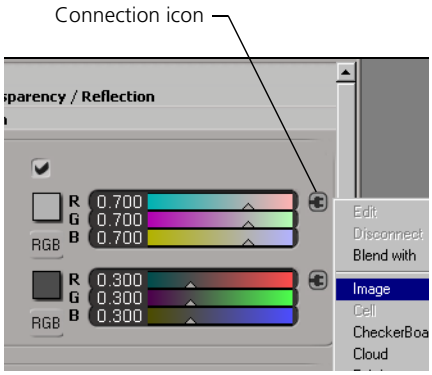
56. In the Phong surface shader property page, click the connection icon beside the **Diffuse Color** sliders and select **Image**. The Image shader property page opens.

57. From the Image shader property page, select the **Dragon_wings.pic** image clip from the **New** > **New From Source** pop-up menu. This image is already defined as an image source because it was loaded earlier when you dragged and dropped the files into the SOFTIMAGE|XSI window.

58. From the **Texture Space** menu, click the arrow and select the texture space you created for the wings earlier, named WingXZ.

59. Draw a render region (press **q**) around a wing in a perspective view to render a wing's texture.

    Make sure the texture controls are visible from the viewports. If they aren't, press **j** to turn their visibility on.



60. Select a texture support on one of the wings. The selected texture support's name is then listed in the Selection text box.

61. Replace the number at the end of the texture support name with a wildcard (such as `wingXZ1` to `wingZX*`) and press Enter. This multi-selects all the wings' texture supports.


62. If not already open, press Alt+Enter to open the Texture supports' multi-property page. This is a property page that contains the properties for all of the selected objects at once.

63. Select the **Swap UV** option and enter `-1` in the **Z Scaling** text box. This flips and rotates the wings' texture supports and scales them so they fit the wing model properly.

## Using the Alpha Channel

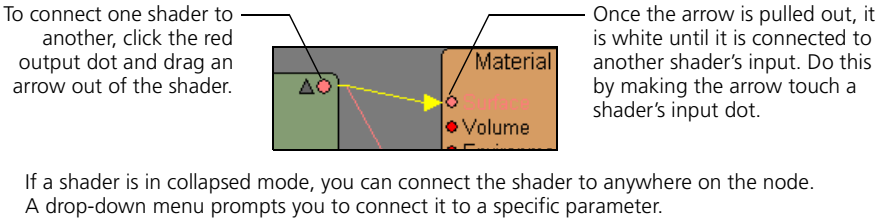
In the following steps, you will isolate the alpha channel using the render tree and use it to define the transparency of various textures.

64. Select the wings' group from the explorer.

65. If not already open, open a render tree and click **Update** in the render tree command bar. This once again displays the wings' group and the shaders attached to it.

66. From the command bar, choose **Nodes** > **Conversion** > **Color to Alpha**. This makes the Color2Alpha shader appear in the render tree work area. This tool shader extracts the alpha channel from an color (RGBA) image.

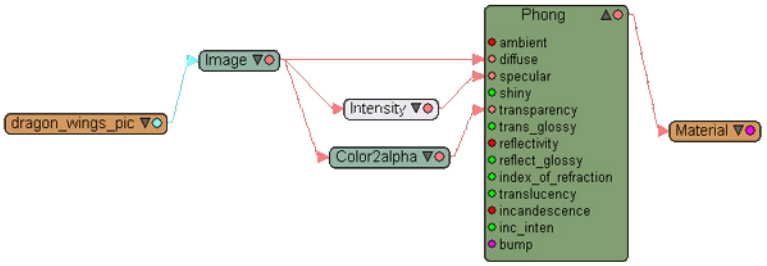
67. Drag the output arrow from the red dot of the **Image** shader node to the input of the **Color2Alpha** shader node (make the arrow touch to the left of the shader's name).




68. Connect the output of **Color2Alpha** into the **transparency** input of the **Phong** surface shader.

69. Choose **Nodes** > **Image_Processing** > **Intensity**. This makes the Intensity shader appear in the render tree work area. The Intensity shader increases or decreases the output/power of another shader.
70. Connect the output of the **Image** shader into the **Intensity** shader's input.
71. Connect the **Intensity** shader's output to the **specular** parameter of the **Phong** shader.

    When you're done, your render tree should look like this:




72. Double-click the **Intensity** node to open its property editor and enter a **Factor** value of 2.
73. Choose **Render** > **Region** > **Show RGB** from the Render toolbar. This renders the image over a default black background.

## Applying the Body Texture

In order to create a realistic texture, you will use a UV texture space as well as a 2D texture on the dragonfly's body. Then you'll apply a specular map and some displacement mapping to add details to the model.

