SOFTIMAGE®

SOFTIMAGE®|XSI™

Version 1.0

Animating

Avid®

Animating was written by Judy Bayne, and Grahame Fuller; edited by Edna Kruger and John Woolfrey; and formatted by Luc Langevin.

Printed in Canada.

Document No. 0130-04614-01 0400

Contents

Section I • Animation Basics

Section III • Animation Mixing

Section IV • Simulation

Roadmap

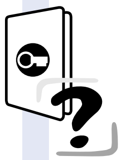
About This Guide

Animating contains information on creating the illusion of movement.

- ***Chapter 1: Basics of Animation***—the different methods used to create and play back animation.
- ***Chapter 2: Animating with Keys***—creating and editing keyframes, the fundamental building blocks of animation. Also covers the animation editor, where you visually edit animation represented by function curves.
- ***Chapter 3: Animating along Paths and Trajectories***—how to make scene elements follow drawn paths.
- ***Chapter 4: Animating with Constraints***—about onstraining objects, including lights and cameras, to other objects.
- ***Chapter 5: Animating with Expressions***—how to use mathematical formulas to animate parameters.
- ***Chapter 6: Linked Parameters***—how to link two parameters so that moving one moves the other in a manner determined by a function curve.
- ***Chapter 7: Custom Parameters***—how to define custom parameters so that you can control elements using sliders in a custom property editor.
- ***Chapter 8: Skeletons & Inverse Kinematics***—how to create and animate skeletons made up of chains and objects.
- ***Chapter 9: Envelopes***—how to assign an object as an envelope so that it becomes a “skin” that deforms as its skeleton moves.
- ***Chapter 10: The Animation Mixer***—animation-mixer basics and clips in general: scaling, trimming, extrapolating, mixing, and so on.
- ***Chapter 11: Actions***—how to store animation as actions, then use the animation mixer to reuse the actions over and over.
- ***Chapter 12: Shape Animation***—describes the various techniques for animating the shapes of objects, as well as blending shapes using the animation mixer.
- ***Chapter 13: Audio***—importing audio files when matching animation to sound.
- ***Chapter 14: Particles***—how to create a particle simulation by defining the appearance of particles as well as their behavior as they decay and interact with natural forces and obstacles.
- ***Chapter 15: Soft Body***—describes the concept of soft body simulation, which mimics the deformation of a malleable object as it collides with obstacles or is acted upon by natural forces.

Where to Find Information

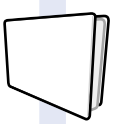
The SOFTIMAGE|XSI package includes a comprehensive set of learning materials. Use this Roadmap to find the information you need to get up and running quickly and effectively.



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 I • Animation Basics

Chapter 1 Basics of Animation

To animate means to breathe life into something; life, in turn, is always signified by change: growth, movement, dynamism. In SOFTIMAGE®|XSI™, everything in a scene is represented by numeric values, and as such animation is the process of changing these values—position, color, or any other property—over time. For example, you can make a cat leap on a chair, a light grow dim, a camera pan across a scene, a material change color, or fog evaporate.

The animation tools let you create animation quickly so you can spend the time you need to perfect it by editing the movement, changing the timing, and trying out different tools as often as you need to. SOFTIMAGE|XSI gives you the control and quick feedback you need to produce great animation.

Like SOFTIMAGE®|3D and other modeling applications with animation capabilities, SOFTIMAGE|XSI's most basic method of animation is keyframing. You set property values, or *keys*, at specific frames, called *keyframes*, and values for the frames in between are calculated by interpolation. This process is described in *Chapter 2: Animating with Keys* on page 45.

Creating two or more keyframes results in the creation of a function curve that you can then edit graphically with the animation editor. For more information on editing function curves, refer to *Editing Function-Curve Animation Using the Animation Editor* on page 62. Any animation can be represented by a function curve, linking time with values over a graph. Editing a function curve gives you an enormous amount of control over your animation.

You can also animate an object's properties by using constraints, linked parameters, or expressions. As a general rule, using these methods involves using one object's animation to drive another's.

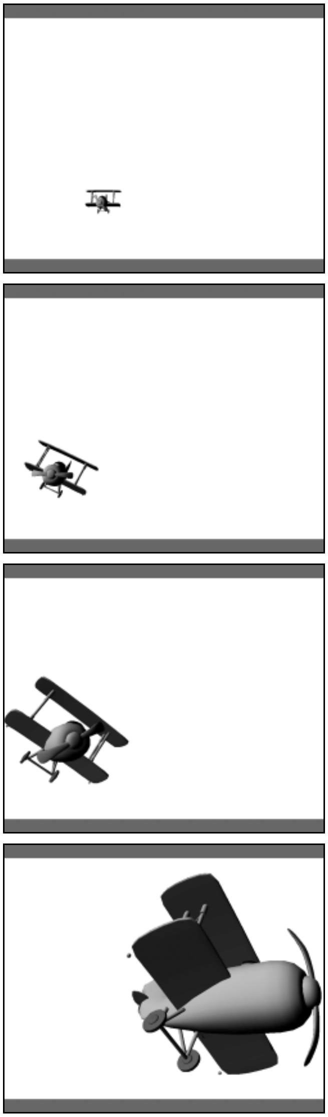
Using constraints, you can make an object move, rotate, or change size based on another object's transforms; or you can use a path constraint to make an object follow a path defined by a curve. For more information on constraints, see *Animating along Paths and Trajectories* on page 91 and *Animating with Constraints* on page 107.

With linked parameters, you aren't limited to transforms; you can create a relation between any two animatable parameters. For instance, you could cause a sphere to change its color based on its radius. For more information on linked parameters, see *Chapter 6: Linked Parameters* on page 153.

In some ways, expressions can be considered a more advanced version of constraints and linked parameters. Expressions are mathematical formulas that drive an object's animation based on other objects, time, and other parameters. Expressions allow for more complex relationships to be built between an animated parameter and other elements of a scene. For more information on expressions, see *Chapter 5: Animating with Expressions* on page 133.

You can preview your animation in a viewport by using the Playback panel or by generating a flipbook (see *The Playback Panel* on page 36).

What You Can Animate



You can animate scene elements and most of their parameters—in effect, if a parameter exists on a property page, it can probably be animated.

Animatable parameters fall into four categories:

- **Motion:** Probably the most common form of animation, this involves displacing an object from one point to another, or rotating it.
- **Geometry:** You can animate an object's structure by changing values such as U and V subdivision, radius, length, or scale. You can also use surface deformations and skeletons to bend, twist, and contort your object.
- **Appearance:** Material, textures, visibility, and transparency are just some of the parameters controlling appearance that can be changed over time.
- **Behavior:** Constraints, rotation limits, and bounding volumes are used to limit or control an object's animation. These all employ parameters whose values can be changed over time; for instance, changing the size of a bounding volume can gradually restrict an object's movement.

You cannot animate global-rendering options.

Animation Concepts

Layers of Animation

One of the most important features of SOFTIMAGE|XSI is its layered approach to animation. Each layer represents a different level of control, from editing a single parameter's value to copying animation from one character to another. This allows you to easily manage complex animation yet retain the ability to work at the most granular level.

It is important to understand the hierarchical nature of the control layers. Each layer provides tools that are compatible with and make use of any layer below it. Tools at any level internally use tools from any of the lowest levels but not from above. For example, you can use the animation mixer to blend actions together in a transparent manner, whether the action contains function curves, constraints, or expressions.

Actions

An action is any type of animation that can be stored. Actions are stored per object and can then be reused in different ways. Once you have created some actions, you can mix them as you like in the animation mixer.

Mixing

You can create an animation sequence by linking actions over time. You can also mix actions together, weighting the actions as you like. Mixing can be done with various transition effects, from the simplest (for example, weighted averages) to the more sophisticated (such as timewarps).

Connection and Mapping Templates

Animation information is stored per model for the purposes of easily editing animation and sharing animation between characters.

Any type of animation that you generate can be stored as an action for an object or on disk. Actions can be reused later and applied to the same or a different model using filters and connection and mapping templates. Actions can also be combined together. For example, you can create a new action for a model based on the lower-body motion from one action and the upper-body motion from another.

Time

Time is relative in SOFTIMAGE|XSI.

You can modify the time reference of a value or the animation of a parameter. This includes creating time offsets, time scaling or warps for non-linear time, and cycling animation.

You can change the time directly on any object: to make a character move twice as quickly, simply add a time-control property to it.

You can also change the duration of an action indirectly in the animation mixer when you bring it onto a track.

Useful Tools for Animation

A number of tool groups in the main window are designed specifically to help you accomplish your animation tasks. This chapter briefly introduces them.

The Animate Toolbar

You'll find the Animate toolbar at the far left of the screen.

Get commands
Create generic elements, including primitive objects, human models, cameras, and lights. (Also available on Model and Render toobars)

Create commands
Define expressions, animation paths, and skeletons.

Deform commands
Create and edit deformations.

Actions commands
Save and apply actions.

Simulate commands
Simulate particle clouds, gravity, wind, magnetism, and other behavior.

Tools commands
Plot animated movement.

To display the Animate toolbar
Click on the toolbar title then choose **Animate**.

or

Press 1 from the number row of your keyboard.

The Animation Panel

The Animation panel, at the bottom right of the main window, contains a group of animation controls that let you quickly access a number of important animation options:

Animation button
Accesses a menu of commonly used animation commands that set keys, displays the animation editor, copies, pastes, removes animation, and more.

Autokey button
Automatically sets a keyframe after each parameter edit.

Mark Parameter icon
Displays the parameters of a selected element for marking.

Clear Marked Parameter Display icon
Clears the marking from all parameters.

Next/Previous-Key icons
Moves to the next/previous keyframe associated with the marked parameter.

Keyframe icon
Adds a key at the current frame on the timeline.

Marked Parameter display box
Displays the currently marked parameter.

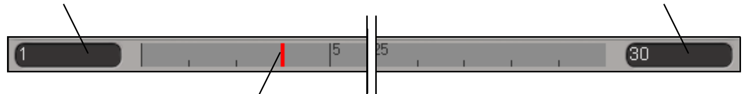
Lock Marked Parameter button
Locks the current marked parameters.

The Timeline

The timeline is used to display the current playback position of the animation and to manually move between different frames. The current frame is indicated by the position of the playback cursor in the timeline display box. The current position of the animation can also be controlled using the Playback panel.