Displacement mapping is done through the displacement connection of the material node.

74. Select the **User** view in viewport B.
75. Press the **s** key to track (left mouse button), dolly (middle), or orbit (right) in order to get the whole dragonfly's body in view.

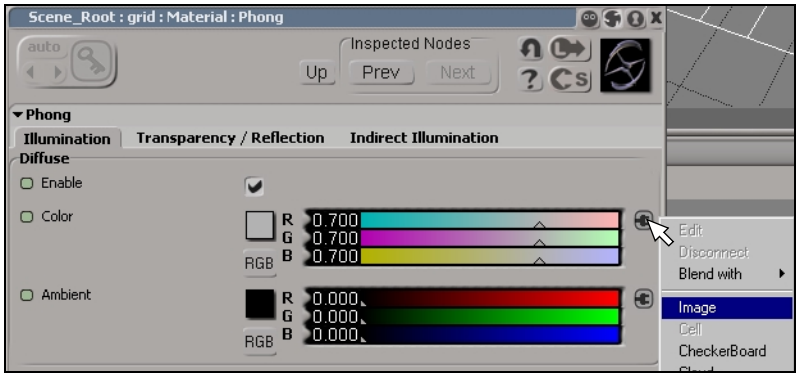
If the render region slows you down too much, you can set it to refresh on demand by choosing **Render** > **Region** > **Auto Refresh** on the Render toolbar to deactivate it. When you are ready to preview your changes, choose **Render** > **Refresh**.

### *Apply a surface shader*

Now you'll apply a surface shader to the dragonfly's body. After editing it, you'll add a texture and use that same texture's alpha channel to create a displacement map.

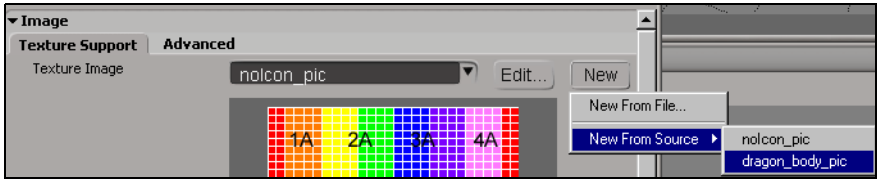
76. Select the body object from a viewport and choose **Get** > **Material** > **Phong** to apply the Phong surface shader. Enter `14` as the **Specular Decay** value and change the **Specular Color** to a light blue.

77. Still in the Phong property editor, click the connection icon to the right of the **Diffuse Color** sliders and select **Image** from the pop-up menu. This maps the **Diffuse** parameter to an **Image** shader.

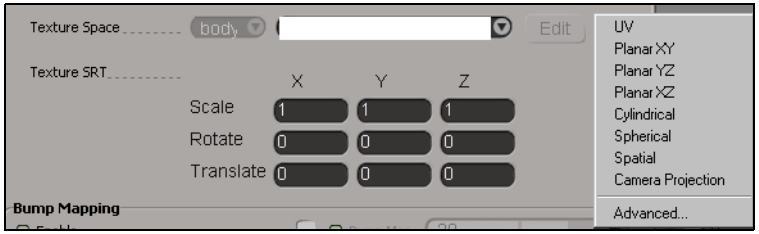



### *Map a texture using UV projection*

78. In the Image shader property editor that opens, choose **New** > **New From Source** > **dragon_body_pic** from the **Texture Image** drop-down list. The image is available as a source because you dragged and dropped it into a viewport earlier.




79. Choose **New** > **UV** from the **Texture Space** menu. This applies the dragon_body texture image using a UV texture space. Because the dragonfly's body is an irregular shape, a UV projection will follow its curves more accurately than a planar or cylindrical projection would.



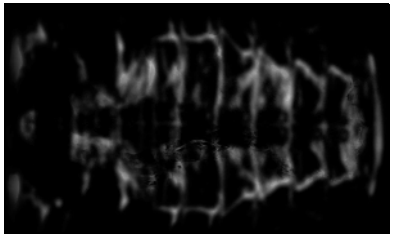

80. With the body hierarchy still selected, open a render tree (if one isn't already open) and click **Update** to see the image/surface/material connection you've just created.

81. If you have set the render region to not automatically refresh, click the **Refresh** button in the Render toolbar to see your texture map.

## Creating a Displacement Map

You will now use the alpha channel of the same texture you just applied to the body (dragon_body.pic) to create a displacement map on the dragonfly's body.

The following illustration shows the alpha channel of the body image that you will use to slightly displace the body's geometry.

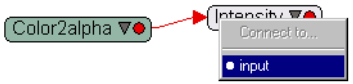


The alpha channel of the dragon_body.pic is used to drive the displacement parameter of the material node.

You may want to turn off the Auto Refresh in the render region so it doesn't update every time you make a change to your displacement value—choose **Render** > **Region** > **Auto-Refresh** from the Render toolbar.

82. From the command bar in the render tree, select the following shaders to appear in the work area:

    - **Nodes** > **Conversion** > **Color to Alpha** shader
    - **Nodes** > **Image Processing**> **Intensity** shader




Connect the Color2Alpha shader to the Intensity shader by pulling an arrow from the red dot.

Getting the alpha channel from an RGB pic file is done with the **Color2Alpha** tool shader. The **Intensity** tool shader is an image processing shader that lets you define the intensity of a color value. Use the Intensity shader to change the intensity of the alpha channel, therefore changing the intensity of the displacement itself.

83. In the render tree, pull another connection from the **Image** shader's red dot and connect it to the input of the **Color2alpha** shader. This shader extracts the alpha channel from the image.
84. Connect the output of the **Color2alpha** shader to the input of the **Intensity** shader.
85. Choose **Nodes** > **Conversion** > **Color to Scalar** tool shader.

The displacement connection entry needs a scalar value (a value between 0 and 1). You will convert the color information (a value between 0 and 255) from the Intensity shader to a scalar value with the **Color2scalar** tool shader.

If you want to see all the connection points for each node, choose **Show** > **Expand All** from the Render Tree command bar.

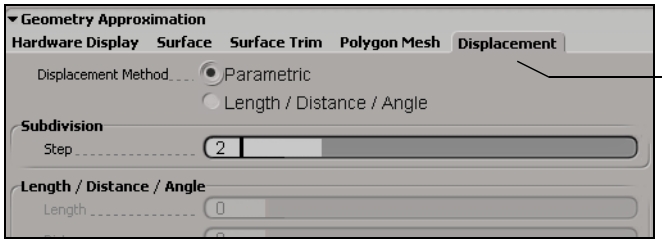
86. Connect the **Intensity** node to the **Color2Scalar** node.
87. Finally, connect the **Color2Scalar** shader to the **displacement** input of the Material node.

    The displacement map is now applied to the dragonfly's body, but you need to make some adjustments first.

88. Double-click the **Intensity** node to open its property editor. Adjust the **Factor** to 0.3 to control the amount of displacement on the dragonfly's body.


Use the Factor slider to control the amount of displacement. You could also use negative values if you wish. In this case, set **Factor** to 0.3.

89. With the body hierarchy still selected, choose **Get** > **Property** > **Geometric Approximation** to open the geometric approximation property editor for your selection.
90. Select the **Displacement** tab and set the **Subdivision Step** value to 2.




The Displacement property page allows you to set the geometric approximation as it applies to displacements.

91. Click the **Refresh** button and watch the result in the render region.

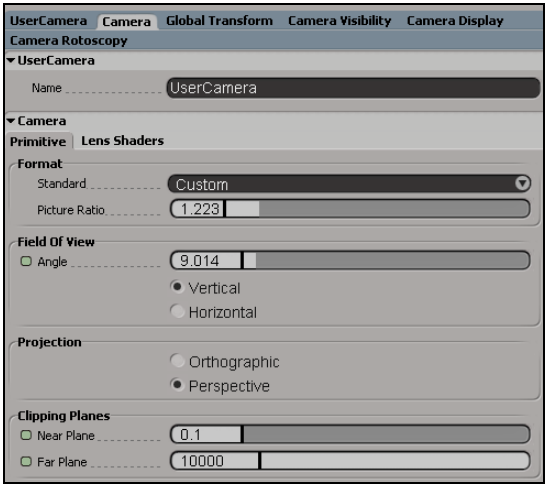
## Merging a Background Scene

The Merge command allows you to load one scene on top of another. As it happens, there is a background scene ready for you to load.

92. Choose **File** > **Merge**.

93. In the browser that appears, find and select the **dragonfly_background** scene from the tutorial_project's **Scenes** directory. The new scene loads and appears in the viewports. It's a simple background containing a corn stalk model.

You may encounter the default clipping planes of the viewports. From the viewport's command bar, select **Nav** > **Properties** to open the viewport camera's property page. From here, you can set **Near** and **Far Clipping Plane** values to determine what near and far distances your camera will "see" and render.



## Conclusion

You have taken a lifeless model and added textures with specific texture projections, created detailed textures, and added a displacement map. Finally you added a background scene to put the finished model into context.