Start Frame box: Sets the first frame of the timeline.

End Frame box: Sets the last frame of the timeline.



The playback cursor indicates which frame is currently displayed in the viewports. Drag the cursor to a specific frame, or use the left- and right-arrow keys to move back or forward through frames one at a time.

Dragging the pointer with your middle mouse button will move you to the selected frame, but the scene will not update in the viewports until you release the button. This lets you drag quickly to the required frame while avoiding lengthy refresh time.

The Playback Panel

This portion of the main window lets you control the playback of your animated scenes.

Play/Play Backward icons Plays/stops the animation.

Audio icon Enables/disables sound during playback.

Playback button Displays a menu of playback options.

Time display box Displays/sets the current frame.

Update All button Lets you specify what in the scene will update during playback.

Increment icons Moves the currently displayed frame forward/backward by pre-defined increments (default is 1).

Start Frame/End Frame icons Resets the displayed frame to the beginning/ end of the timeline.

Loop icon Repeats the animation in a continuous loop.

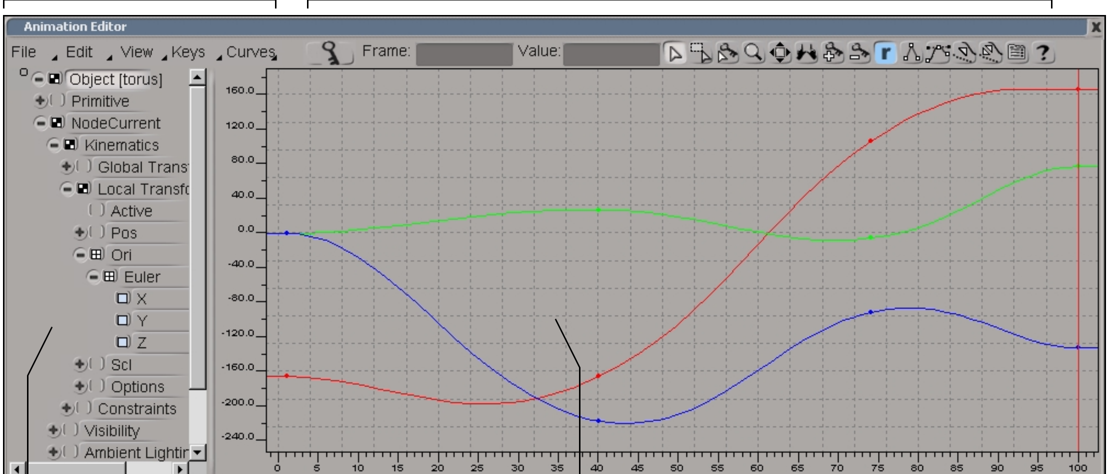
All/RT button Lets you choose between playing back frame by frame (All) or in real time (RT).

The Animation Editor

The animation editor lets you control the animation of a selected element by means of function curves. It contains tools that you use to manipulate the function curves of the selected element's properties. For more information on how to display and use the animation editor, see *The Animation Editor* on page 63.

Menu bar Contains commands that let you save, edit, and view the function curves of animated parameters.

Toolbar Provides fast access to frequently used options, including viewing, selection, and curve-interpolation commands.



Property tree Displays the animatable parameters of elements that have been selected in the viewports or explorer.

Graph editor Displays and permits manipulation of parameter animation by means of function curves.

The Animation Mixer

The animation mixer gives you high-level control over actions. Using the mixer, you can manipulate and combine animated actions. For more information on how to mix actions, see *Chapter 11: Actions* on page 253.

The overview shows the portion of the timeline visible in the track area. Drag the edges to resize, or drag the center to move.

Shows the tracks of the selected object’s model.

Global time is the time of your scene. Local time is the time inside a compound clip.

The current frame is shown by a red line on both the timeline and overview.

The Start and End frames control the scope of the overview.

Clears the tracks from the display.

Navigates to parent tracks.

Advanced functions for modifying actions sources and clips.

Icons let you magnify, select, ripple, mute, and solo tracks.

Clips appear as colored bars on tracks. You can create sequences of actions on the same track.

Multiple tracks let you overlap clips in time and mix their weights.

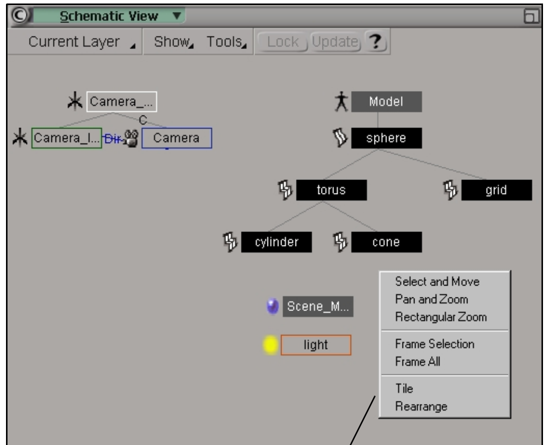
Mix clips by setting and animating weight values

The Schematic View

The Schematic view allows you to view the relations between the objects in your scene, such as the hierarchy of objects and how objects constrain one another. For more information on the Schematic view, see *The Schematic View* in Chapter 2 of the *Fundamentals* guide.

Use the menu bar to access display, selection, and navigation commands.

Use the display area to view and edit scene elements and their properties.



Right-click in an empty area to quickly access a number of viewing and navigation commands.

Setting Preferences

Before you begin animating, you can set animation preferences that let you automatically set keyframes for animation, choose the type of movement that is calculated between the keyframes, set the time display, and define the default interpolation method.

To set animation preferences, open the User Preferences dialog box by choosing **File** > **User Preferences** from the main-menu bar.

Setting the Time Display Format

The Time Display property page in the User Preferences dialog box lets you specify how time is displayed in the animation editor timeline. By default, time is represented on the timeline as 29.97 frames per second.

To set the Time Display format

1. In the User Preferences dialog box, select the **Time Display** tab to display the Time Display settings.
2. In the Time Display Format box, select **Use** and choose the frame rate to be represented on the timeline in the animation editor.

   Select **Display as Frames** if you want the frame rate you selected in step 2 to be represented in frames rather than as timecode on the timeline.

Regardless of whether you type timecode or a frame number in a text box, SOFTIMAGE|XSI internally converts your entry into timecode.

Activating Autokey Mode

If you frequently animate objects and properties, you can choose to automatically set keyframes as you adjust the object's properties. Instead of clicking the keyframe icon in a property editor each time you want to add a key, you can let the autokey mode automatically record the adjustments that you make.

This has the same effect as activating the auto button in the Animation panel. The difference between the two is that the second method sets autokey mode for the current session only and is not saved to your preferences file.

For more information on keyframing, see *Chapter 2: Animating with Keys*.

To activate autokey mode

1. In the User Preferences dialog box, select the **Animation** tab to display the Animation property page.
2. Select the Autokey option.

   When you save this setting, the auto button in the lower-right corner of the Animation panel is activated.

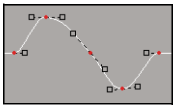
Setting the Insertion Type

You can determine how movement will be calculated in a function curve between keyframes.

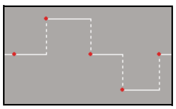
To set the insertion type

1. In the User Preferences dialog box, select the **Animation** tab to display the Animation property page.

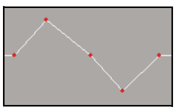
2. Select one of the Default Interpolation for Inserted Keys options:



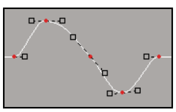
- Select **Automatic (From Neighboring Keys)** to ensure that the slopes for the keys are automatically calculated from the neighboring keys. This is the default option, which uses splines to ease in and ease out of each key.



- Select **Constant** to repeat the value of a key until the next keyframe. The resulting movement is characterized by static positions, followed by sudden changes.



- Select **Linear** to set regular intervals between keyframes, making each segment of a function curve appear as a straight line. The resulting movement is characterized by constant speed and sudden changes of movement at each key point.



- Select **Spline** to set an acceleration (ease-out) and a deceleration (ease-in) immediately before and after each keyframe. This results in a smooth transition of motion at each keyframe.

Viewing Animation

There are different ways of playing back animation, but the most common ways are to use the timeline and the Playback panel.

You will also want to preview your animation scene in a render region prior to the final render. You can get real-time playback of animation in the render region by using its caching capabilities. See *Creating a Flipbook from Cached Images* on page 39.

Timing and Frame Rates

In animation, the smallest unit of time is the time required to display a single frame. The speed at which frames are displayed, or the *frame rate*, is always determined by how the final animation will be viewed. For example, in North America, television plays at 30 frames per second (or fps), while in Europe it plays at 25 fps. Film, on the other hand, plays at 24 fps. Most traditional animation is shot "on twos," which means each individual image is held for two frames. So, practically speaking, film animation shot on twos runs at 12 fps, though the film itself plays at 24 fps.

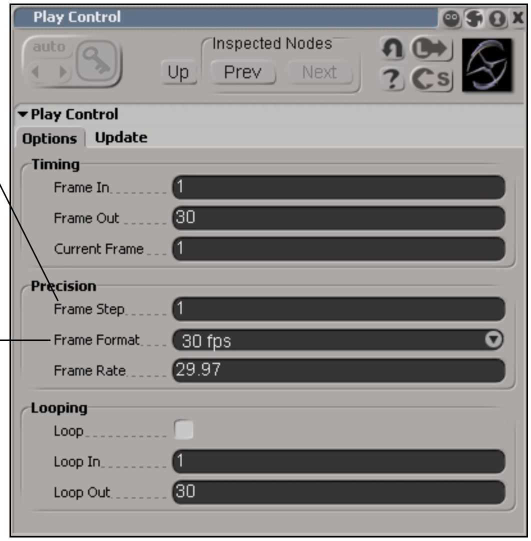
If you are compositing your animation with other film or video footage, it's usually best for the animation to be at the same frame rate as the footage.

When you change the timing of the animation, you change the way that the actions look. This means that the timing that looked correct while you were previewing it in SOFTIMAGE|XSI may not look as good on video or film. For example, an action that spans 24 frames would take one second on film; changing the frame rate to suit North American video at 30 fps would cause the same 24 frames to span 0.8 seconds. As a result, you should make sure to preview your animation at the same frame rate that will be used in the final format.