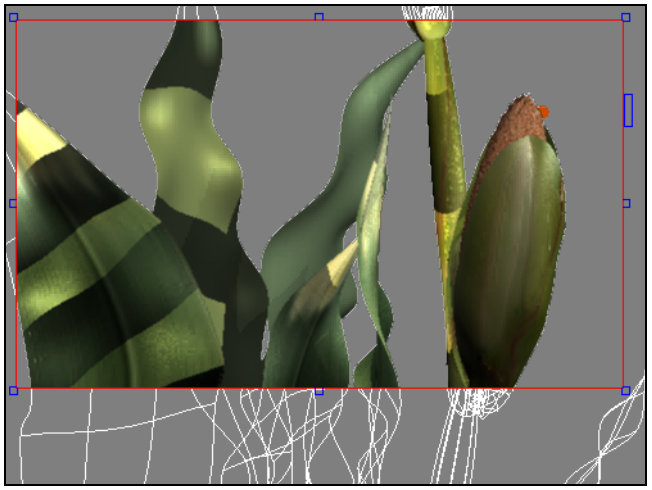
Using the Render toolbar and the render tree, you are able to "finish" a model and begin to render. The render region and the interactive property editors streamline your workflow so that there is no guesswork to your edits—you can immediately see the changes you are making.

From here, the next step is to create render passes to control how certain areas of your scene will be rendered.

For more information, see the *Shaders, Lights & Cameras* and *Rendering* guides.

# Tutorial 17: Creating Render Passes

In this tutorial, you will create a highlight pass, a depth pass, and a shadow pass for rendering. These passes automatically become a permanent part of the sequence you're working on and let you edit specific areas of your scene quickly and easily. Once defined, they will not have to be defined again.

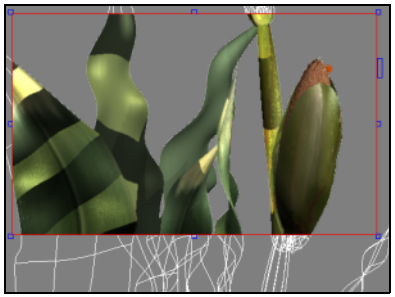


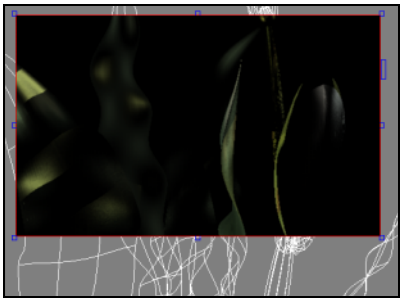
This tutorial shows you how to:

- Create a highlight pass.
- Create a depth pass.
- Create a shadow pass.
- Render all passes.

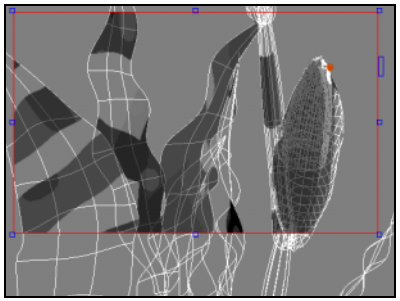
# Overview

1 **Create a highlight pass of the scene's objects**.

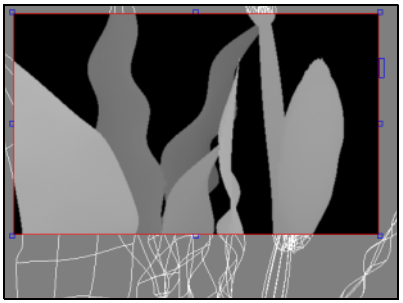




2 **Create a shadow pass**.



3 **Create a depth pass**.

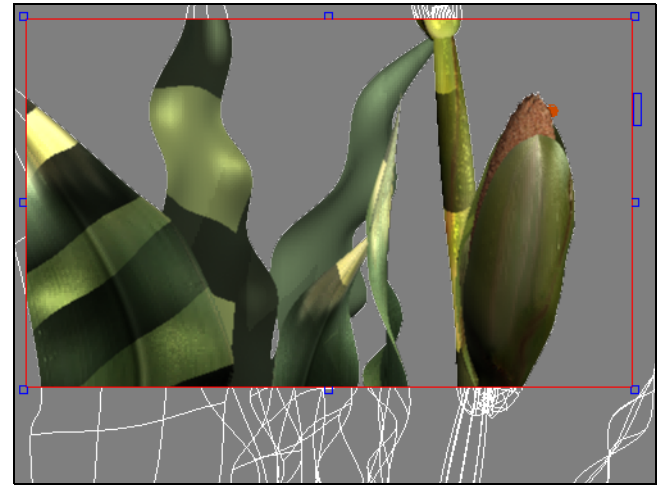


## Creating a Highlight Pass

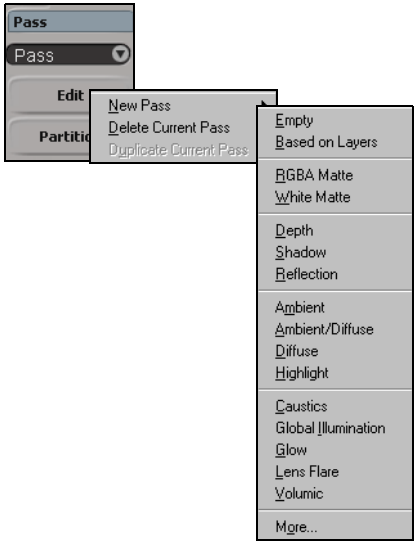
As you create new passes, the last pass defined becomes the current pass and you can immediately see what the rendered scene looks like in the render region.

The highlight pass uses all of a scene's objects and isolates their highlights. Also called a specular pass, this pass can help you tweak the brighter areas of a scene or even add shaders to the highlights while retaining the original materials of the model or object.

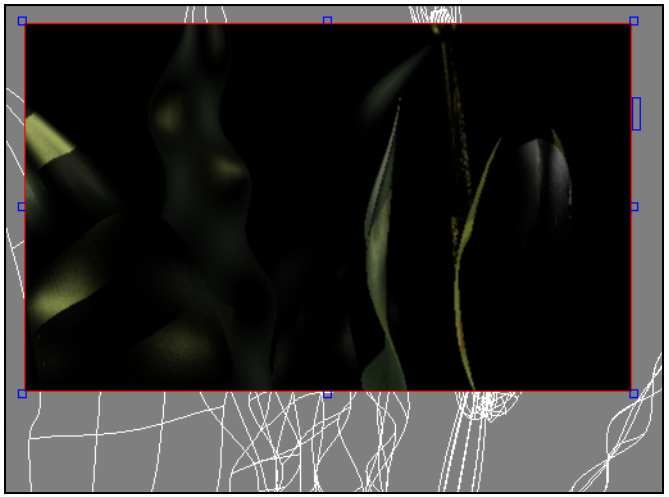
1. Choose **File** > **Open** from the main-menu bar to import the **dragon_background** scene from the tutorial database:
   **<install directory>\content\TUTORIAL_PROJECT\Scenes**



Default pass with render region set to RGB + Alpha.




2. Choose **Pass** > **Edit** > **New Pass** > **Highlights** from the Render toolbar to create the highlight pass for every object in the scene. The highlight pass is set as the current pass and the highlight pass property editor opens.
3. If you wish, enter an new **Name** for the pass in the property editor.
4. Click the Render Options tab to open the render options property page for the highlight pass. Set the options as desired and close the editor.
5. Draw a render region (press **q** and drag) in the camera viewport.
6. Set the region to show RGB only by choosing **Render** > **Region** > **Show RGB** from the Render toolbar.



Highlight pass with render region set to RGB.

Everything you create a new pass it becomes the current pass. You can specify the current pass by choosing it from the **Pass** pull-down menu on the Render toolbar.

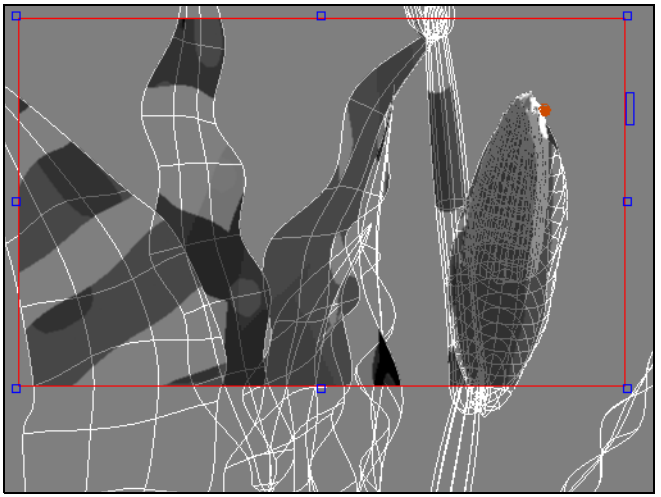
## Creating a Shadow Pass

The shadow pass lets you isolate a scene or object's shadow and edit it in a way that can compliment the scene. For example, you could apply a transparency or blur to the shadow to make it more realistic instead of a sharp, opaque shadow.