To set the frame rate, choose **Playback** > **Playback Options** to open the Play Control property editor and click the **Options** tab.

Frame Step determines how many frames should be skipped when playing back your animation. For example, a Frame Step of 2 would play every other frame. Frame Step does not work when using real-time playback.

The Frame Format should be set to match the intended final destination of your animation. For example, you would use 24 fps for film, 29.97 fps i for NTSC video, and so on. If the frame rate you want isn't listed, enter it in the Frame Rate box.



The Playback Panel

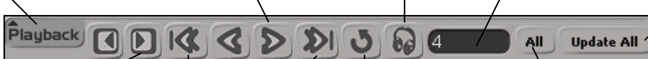
The Playback panel at the bottom of the main window allows you to view animation in a variety of ways:

Play/Play Backward icons Plays/stops the animation.

Audio icon Enables/disables sound during playback.

Playback button Displays a menu of playback options.

Time display box Displays/sets the current frame.



Update All button Lets you specify what in the scene will update during playback.

Increment icons Moves the currently displayed frame forward/backward by pre-defined increments (default is 1).

Start Frame/End Frame icons Resets the displayed frame to the beginning/ end of the timeline.

Loop icon Repeats the animation in a continuous loop.

All/RT button Lets you choose between playing back frame by frame (All) or in real time (RT).

- To stop playback at any time, click any mouse button anywhere in the viewports or timeline or click the Play icon.
- When Loop is activated, two vertical bars appear in the timeline, at the first and last frames. Ctrl+drag to move these bars anywhere in the timeline; they indicate the start and end frames of the loop.
- To change the playback frame rate, choose **Playback** > **Playback Options** to open the Play Control property editor and click the **Options** tab. In the Frame Format drop-down list, choose the required frame rate format.

By default, the Frame Step controls in the Play Control property page are set to 1, meaning that each frame in the scene is played back. If you wish to view every third frame during playback, set this value to three. If you wish to view every fifth frame, set this value to 5, and so on.

- You can have SOFTIMAGE|XSI execute a script whenever a frame is displayed by choosing **Playback** > **Playback Options** and clicking the **Options** tab. Enter the script name in the Command box.

Setting the Time Format

The time display box displays the current frame number or time position in the scene. As you move the playback cursor, the time display box displays the frame at which the cursor is located. Choose **File** > **User Preferences** > **Time Display** from the main-menu bar to set the timeline to display in seconds, milliseconds, or frames. By default, the timeline is set to display as frames.

Optimizing Animation Playback in a Viewport

The more you can see of the animation's essentials, the better the visual feedback you have to work with. Ultimately, this helps you produce better animation.

You may not always be able to play back each frame at the correct rate, however. At every frame, SOFTIMAGE|XSI must check if positions, orientation, scaling, geometry, visibility, and many other animatable parameters have changed. If they have changed, SOFTIMAGE|XSI must then calculate the extent of the change, calculate the effects of the change, and then redraw the new image. The more calculations to be done, the longer it will take to display each frame.

If playback is too slow, you must decide what aspect of the animation is critical for display and eliminate the extraneous elements that may be limiting the playback speed. For example, if you're just verifying that your phantom's coloring is shimmering properly, you can display the shifting color but not the phantom's translation. Since there is less to calculate, the scene will play back faster—especially if the phantom's animation was very complex.

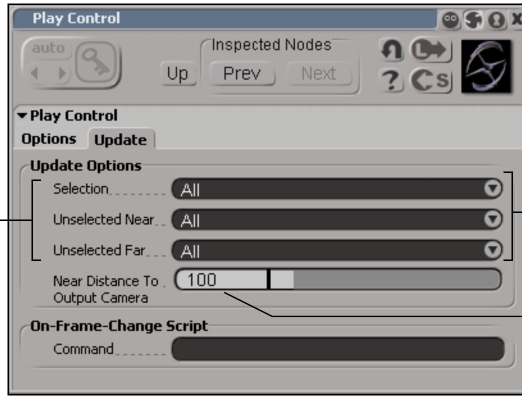
Some factors that slow down playback are inevitable. For instance, the more objects in a scene, the slower the playback. However, most causes can be easily remedied, such as:

- Other tasks running on your computer.
- The number of active viewports. Muting non-essential viewports speeds up playback in the remaining viewports. For more information, see *Muting and Soloing Viewports* in Chapter 4 in the *Fundamentals* guide.
- The method of displaying models. Using options such as textures and shaded models significantly slows playback. For more information, see *Display Types* in the *Fundamentals* guide.

You can also control which animatable parameters will be updated during playback using the **Update** tab on the Play Control property editor. To open the Update page, choose **Playback** > **Playback Options** to open the Play Control property editor and click the **Update** tab.

By specifying update parameters for selected objects and near or far unselected objects, you can visualize exactly what you need while eliminating unnecessary animation.

You can decide what kind of animation should be displayed for these objects: all, none, SRT, geometry, other, or combinations thereof.



Any objects that fall within this distance from the camera (in SOFTIMAGE units) are considered near. All other objects are considered far.

Real-Time Playback

SOFTIMAGE|XSI can accomplish real-time playback by choosing the frames to display based on the elapsed time. For instance, if it takes half a second to display the first frame in a 24 fps animation, frames 2 through 12 will be skipped and frame 13 displayed next. In other words, when it comes time to display a frame, the frame that should be shown at that moment is displayed.

To use real-time playback, choose **Playback** > **Real-Time Playback** . To see all the frames in your animation (though not necessarily in real-time), choose **Playback** > **Play All Frames.**

To see the rate at which your animation is played back, hold down **Ctrl+r** and click in a viewport. The displayed rate will be updated during playback.

Previewing Animation as a Flipbook

The only way to guarantee that all frames will be displayed at the correct frame rate is to cache the images and create a flipbook. You can cache the images of a rotoscoped scene and background or cache the frames of an animation you play back in a render region. Unlike animation playback from the Playback panel, a flipbook will always play back at the defined frame rate, provided you have enough memory for the cached images.

Creating a Flipbook from Cached Images

A flipbook is any scene of cached images saved to disk. These cached images can be loaded and quickly played back in real time in any of the viewports or in a floating window. Flipbooks can be exported in a variety of standard formats, such as AVI and QuickTime.

To create a flipbook, you cache each frame of an animation that you played back in the render region.

To record a set of cached images

1. Load a scene with animation.
2. In the viewport whose images you want to capture, set the display options as desired. For instance, you may want to set the display mode to Hidden Line Removal for a “pencil test” effect, or use the quick render in part of the viewport.
3. Set the start and end frames on the timeline to the start and end frames you want to capture.
4. Choose **Nav** > **Start Capture** from the viewport menu. The flipbook window appears, playing back the frames as they are cached. During subsequent playbacks, you can pause, stop, or loop the playback using the controls in the Flipbook toolbar.

   By default, the animation plays in a continuous loop in real time. You can set how cached images are played back by choosing **File** > **Cache Properties** in the Flipbook toolbar. The Cache Properties property editor appears, allowing you to set playback options.

   The first time the images are played back, the frame rate will be slow because SOFTIMAGE|XSI is capturing the images. Subsequent replays will be in real-time.

5. Click **Stop** in the flipbook window’s title bar when you want to end the cache playback.
6. To save your cached images for later use, choose **File** > **Save** in the flipbook window’s title bar to save the images in the format and location of your choosing.

Loading a Flipbook

You can load image scenes and movie files as flipbooks in any of the viewports or as a floating window. To open a floating flipbook window, choose **View** > **Views** > **Flipbook** from the main-menu bar. To switch to the Flipbook view in a viewport, click on a viewport's Views menu and choose the Flipbook view.

You can load a flipbook at any time while working—it does not matter if you have a scene currently loaded. However, if you load a new scene, the flipbook will be removed from the viewport.

Each time you display cached images in the viewport, the images that appear will be those that were most recently cached.

1. Open a floating flipbook window or a Flipbook view in the viewport.
2. Choose **File** > **Load** from the Flipbook menu.
3. Specify the path and file name you want to load. Click OK.

   You can also load several flipbooks by dragging and dropping them into a flipbook window from a Windows Explorer window. You can then select a flipbook from the Sources menu.

4. Choose the **Play** button from the viewport title bar to play the animation. You can also pause, stop, and loop the playback from the title bar.

Changing Flipbook Playback Options

By default, a flipbook plays back the original cached or loaded scene in exactly its original size, frame rate, and so on. However, you may want to change some of these settings for more efficient playback, to see different timings, or to highlight parts of a scene.

Flipbook settings are modified in the Cache Properties property editor. Choose **File** > **Cache Properties** from the Flipbook menu to open the Cache Properties property editor.

Changing the Flipbook's Appearance

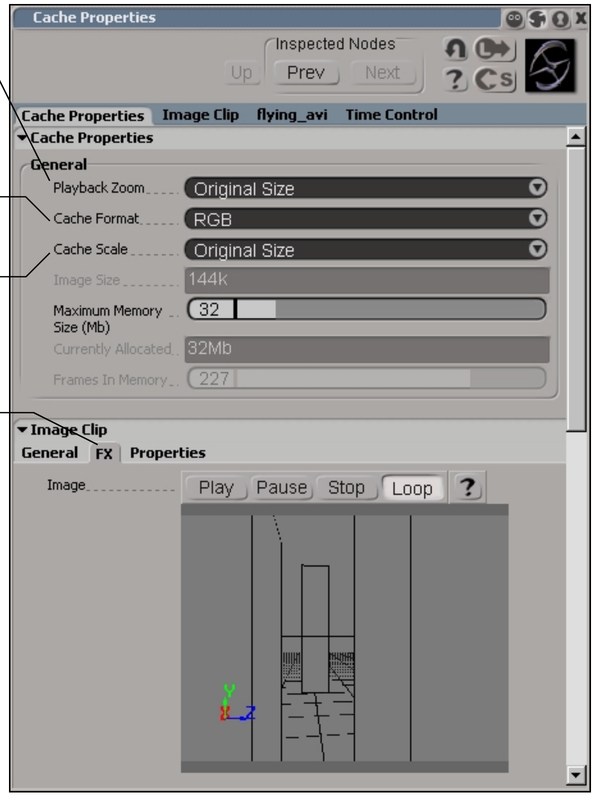
You can change your flipbook's size or color or apply effects such as blurring, cropping, and color correction.

Playback Zoom controls the scale of the flipbook with respect to the size of the cached images.

Cache Format controls the flipbook's color depth.

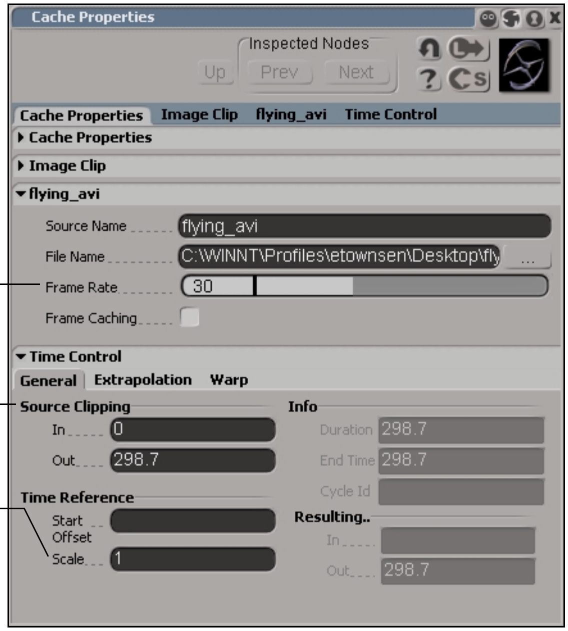
Cache Scale controls the scale of the cached images with respect to the original images (the viewport or the loaded video file).

Controls for color correction, blurring, cropping, and more can be found on the **Image Clip** property page under the **FX** tab.



Changing the Flipbook's Timing

You can change your flipbook's playback speed and its start and end frames.



Frame Rate sets the speed of the flipbook's playback.

Source Clipping sets the flipbook's start and end frames.

Scale sets the speed of the playback with respect to the frame rate. For instance, a Speed of 2 plays a flipbook at twice the frame rate.

Rotoscoping

Rotoscoping is a technique in which video or film images are placed in the background of a scene, one frame at a time. You can use these reference images to create your own animation by tracing objects from the images or matching your objects with the images' motion, or to match your camera's point of view with the images'. You can zoom and pan the scene while maintaining perfect registration with the imported background.

You can perform rotoscoping in any geometry view using any display type. Furthermore, you can use different rotoscope images for each viewport.

To display rotoscoped images

1. From any viewport, choose **Display Type** > **Rotoscopy Options** to open the Rotoscope Options property editor.
2. In the **Image Name** text box, specify the path and file name of the image or movie clip you want to load as a background. Click the browser button (**...**) to browse for files on your system.
3. Choose **Display Type** > **Rotoscope** to rotoscope your scene over the background. You can modify the rotoscope settings from the Rotoscope Options property editor. Refer to Online Help for a description of each option in the editor.

You can use the play, loop, forward, and backward arrows in the Playback panel to advance the sequence of your rotoscoped background along with any animated objects in your scene.

At any point in time you can cache the contents of the render region and create a flipbook that you can play back in the viewport. For more information on this, see *Creating a Flipbook from Cached Images* on page 39.

Zooming in a Rotoscoped View

When zooming in a rotoscoped view, you must use the rectangular zoom option (Shift+z supra key) to define the zoom area, which magnifies the selected rectangular area to fill the screen. The camera position does not change. Once you have done a rectangle zoom, the mouse buttons are configured as follows:

- Left-click or middle-click to zoom in.
- Right-click to zoom out.

You can then use the zoom option (z supra key) to pan, zoom in, and zoom out on the defined area. When you pan, it has the effect of moving the snapshot in front of a still camera.

Rotoscoping in the Render Region

You can also view your rotoscoped scene and background in a render region.

1. Choose **View** > **Render** > **Region** > **Region Tool** from the Render toolbar.
2. Draw a render region in the viewport containing the rotoscoped scene and background (see *Previewing Rendered Scenes* on page 77 of the *Fundamentals* guide).

   The region renders the scene composited over the background.

Compositing the Alpha Channel

The render region also allows you to composite the alpha channel of objects in your scene over the rotoscoped background. To do this:

1. Choose **View** > **Render** > **Region** > **Show Alpha** from the Render toolbar to activate the alpha channel of the scene in the render region. This is useful if you want to quickly check the accuracy of a particularly complex alpha channel.
2. Choose **View** > **Render** > **Region** > **Show Alpha + RGB** to use the alpha channel of your scene and composite over the background. This allows you to view motion blur and check the registration of shadows composited directly over the imported background. These shadows must have alpha channels defined, such as those created by a shadow shader.

Chapter 2 Animating with Keys

Traditional hand-drawn animation is generally created using keyframes—an animator draws the extreme or critical poses at the appropriate frames, creating “snapshots” of movement at specific moments. The frames in between the keyframes are filled in afterward. This filling-in process is called *in-betweening*.

Keyframes are the cornerstone of convincing animation. The drawings themselves determine what a figure looks like at specific moments in time, the frames chosen for the keyframes determine the timing and weight of a figure, and the number of keyframes can reflect the complexity of the movement.

In SOFTIMAGE|XSI, any form of movement (for example, translation or rotation) is a matter of manipulating values. Keyframing is a process for animating those values over time. As in traditional animation, a keyframe is a “snapshot” of one or more values at a given frame. Unlike traditional animation, SOFTIMAGE|XSI handles the in-betweening for you, computing the intermediate values by interpolation. Just about anything that has a value can be animated; this includes scaling, colors, textures, lighting, visibility, and even an object’s geometry.

Before you begin to keyframe, analyze the effect you intend to create: which are the critical frames you should set values for, and which frames can you let SOFTIMAGE|XSI calculate? The critical frames and their values determine the keys you should set.

For example, suppose you want a light to grow dim for 50 frames then grow brighter for the next 50. Frame 1 sets the initial light intensity, frame 50 sets the lowest intensity, and frame 100 sets the final intensity. Values at frames 1, 50, and 100 should be keyframed; the other values will be calculated by SOFTIMAGE|XSI.

Setting Keyframes

There are a number of ways to keyframe values.

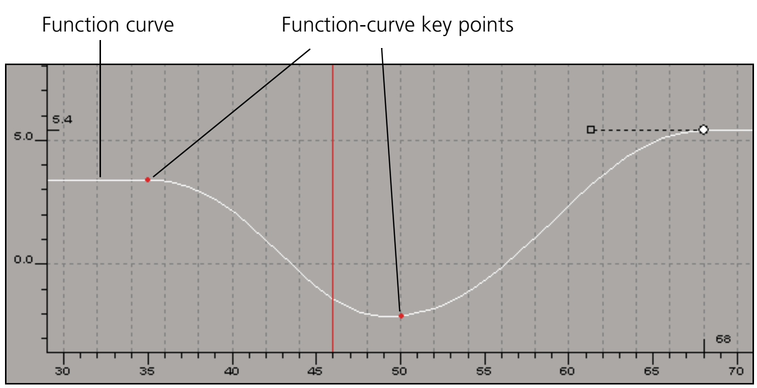
You can:

- Set keyframes for parameters using the keyframe icon in the Animation panel.
- Set keyframes for parameters from a property editor.
- Set keyframes for parameters from the explorer.
- Automatically set keyframes for individual parameters as you modify their values in autokey mode.

You can also keyframe transformations using a combination of supra keys or 3D manipulators together with either of the above methods. For cameras and spotlights, you can also keyframe by dollying and orbiting in a viewport in autokey mode.

You can set keyframes in any order and at any time. Saving a key over an existing one replaces the old key. Whenever you add a new key, SOFTIMAGE|XSI recalculates the interpolation between the previous and next keys.

Each time you set a key, a function curve, or *fcurve*, is created. A function curve is a graph that represents the changes in a parameter's value over time. Function curves are displayed in the animation editor, which contains a powerful set of tools that let you edit function curves and thereby fine-tune the animation.



For more information on editing function curves, see *Editing Function-Curve Animation Using the Animation Editor* on page 62.

Marking Parameters for Keyframing

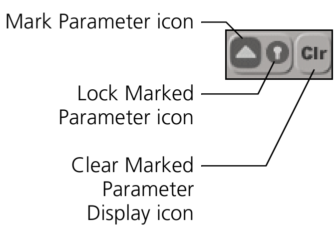
Marking parameters allows you to select which parameters will be keyframed for the currently selected object or objects when using the keyframe icon or the **k** supra key. You can mark parameters from the marked parameter list, a property editor, or the explorer.

Properties remain marked even if the selection changes. For example, if you were to mark properties on a sphere then select a cube, the same properties would be marked on the sphere. Any marked properties that did not apply to the cube (such as Radius) would be ignored.

To mark a parameter from a property editor

1. Select the element whose parameter you want to mark.
2. Open a property editor and display the first property page with the parameters you want to mark for keyframing.
3. Click on a parameter name. The parameter name is highlighted, indicating that it has been marked for keyframing.

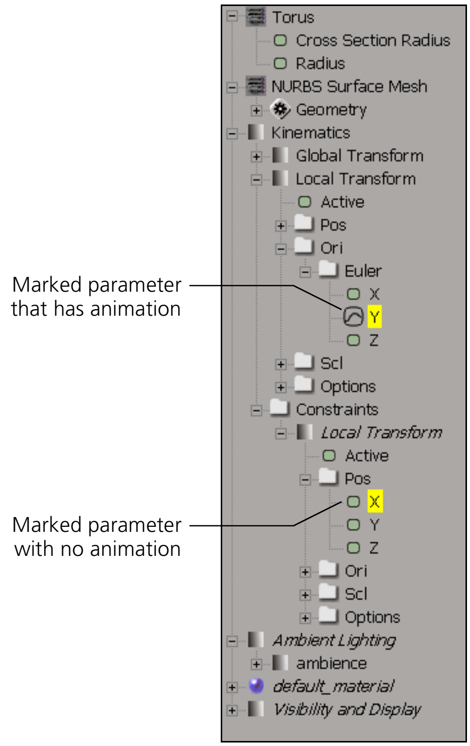
   Click the parameter name a second time to unmark it. Use Ctrl+click add or remove other marked parameters.



To mark a parameter from the marked-parameter list

1. Select the element whose parameter you want to mark.
2. Click the Mark Parameter icon on the Animation panel to open the property explorer.
3. Expand a property node until you display the property or parameter you want to mark.