7. Choose **Pass** > **Edit** > **New Pass** > **Shadow** from the Render toolbar to create the shadow pass for every object in the scene. The shadow pass is set as the current pass and the shadow pass property editor opens.
8. If you wish, you can enter an new **Name** for the pass in the property editor.
9. Click the Render Options tab to open the render options property page for the shadow pass. Set the options as desired and close the editor.

The render region updates with the new shadow pass.

10. Choose **Render** > **Region** > **Show RGB + Alpha** in the Render toolbar to see the altered alpha channel in the render region.



Shadow pass shown with render region set to RGB + Alpha

## Creating a Depth Pass

A depth pass is used to create a depth effect for objects that are far from the camera. When composited with a rendered scene, objects that are far in the background have a grayish layer applied to them to reduce detail and simulate a type of depth fading.

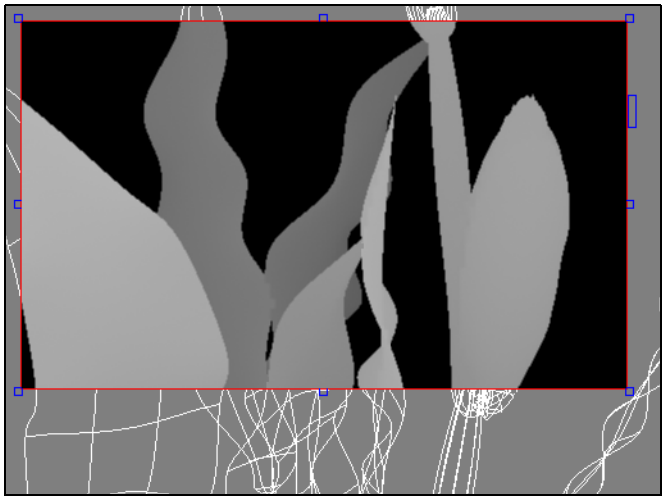
11. Choose **Pass** > **Edit** > **New Pass** > **Depth** from the Render toolbar to create a depth pass for all of the scene's objects. The depth pass is set as the current pass and the depth pass property editor opens.
12. If you wish, you can enter an new **Name** for the pass in the property editor.
13. Click the Render Options tab to open the Render Options property page for the depth pass. Set the options as desired and close the editor.

The render region updates with the new depth pass but renders black because the distance between the camera and objects is too long.

14. Open an explorer and press the **u** key to set the scope on the Current Pass.
15. Double-click the **Constant_density** shader that was defined by the default depth-pass preset.

16. Turn off **1:10** and turn on **1:100.** This sets the **Density** scale to one to a hundred. The scaling determines the depth used (in SOFTIMAGE units) to display the defined density. Use 1:100 to create a detailed depth pass of a scene that extends far from the camera, and 1:1 for a scene that is relatively flat.

17. From the Render toolbar, choose **Render** > **Region** > **Show RGB** for a clear view of what the depth pass will render.



Depth pass with render region set to RGB. Notice how objects closer to the camera are light and objects further from the camera are darker.

## Render All Passes

Once you have edited all your passes and set the render options, you can render all the passes as separate files and composite them.

18. Choose **Render** > **Render** > **All Passes.** Watch the passes being rendered one after the other without the need to re-tessellate each of them, but only at frame refresh.

## Conclusion

You have created three of the most basic render passes a scene will use. In addition, you can create and edit a matte pass, a flare pass, a diffuse and ambient pass, and so on. Each pass gives you the ability to change how certain areas, effects, or properties will be rendered and to what extent they will contribute to the final look of your scene.

For more information, see the *Shaders, Lights & Cameras* and *Rendering* guides.

# Tutorial 18: Editing Shaders in the Render Tree

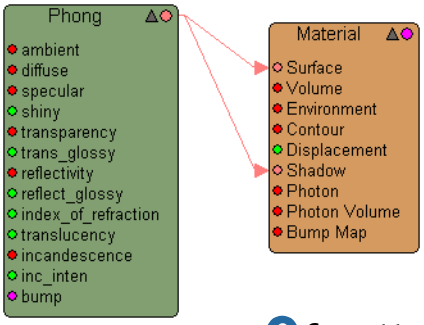
This tutorial guides you through basic shader editing in the render tree. You'll construct a glass shader using a variety of basic shaders and tool nodes found in the SOFTIMAGE|XSI shader library.