   Animatable parameters are indicated by a green box to the left of their names if no animation has yet been assigned to them. Once a parameter is keyframed, a larger box with a small curve appears.
4. Click on the property or parameter name to mark it. The parameter name is highlighted, indicating that it has been marked for keyframing. Click the name a second time to unmark the property. Shift+click to mark more than one node at a time, and Ctrl+click to toggle a node's marking.



5. Click anywhere outside the property explorer to close it.

Locking and Clearing Marked Parameters

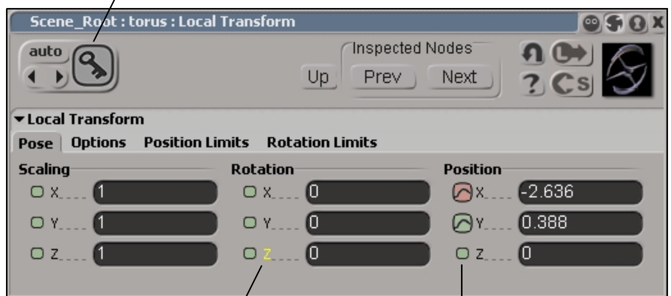
Activating any of the tools from the Transform panel automatically marks the appropriate scale, rotate, or translate parameters. This also has the effect of removing any existing marking from other parameters. To ensure that your marked parameters are unaffected by using the transformation tools, click the Lock Marked Parameter icon on the Animation panel. You can unlock the parameters by clicking the Lock Marked Parameter icon again.

To deselect all marked parameters, click the Clear Marked Parameter icon. Locked parameters cannot be cleared.

If you hold down the supra key while using a transform tool, your existing marked parameters will only temporarily be unmarked, even if you haven't locked them. After you release the supra key, they will be marked again.

Keyframing with Property Editors

Every property editor lets you set keyframes for any or all of its animatable parameters by using the parameter-specific animation icons or the keyframe icon at the top of the property editor. You can also predetermine individual properties to be keyframed by marking them.



Either the keyframe icon or the animation icons can be used to set keyframes in a property editor.

Clicking the keyframe icon sets or removes keyframes for all marked parameters in the property editor. If no parameters are marked, keyframes will be set or removed for all animatable parameters in the property editor. Clicking a parameter's animation icon sets or removes keyframes for the parameter whether it is marked or not.

The keyframe icon's appearance changes, depending on the status of the property editor's marked parameters.

- If no marked parameters have been animated, the keyframe icon is gray.
- If any animated marked parameters on the current property editor have modified values but are not yet keyframed, the keyframe icon is yellow.
- If a keyframe is defined for any marked parameter on the current property editor page, the keyframe icon is red.
- If a marked parameter is animated but the current frame is not a keyframe, the keyframe icon is green.

If a property editor's marked parameters have different animation icons, the keyframe icon's color is based on priority: it is red only if there are no yellow animation icons, and green only if there are no yellow or red animation icons.

Similarly, the animation icon's appearance changes, depending on the status of its parameter.

- If the animation icon is a small green box, the parameter is not animated.
- If the animation icon is a red box with a curve, the parameter is keyframed at the current frame.
- If the animation icon is a green box with a curve, the parameter is animated but the current frame is not a keyframe.
- If the animation icon is a yellow box with a curve, the parameter is animated but the has been changed from its original value and has not yet been set as a keyframe. If you change the current frame without keyframing the parameter's new value, the parameter will revert to its interpolated value.

To keyframe all animatable parameters in a property editor

1. If you have not already done so, set the animation scope. Choose **File** > **User Preferences** > **Animation** from the main-menu bar and check **Apply to All Parameters When Nothing Is Marked**.
2. Move the playback cursor to a frame where you want a key and open the property editor.
3. Display a property page with parameters you want to animate.
4. On the property page, set the parameters to the values you want.

5. Click the keyframe icon in the property editor.

   If you want to keyframe parameters in other property pages or property editors at the current frame, display them and repeat steps 3 and 4.
6. Move the playback cursor to another frame where you want a key.
7. Set parameters and click the keyframe icon again.
8. Repeat steps 6 and 7 until you have added all the keyframes necessary to create your animation.

To keyframe all marked parameters in a property editor

1. Move the playback cursor to a frame where you want a key and open the property editor.
2. Display a property page with parameters you want to animate.
3. On the property page, set the marked parameters to the values you want.

4. Repeat steps 2 and 3 for each property page in the property editor.

5. Click the keyframe icon in the property editor.

   If you want to keyframe parameters in other property editors at the current frame, display them and repeat steps 2 to 4.

6. Move the playback cursor to another frame where you want a key.
7. Set parameters and click the keyframe icon again.
8. Repeat steps 6 and 7 until you have added all the keyframes necessary to create your animation.

To keyframe specific parameters in a property editor using the animation icons

1. Open the property editor and, if it has animation-icon controls, display the first property page with the parameters you want to animate.
2. Move the playback cursor to a frame where you want a key.
3. Set the value for a parameter you want to animate and click the animation icon to the left of the parameter name.
4. Repeat step 3 for all the parameters you want to animate in the property editor.

   If you want to keyframe parameters in other property editors at the current frame, display them and repeat steps 2 and 3.

5. Move the playback cursor to another frame where you want a key.
6. Repeat steps 3 to 6 until you have added all the keyframes necessary to create your animation.

You can right-click on an animation icon to access a menu with animation controls that let you set and remove keyframes as well as copy, paste, and remove animation. You can also access this menu by right-clicking on the animation icon in the scene explorer (animatable parameters must be visible in the explorer to do this).

When keyframing a color value, the alpha channel will also be keyed, even if no alpha channel information is visible in the property page.

Setting Keyframes Automatically

You can have SOFTIMAGE|XSI automatically set a keyframe each time you move to a new frame and modify the values of specific parameters.

To set keyframes automatically

1. Move the playback cursor to a frame where you want a keyframe (in other words, where you want the animation to start).
2. Open the property editor and set the properties to the values you want.
3. Click the **auto** button in the property editor or the Animation panel.
4. Move the playback cursor to a frame where you want the next keyframe, and change the animated properties again. Every time you change a property value, it is keyframed.

   If you make a mistake, click the Undo icon (left). You can also remove unwanted keyframes as described in *Removing Animation from Properties* on page 61.
5. Repeat step 4 as often as you need.
6. When you have finished, turn autokey mode off by clicking the autokey button again.

Be careful when dollying, orbiting, or framing in viewports while autokey mode is on. If you are looking through a camera or spotlight viewpoint, the camera or light is moved and keyframed. For more information, see *Autokeying a Camera's Viewpoint* on page 54 and *Autokeying a Spotlight's Viewpoint* on page 55.

To prevent accidentally autokeying the wrong parameters, use the Lock Marked Parameter icon on the Animation panel to ensure that keyframing will only affect the selected parameters.

Autokeying a Camera's Viewpoint

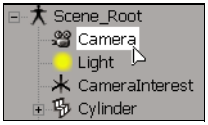
In addition to using any of the previous methods to manually keyframe camera parameters, you can keyframe camera parameters interactively while looking through it in a viewport.

When you look through a camera (choose a camera view in the viewport) and then change the viewpoint by dollying, orbiting, or framing, one of two things can happen:

- If autokey mode is off, the viewport changes the camera's viewpoint, but this change is not saved. If you move to the next or previous frame in your animation, the camera's viewpoint reverts back to the position determined by its existing animation. Viewpoints let you see your project from any angle and are not related to any camera. This prevents you from accidentally moving the camera.

- If autokey mode is on, the camera itself moves as the viewpoint changes. In addition, a keyframe is created for the camera at the current frame.

To animate a camera while looking through it



1. Select a camera from a scene explorer.
2. Click the translate button in the Transform panel, then turn autokey mode on by clicking the **auto** button in the Animation panel.
3. Use the keyboard and mouse to change the point of view. You can:
   - Orbit using the **o** key. Orbiting affects the translation and rotation of the camera.
   - Dolly using the **p** key. Dollying affects only the position of the camera.
   - Frame selected objects using the **f** key.
   - Frame all objects using Shift+f.

   For more information about using keys to change the point of view, see *Changing the Viewpoint* on page 83.
4. To animate the camera, change the current frame using the playback cursor and perform step 3 again. Repeat as often as necessary.
5. When you have finished, turn autokey mode off by clicking the **auto** button again.

If you don't turn autokey mode off and later change the point of view in a camera or spotlight view, the camera or light is affected and the new settings are keyframed. If autokey mode is off, the viewpoint is affected, not the camera.

Autokeying a Spotlight's Viewpoint

In addition to using any of the previously described methods for keyframing light parameters, you can keyframe spotlights interactively while looking through them in a viewport.

When you look through a spotlight and then change the viewpoint by panning, orbiting, and dollying, one of two things can happen:

- If autokey mode is off, the viewport changes only the viewpoint and the light is not affected. Viewpoints let you see your project from any angle and are not related to any spotlight. This prevents you from accidentally moving the light.
- If autokey mode is on, the spotlight itself moves as the viewpoint changes. In addition, a keyframe is created for the light at the current frame.

To animate a spotlight while looking through it

1. Click on a viewport's Views menu.
2. Choose **Spot Lights** and then choose the light you want to move. The viewport display changes so that you are looking through the spotlight.

You can look "through" a spotlight toward its interest by selecting it from the Spot Lights menu. This lets you see where a light is pointing in one viewport while you manipulate the light in another.

3. Click the translate button in the Transform panel, then turn autokey mode on by clicking the **auto** button in the Animation panel.
4. Use the keyboard and mouse to change the point of view. You can:
   - Pan (track) and zoom using the **z** key.
   - Perform a rectangular zoom using Shift+z.
   - Orbit using the **o** key.
   - Dolly using the **p** key.

   For more information about using keys to change the point of view, see *Viewport Views* in Chapter 4 of the *Fundamentals* guide.
5. To animate the spotlight, change the current frame using the timeline and perform step 4 again. Repeat as often as necessary.
6. When you have finished, turn autokey mode off by clicking the **auto** button again.

If you don't turn autokey off and later change the point of view in a spotlight view, the light is affected and the new settings are keyframed. If autokey is off, only the viewpoint is affected, not the spotlight.

Keyframing Transformations

You can use the SRT transformation controls as well as supra keys or 3D manipulators to keyframe the scaling, rotation, and translation of scene elements, including cameras and lights. You can create keyframes manually, or, when autokey mode is on, automatically.

Camera rotation in the X and Y axes cannot be keyframed because the camera is constrained to point at the camera interest. You can animate a camera's Z rotation by either deactivating its up-vector constraint or modifying its Roll setting. Both parameters are available by making Properties visible in an explorer (**Show** > **Properties** on the explorer menu) then selecting the camera and choosing **Kinematics** > **Direction Cns** > **Up Vector**.

For more information on up-vector constraints, see *Up-Vector Constraints* on page 128.

To keyframe transformation

1. Select one or more scene elements.
2. Move the playback cursor to a frame where you want a keyframe.
3. Use supra keys, 3D manipulators, or the Kinematics property editor to transform the selected elements in local or global space (keyframes are set in local space by default). For more information on transforming elements, see the *Chapter 6: Working in 3D Space* chapter in the *Fundamentals* guide.

keyframe icon

4. Click the keyframe icon on the Animation panel or use the **k** supra key.
5. Move the playback cursor to another frame where you want a keyframe.
6. Transform the elements and click the keyframe icon again.
7. Repeat steps 5 and 6 until you have added all the keyframes necessary for the animation.

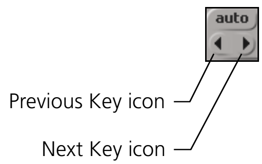
To keyframe transformations using autokey

1. Select one or more scene elements and display their property editors.
2. Click the **auto** button in the property editor or the Animation panel.
3. Use supra keys or 3D manipulators to transform the selected elements. For more information on transforming elements, see the *Chapter 6: Working in 3D Space* in the *Fundamentals* guide.
4. Move the playback cursor to another frame where you want a keyframe.
5. Transform the elements again.
6. Repeat steps 4 and 5 until you have added all the keyframes necessary for the animation.
7. When you have finished, turn autokey mode off by clicking the **auto** button again.

Remember, selecting any transformation tool automatically marks the corresponding scale, rotate, and translate parameter. Therefore, using a transformation tool after clicking the autokey button sets a keyframe for the transformation.

Keyframing with the transformation controls without explicitly marking parameters always generates three function curves for X, Y, and Z, even if only one axis has been selected for transformation.

Moving between Keyframes



There are several ways to move between keyframes defined for any property on a currently displayed property editor:

- To move to the next keyframe defined along the timeline, click the Next Key icon in the animation controls in the upper-left corner of the property editor.
- To move to the previous keyframe, click the Previous Key icon in the animation controls in the upper-left corner of the property editor.
- To move to the first keyframe, right-click the keyframe icon and choose **First Key** from the pop-up menu.
- To move to the last keyframe, right-click the keyframe icon and choose **Last Key** from the pop-up menu.

The keyframe icon displays visual cues to help you navigate among keyframes:

- If a keyframe is animated for any marked property but the current frame is not a keyframe, the keyframe icon turns green.
- If a keyframe is defined for any marked property, the keyframe icon turns red.

Copying Keyframes

Copying Keyframe Values to a Different Frame

If your animation requires repetitive keyframes where a parameter regularly returns to the same value, you can quickly copy keyframe values to several frames.

To copy a parameter's keyframe values

1. With the animatable values set at the current frame, use the right mouse button to drag the playback cursor to its new position on the timeline.

   The playback cursor is green, indicating that it is waiting for the keyframe to be set.

2. Set the keyframe by clicking the keyframe icon or choosing **Animation** > **Set Key** from the Animation panel.

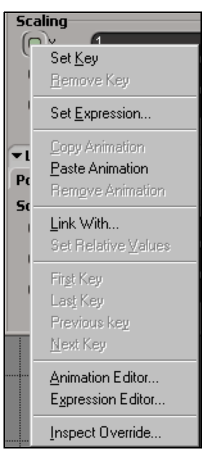
   Once the keyframe is set, the red playback cursor will jump to the same position as the green playback cursor and a duplicate keyframe will be created in the new position.

3. Repeat steps 1 and 2 for any other duplicate frames you would like to create.

This technique can also be used if you've made the mistake of changing the animated values before moving the playback cursor to a new keyframe position. Copy the keyframe values using the above method, then delete the original keyframe.

Copying Keyframes between Parameters

Keyframe values and interpolation information—a function curve—can be duplicated between different parameters for the same object or for different objects. For example, you could copy the X rotation values from one object to the Y scaling of another object.



To copy animated parameters in a property page

1. Open the property page containing the animated parameter you want to copy.
2. Right-click on the parameter's animation icon and choose **Copy Animation** from the pop-up menu.
3. If necessary, open the property page for the parameter to which you want to copy the function curve.
4. Right-click on the parameter's animation icon and choose **Paste Animation** from the pop-up menu.

To copy animated parameters from an explorer window

1. Open an explorer window. If necessary, choose **Show** > **All Nodes** from the explorer menu to display animatable properties.
2. Expand the nodes until the animated parameter you want to copy is displayed.
3. Right-click on the parameter's animation icon and choose **Copy Animation** from the pop-up menu.
4. Expand the nodes until the animated parameter you want to copy to is displayed.
5. Right-click on the parameter's animation icon and choose **Paste Animation** from the pop-up menu.

You can also copy animation in the scene explorer by dragging one animated parameter node on top of another parameter.

Removing Animation from Properties

Animation can be removed from:

- Marked parameters.
- The current keyframe for all parameters in a property page. SOFTIMAGE|XSI automatically recalculates the interpolation between the remaining keyframes.
- All keyframes for all parameters in a property page.

Alternatively, you can use the animation editor to delete parameter keyframes on the function curve (see *Deleting Key Points* on page 77).

To remove a single keyframe from a parameter on a property page

1. Open the property editor and the specific page from which you want to remove a keyframe.
2. Move the mouse pointer to the keyframe you want to remove.
3. Click the animation icon, or right-click the animation icon and choose **Remove Key** from the pop-up menu.

To remove all keys from a frame for parameters in a property editor

1. Display the property editor from which you want to remove animation.
1. Click the keyframe icon until it is no longer red.

To remove all keyframes for all parameters in a property editor

1. Display the property editor from which you want to remove animation.
2. Right-click the keyframe icon and choose **Remove Animation** from the pop-up menu.

To remove all keyframes from marked parameters

From the Animation panel, choose **Animation** > **Remove Animation**.

This will also remove animation from any parameters immediately below the marked parameters in the hierarchy. For example, if an object's Local Transform parameter was marked, the Pos, Ori, and Scl parameters immediately below it would have their animated removed.

To remove all animation from an object

1. Select the object from which you want to remove animation.
2. From the Animation panel, choose **Animation** > **Remove All Animation**.

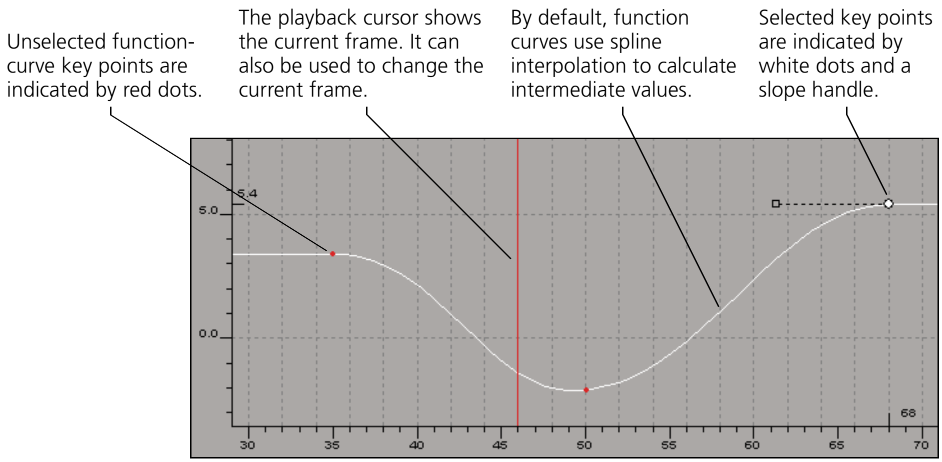
This command removes all keyframe animation and expressions from the object's parameters.

Editing Function-Curve Animation Using the Animation Editor

A function curve, or *fcurve*, is a graph that represents the animation of a parameter. When you edit a function curve, you change the animation.

Time is shown plotted along the X axis (horizontal) of the animation editor, and the parameter's value is plotted along the Y axis (vertical). The shape of the curve shows how the parameter's value changes over time.

You define the value of a parameter at a particular point in time by setting keyframes. SOFTIMAGE|XSI calculates values between keyframes by interpolation. On the function curve, keyframes are represented by *key points*, and the interpolation is represented by segments linking the key points. Key points and their associated slope handles define the function curve in the same way that control points define Bézier curves.



After you have defined animation by keyframing, you can display the resulting function curve in the animation editor. You can use the animation editor to modify the function curve or to copy the function curve to another property.

Overview of Editing Function Curves

The following steps briefly outline how to use the animation editor to set, modify, and fine-tune your animations:

1. Animate object properties, lights, or cameras by setting keyframes as described in *Setting Keyframes* on page 48.
2. Display the animation editor as described in the next section.
3. Select the objects whose animation you want to edit. Their animated parameters are displayed in the animation editor.
4. Select which function curves you want to display and work with. You can also zoom, pan, and frame function curves. These procedures are described in *Viewing Function Curves* on page 68.

5. Select and manipulate key points and function curves. You can move, add, delete, cut, copy, and paste, as well as control the interpolation. For procedures, see *Selecting Function Curves* on page 71 and *Editing Function-Curve Animation Using the Animation Editor* on page 62.

6. As you work, play back your animation in a viewport. You can also create a flipbook to play back a fully rendered animation in real time.

Undoing and Redoing Function-Curve Modifications

To undo any modifications to a function curve, press Ctrl+z.