This tutorial shows you how to:

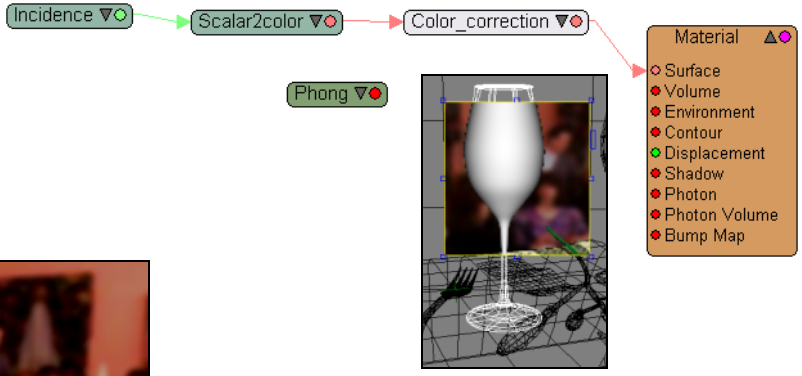
- Connect shaders to a material using the render tree.
- Use tool shaders.
- Convert values.
- Edit shader parameters.

# Overview

**1** **Apply a basic surface shader.**




**2** **Connect tool shaders to create a glass surface.**

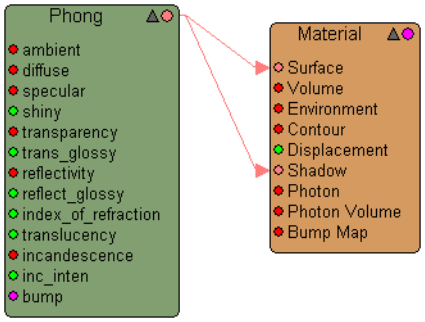



**3** **Perfect the glass surface.**

**4** **Apply a texture to both sides of a flower petal.**

## Attaching a Material Shader

1. Choose **File** > **Open** to import the **restaurant** scene from the tutorial database**:** **<install directory>\content\TUTORIAL_PROJECT\Scenes**
2. Select a wine glass and change the Front viewport (C) to a render tree view.
3. Choose **Get** > **Material** > **Phong**. You should now have a structure like this (you may have to click the Update button in the render tree):




The Phong surface shader is automatically added to the Surface and Shadow inputs of the object's Material node.

4. Draw a render region in the Camera view around the wine glass (pressing **q** and drag).
5. Double-click the **Phong** shader node to open its property editor and change these settings:
   **Illumination** > **Specular decay**: 200
   **Illumination** > **Ambient**: 0,0,0
   **Transparency/Reflection** > **Index of Refraction** (**IOR**): 1.4

To keep the refresh fast, do not change the other values right now.

## Using a Tool Shader

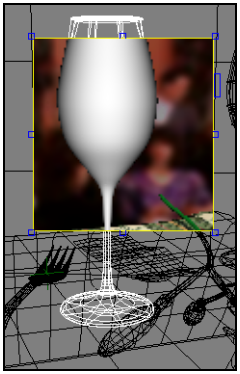
6. Add the Incidence shader to the render tree work area by choosing **Nodes** > **Illumination** > **Incidence** from the render tree command bar**.**

The **Incidence** tool shader computes the difference between an object's normal angles with the camera's point of view. You will use this information to modify the edge transparency of the glass.

*Convert a value (scalar to color)*

If you wish to connect the Incidence shader's output (a scalar value) to the Surface input of the Material node, you must first convert it to a color value.

7. Choose **Nodes** > **Conversion** > **Scalar to Color** from the command bar to add the **Scalar2Color** converter shader to the render tree workspace.



8. Connect the **Incidence** shader's output into the input of the **Scalar2Color** shader.
9. Select the **Scalar2Color** shader and press the **p** key to preview it. The Preview tool temporarily connects the selected node to the surface input of your object so you can see the values you edit.

When previewing, notice the change in the render region. You can see that the surface changes to black when the angle between the normal and the camera's point of view is high and shifts to a white/gray when there is less difference between the two angles.

Don't forget to press **p** again when you're finished previewing. Staying in Preview mode updates the render region with every selection.

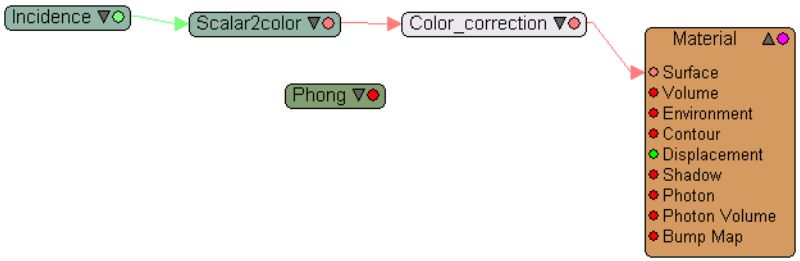
10. Open the **Incidence** tool shader by double-clicking its node.
11. In the **Incidence** property editor, select **Camera Direction** as the **Incidence Mode** and enter `1.2` as the **Exponent Value**.

To continue seeing the surface effects while using the Preview tool, lock the **Incidence** property page and select the **Color2Scalar** conversion shader so that its output is visible when connected to the surface of the Material node.

## Fine-tuning an Effect

Now adjust the edge intensity of the glass by using the ever-useful Color-correction tool shader.

12. From the **Nodes** menu, select the **Image Processing** > **Color_correction** shader. The red dot from which connections are "dragged" signifies that this shader's output is a color value.
13. Connect the output of the **Scalar2color** shader into the input of the **Color_correction** shader.
14. Select the **Color_correction** shader and press **p** to preview. Your render tree should look like this (while in Preview mode):




15. Double-click on the **Color_correction** shader node and input the following values in its property editor:

    **Gamma**: 1.7
    **Contrast**: 0.9
    **Hue**: 0
    **Saturation**: 0
    **Level**: 0.01

16. From the **Nodes** menu, select the **Image Processing** > **Invert** shader so it appears in the render tree work area.

17. Connect the output of **Color_correction** to the **Invert** shader, and the output of the **Invert** shader to the **Phong** surface shader's **diffuse** input.

### *Apply a finish*

18. Connect the output of **Color_correction** to the **transparency** input of the **Phong** surface shader.

19. Connect the output of the **Phong** shader to the **Surface** input of the Material node. Your glass effect is now visible.

If your glass is not transparent, check the default ray depth of the render region. A setting of 1 will not show any transparency. Select **View** > **Setup** > **Properties** and click on the **Optimization** tab. Change the **Ray Depth** values to:

Reflection: 4
Refraction: 4
Maximum Ray Depth: 4

20. To accentuate the realism and glass imperfections, add some glossy reflections and transparency frostiness. Double-click the **Phong** shader and play with the following parameters:

    **Transparency > Frost**
    **Transparency > Samples**
    **Transparency > Index of Refraction**
    **Reflection > Color**
    **Reflection > Gloss**
    **Reflection > Samples**

## Grouping Objects to Apply a Material

You will now modify the surface and texture of the flower petals as a group and apply a shared material to the group. The flower petals are made from a single surface and therefore inherit the same texture on both sides of the surface. You'll apply different textures on the interior and exterior of the petal using the Front/Back tool shader.

21. Select all six open petals of the main flower (use Shift-select to add to a selection or Ctrl-select to subtract).

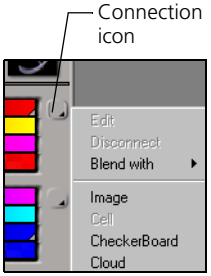
22. Click the **Group** button from Edit panel of the the main command area and enter `Petal_group` as the group's name. Close the Group property editor.

23. Set the explorer so that it is focused on the **Scene** and **Groups**. Make sure the Petal_group is selected. Selected groups appear as a light-gray wireframe in the viewports.

24. Draw a render region across the petals in the camera view.

25. Choose **Get > Material > Phong** from the Render toolbar. Keep the default **Phong** surface shader's property page open for now (you can lock it open if you wish).

26. Reselect the Petal_group and open a render tree in one of the viewports. Click **Update** in the render tree to display the group material node and its assigned (Phong) surface shader.




### *Load the Front/Back tool shader*

27. From the render tree command bar, choose **Nodes > Switch > Front/Back** tool shader to place it in the render tree work area.

28. Connect the **Front/Back** shader's output to the **diffuse** input of the Phong shader.

29. Double-click on the **Front/Back** node to open its property editor. Click on the **Front** connection icon and select the **Image** shader.

30. On the Image shader property page, select **New** > **New From File** to define an image clip. Navigate to the SOFTIMAGE|XSI content database and select the **hemerocalis_branche.pic** image.
31. From the **Texture Space** drop-down menu, click **New** and select a UV texture space.
32. Open the **Front/Back** shader property page again (click **Prev.** at the top of the present property page to cycle through previously opened property pages).
33. Click on the connection icon and repeat step 30 to select the hemerocalis.pic image.
34. Select **UV** as the **Texture Space** in the drop-down menu of the Image property page.
35. Orbit in the viewport (press **o**) to admire both sides of the petals!

36. You may now stop admiring and press Ctrl+n to clean the workspace, or Shift+Delete to delete all.

## Conclusion

You have now used the render tree to achieve a highly customized affect by using specific shaders and shader tools to control a texture's parameters. Using the render tree, fine-tuning texture effects can be done at the shader level with unprecedented control and detail.

# Index

## R



## S



## T

## U



## V



## W



## Z