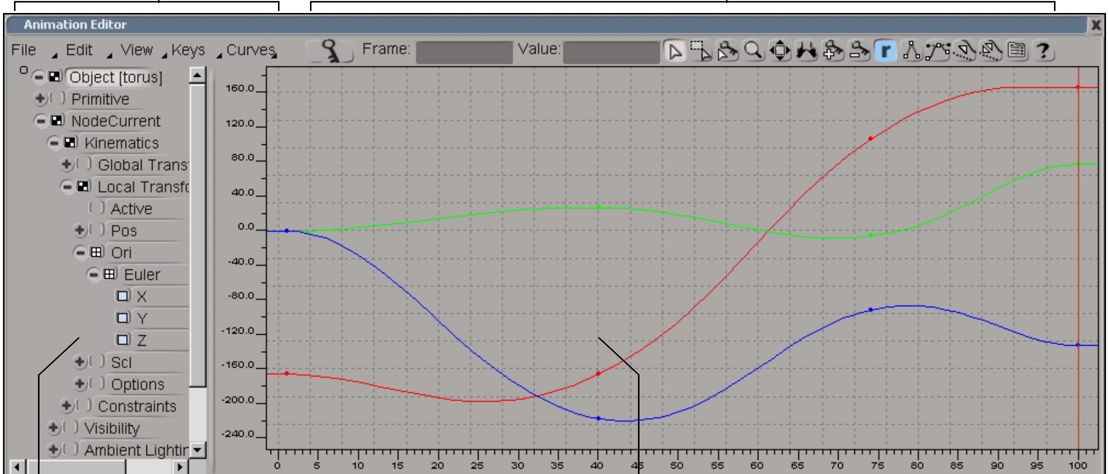
To redo the modification you just undid, press Ctrl+y.

The Animation Editor

The animation editor lets you control the animation of the currently selected element. It contains tools to manipulate the function curves of the selected element's parameters. The animation editor is divided into four sections: the property tree, the graph editor, the menu bar and the toolbar.

Menu bar
Contains commands that let you save, edit, and view the function curves of animated properties.

Toolbar
Provides fast access to frequently used options, including viewing, selection, and curve interpolation commands.

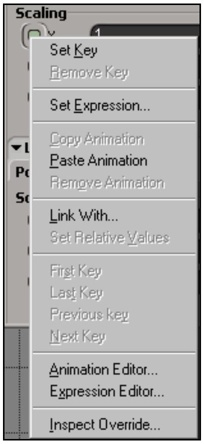


Property tree
Displays the animatable properties of elements that have been selected in the viewports or explorer.

Graph editor
Displays and permits manipulation of property animation by means of function curves. Curves for X, Y, and Z parameters are red, green, and blue, respectively.

Displaying the Animation Editor

There are several ways to display the animation editor:



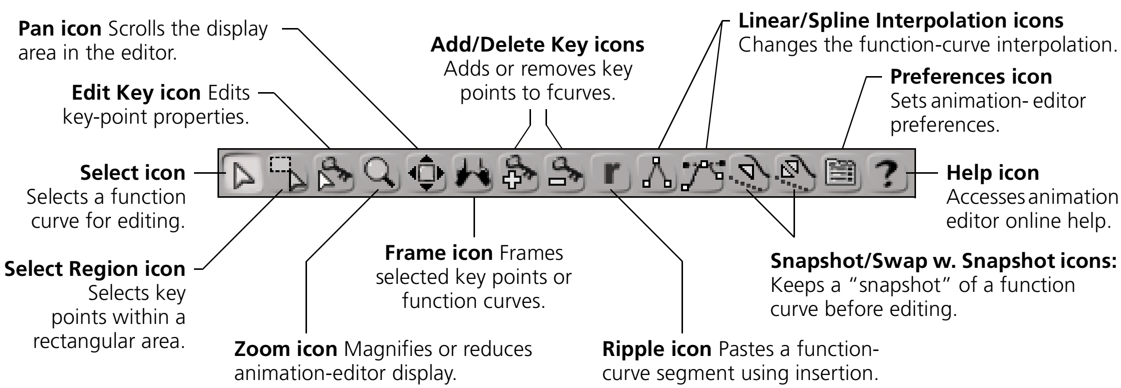
- Choose the **Animation Editor** view from a viewport title bar. Depending on which viewport you made your selection in, the animation editor is anchored across the two upper or lower viewports.
- In a property editor with animation icons, right-click on an icon and choose **Animation Editor**. The animation editor is displayed in its own floating window, which you can move and resize like any other window.
- From the main-menu bar, choose **View** > **Views** > **Animation Editor**. Again, the animation editor is displayed in its own floating window.

The animation editor responds to the currently selected elements. While it is open, it displays the animation (if any) of the elements you select.

The Animation Editor Menu Bar

The animation-editor menu bar contains all the commands, tools, and options required to edit and view a selected element's function curve. It also contains commands that let you import, export, and save animation files.

The Animation Editor Toolbar



If a key point is selected, the animation-editor toolbar displays its frame and the value. If more that one key point is selected, their frames or values will only be displayed if they are the same for all selected points.

You'll notice that a number of commonly used function-curve tools found in the animation-editor menu bar are also represented as icons on the animation-editor toolbar for quick access.

The Property Tree

The property tree displays the animatable properties of the currently selected elements. You can use this tree to choose which properties to display as function curves in the graph editor as well as to select which curves to edit.

To expand and collapse a property tree

To expand an item in a property tree, click on its Expand icon (+). The elements, properties, and subproperties below it are listed, and the associated function curves are shown in the graph editor.

To collapse an item in a property tree, click on its Collapse icon (–). The items below it are no longer listed, and the associated function curves are hidden.

To pin an item (to keep it visible)

When you pin an element, it is always visible. You can then select other elements and still have access to the pinned element in the animation editor.

Pin icon

To pin an element, click on its pin icon in the property tree. If you pin a parent element, you can show or hide its subelements by expanding or collapsing it as described in the previous section.

To unpin an element, click its pin icon again.

To select and deselect a property's function curve

Clicking the element name in the property tree lets you select its associated function curve. Shift+click to select more than one curve at a time, and Ctrl+click to toggle a curve's selection.

Hide/display property's function curve

To hide and display a function curve

The icon to the left of the element name lets you hide or display the associated function curve. If the icon is next to a parent element, it will hide or display all the function curves associated with the subelements.

The Graph Editor

The graph editor is the area where you manipulate the function curve of any animatable parameter. You can edit the keyframed values by adjusting the key points on a selected curve.

You can:

- Add, move, and delete key points on the function curves.
- Change interpolation and extrapolation types.
- Paste, trim, stretch, or crop function-curve segments.

The graph's horizontal axis (X axis) displays the time scale in frames or milliseconds, as set in the user preferences. The vertical axis (Y axis) displays the values of the animated property.

The graph editor also shows a playback cursor (a red vertical line) that indicates the current frame on the timeline.

Controlling the View of Animation in the Graph Editor

There are a number of ways to control how animation in the graph editor is displayed. For example, you can display or hide the rulers along the axes, set ruler increments, and modify the time scale in which the animation is calculated.

To specify the graph editor's time labels

The Time Display property page in the User Preferences dialog box lets you specify how time is displayed, including the graph editor; you can display the X axis in milliseconds, or in frame numbers.

1. Choose **File** > **User Preferences** from the main-menu bar, then click the **Time Display** tab to display the Time Display settings.

2. In the Time Display Format box, choose how you wish time to be represented on the timeline in the animation editor.

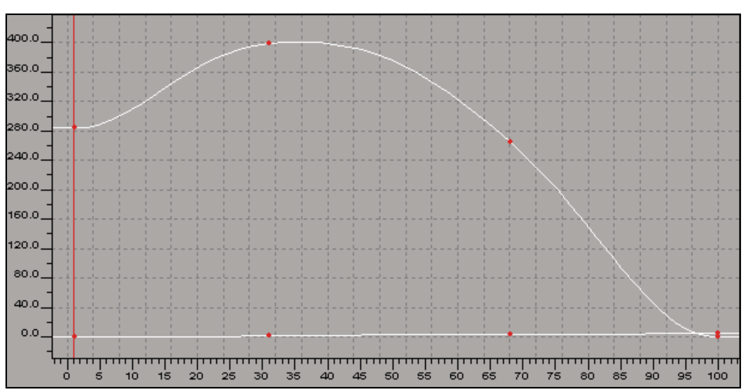
   Select **Display as Frames** if you want the time labels you selected in step 2 to be represented in frames rather than timecode on the timeline. **Display as Frames** is an option for any display format except milliseconds.

Regardless of whether you type timecode or a frame number in the Frame text box, SOFTIMAGE|XSI internally converts your entry into timecode.

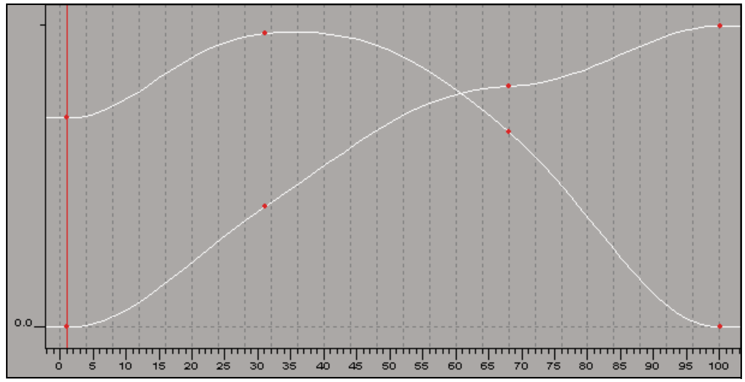
To normalize the graph's Y axis

Normalizing the graph's Y axis lets you view multiple curves that have a broad range of values.

Let's say you have one curve that spans Y values between 0 and 5, and another curve whose values range between 0 and 400. Viewing both of them on a standard scale would make the first curve very difficult to work with.



Normalizing the Y axis lets you view the full range of each curve (that is, you view 0–100% of its Y value).



To normalize the Y axis, choose **View** > **Normalized in Y** (**Value**) from the animation-editor menu bar.

To return to the default linear scale, select the option a second time.

To set grid and ruler display options

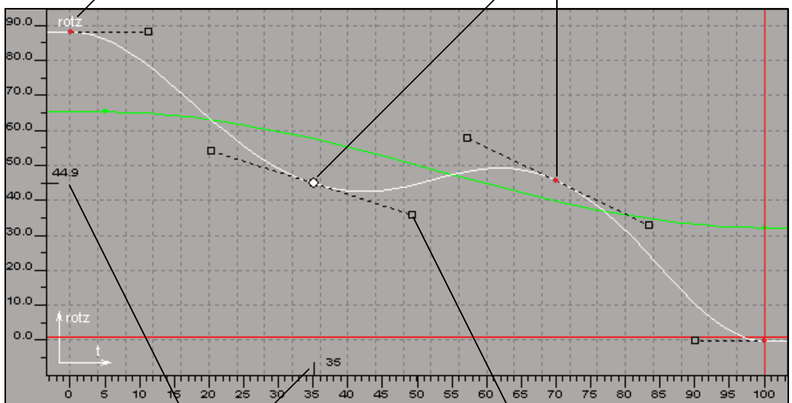
- To show or hide the grid, choose **View** > **Grid** in the animation-editor menu bar. You can also selectively hide the X-axis or Y-axis lines by choosing **File** > **Preferences** > **Editor** > **Grids** > **Display Grid** in the animation-editor menu.
- To show or hide the rulers, choose **View** > **Rulers** in the animation-editor menu bar.
- To modify the height and width of the grid pattern or set the scaling increments of the X and Y axes, choose **File** > **Preferences** > **Editor** in the animation-editor menu bar to display the animation editor preferences. You can also specify various grid snap and visibility options.

Viewing Function Curves

When the animation editor is open, it displays the animation (if any) of any elements you select.

Curve and *axis labels* indicate the type of curve and the affected axis. For example, this function curve represents the object's rotation on the Z axis.

Keyframed values on function curves are indicated by *key points*. Unselected key points are red, selected key points are white.



Key coordinates indicate the exact frame number and value of the most recently selected key point.

The *slope handles* indicate the speed at which a function curve's value changes at the key point. You can change the slope, and consequently the function curve, by manipulating the handles.

Displaying and Hiding Function Curves

The property tree lets you select and control the visibility of the function curves in the graph editor (for more details, see *The Property Tree* on page 65.)

The **View** commands in the animation-editor menu bar give you additional control over how function curves are displayed, such as:

- Which types of function curves to display.
- What sort of information each curve displays.

To control the display of function curves in the graph editor

From the animation-editor menu bar, choose:

- **View** > **All Parameters** to show the function curves of all animatable parameters for the selected object.
- **View** > **Animated Parameters** to show the function curves of all animated parameters for the selected object.
- **View** > **Marked Parameters** to show only the function curves of marked parameters for the selected object.

- **View** > **Clear All Curves** to hide all parameter function curves associated with the selected object without collapsing the property tree.

To specifying curve-display information

If you have many function curves displayed in your graph editor, you may want to hide a certain amount of curve information to unclutter the view.

In the animation-editor menu bar:

- Choose **View** > **Keys on Unselected Curves** to show or hide key points on all unselected curves in the graph editor.
- Choose **View** > **Slopes on Unselected Keys** to show or hide the slopes of all unselected keys on a selected curve.
- Choose **View** > **Curve and Axis Labels** to show or hide labels on all selected curves in the graph editor.
- Choose **View** > **Keys Coordinates** to show or hide the current time and value of a selected key point.

Because function curves for expressions and functions such as twist and bulge do not necessarily use the same horizontal units as other curves in the animation editor, Curve and Axis Labels should be activated in order to know the meaning of their values.

Zooming, Panning, and Framing

The Zoom, Pan, and Frame tools, available from the View menu, toolbar icons, and supra keys, let you view a specific area of the graph editor.

Zooming

You can interactively increase or decrease the magnification of the graph editor view using the Zoom icon in the toolbar or either the **s** (zoom) or **z** (pan and zoom) supra keys.

To activate zoom mode, click the Zoom icon or press the **s** hotkey. The cursor turns into a small magnifying glass.

- To zoom in (increase magnification), drag in the graph editor to define a region to enlarge.
- To zoom out (decrease magnification), Shift+click and drag. The region you draw defines the center of the region you want to see so you can define a large region to zoom out less and a small region to zoom out more.
- To increase and decrease the height of the view, right-click and drag up and down. To increase and decrease the width of the view, right-click and drag right and left.
- To reset the zoom factor, click once in the graph editor.

To deactivate zoom mode, select another tool or press Esc.

As an alternative, you can zoom in and zoom out using the **z** key.

- To zoom in, press the **z** key and hold down the middle mouse button. The zoom-in continues until you release the middle mouse button.
- To zoom out, press the **z** key and hold down the right mouse button. The zoom-out continues until you release the right mouse button.
- To pan, press the **z** key and drag with the left mouse button. The pan continues until you release the button or the **z** key.

Panning

You can pan to scroll the area shown in the graph editor using the Pan icon or either the **x** (pan) or **z** (pan and zoom) supra keys.

To activate pan mode, click the Pan icon in the toolbar.

- To pan, click and drag the mouse in the graph editor. To pan along the X axis, drag to the left or right. To pan along the Y axis, drag up or down.

As an alternative, you can pan using the **x** or **z** key.

- To pan across the graph, press the **x** or **z** key, hold down the left mouse button, and drag. The pan continues until you release the left mouse button.

To reset the graph editor to your original view, click once in the graph editor while pressing the **x** key.

To deactivate pan mode, select another tool or press Esc.

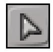
You can make the Pan icon default to the pan and zoom tool by checking **Pan & Zoom Tool by Default** in the animation editor preferences Editor page.

Framing

Framing sets the zoom and pan so that the curves fit in the graph editor within a specified range.

- To frame a curve segment, highlight the region to be framed and click the Frame Selection icon or press the **f** key.
- To frame key points, select the key points and click the Frame Selection icon or press the **f** key. If no key points are selected, the selected curves will be framed.
- To frame all displayed function curves, deselect all keys and function curves and click the Frame Selection icon or press the **f** key.
- To frame all displayed function curves listed in the property tree, press Shift+f.
- To frame all the function curves within the time span specified in the timeline controls, choose **View** > **Frame** > **To Timeline** from the animation-editor menu bar.

Selecting Function Curves



To select function curves and their associated key points, select mode must be on. To activate select mode, click the Select icon in the toolbar or press the space bar. To deactivate select mode, select another tool.

You can select a function curve while any toolbar feature is activated by clicking the element name in the property tree.

You can select one or more function curve at a time. Only selected function curves can be edited. When selected, function curves are highlighted in white and their key points are red.

To select a single function curve

If the function curve is visible in the animation editor, select a function curve by clicking on it.

If the function curve is not visible in the animation editor, click on the curve's associated parameter in the property tree.

Click anywhere outside the function curve or Ctrl+click the curve to deselect it.

To select multiple function curves

There are two ways to select multiple function curves displayed in the graph editor. First, activate select mode, then do one of the following:

- Shift+click the associated parameters in the property tree or the parameters' function curves in the graph editor to add them to the existing selection.

  or

- Ctrl+click the associated parameters in the property tree or the parameters' function curves in the graph editor to toggle their selection.

  or

- Drag the mouse pointer across a range of function curves in the graph editor; the red selection box indicates the region you are covering. Any curves that pass through the box will be selected.

  or

- Choose **Edit** > **Select All Curves** from the animation-editor menu bar to select all curves currently displayed in the graph editor.

To deselect all function curves, click on an empty area in the graph editor.

Selecting function curves by region

Besides selecting entire function curves, you have the option of selecting segments or regions of curves. Most commands related to function curves can be also be applied to segments of curves; also, curve-region edit controls let you move, stretch, or compress all the key points within the selected region as a group.

For information on how to select and edit function-curve regions, refer to *Moving Key Points* on page 73.

Working with Temporary Copies of Function Curves (Snapshots)

By default, when you move a key point or part of a function curve, a “snapshot” of the curve’s original location is taken. This snapshot displays a temporary copy of the original function curve until you are satisfied with the new, edited version. When you are satisfied with the new function curve, you can take a snapshot of the newly edited curve and continue editing; otherwise, you can swap the new function curve with its temporary snapshot to undo your changes.

To replace the snapshot function curve with new snapshot

1. Make sure that Snapshot Curve is active (from the animation-editor toolbar, choose **View** > **Snapshot Curve**).
2. Edit a function curve.

3. If you are satisfied with the edited curve, accept its new shape by clicking the Snapshot icon on the animation-editor toolbar.

   A new snapshot is taken at the curve’s new position and the old snapshot is removed.

To swap an edited function curve with its snapshot

1. Make sure that Snapshot Curve is displayed (from the animation-editor toolbar, choose **View** > **Snapshot Curve**).
2. Edit a function curve.

3. If you are not satisfied with your edited curve and want to return the function curve to its original position, click the Swap w. Snapshot icon on the animation-editor toolbar. The edited function curve and its snapshot exchange places.

Selecting (tagging) Key Points

You can select one or more key points on function curves. Selected key points appear in white, and their exact coordinates are displayed on the X and Y axes of the graph editor. In addition, key points using spline interpolation have slope handles that allow you to modify the slope orientation and length (see *Controlling the Shape of Spline Interpolations* on page 83 for more information).

To select a single key point, activate select mode (click the Select icon or press the space bar) and then click on the point.

- To select the next or previous key point, click the forward-arrow or back-arrow icons from the toolbar.
- To select multiple key points, activate select mode and then Shift+click on the points you want to select.
- To add or remove a key point from the current selection, Ctrl+click the key point.
- To select all the key points on the currently selected curves, choose **Edit > Select All Keys** from the animation-editor menu bar.

To deselect all key points, click on an empty spot in the graph editor. This will also deselect all the function curves.

Moving Key Points

You can move individual key points and curve segments using the mouse in the graph editor or with numerical precision using the animation editor preferences.

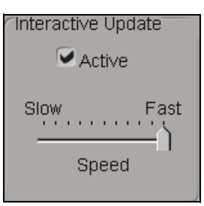
You can also select a specific region of a function curve in the graph editor and move, stretch, and compress the curve's key points within that region as a single unit.

Moving Points Interactively

To move one or more key points interactively with the mouse, select them and drag them to a new location. Moving horizontally changes the frame, and moving vertically changes the property value.

Holding down the Shift key while you drag key points with your mouse will constrain your displacement to the horizontal or vertical direction. In other words, if you intentionally move three units horizontally, then one unit vertically by mistake, holding down the Shift key will prevent you from travelling in the vertical direction. You can also use the middle mouse button to drag key points horizontally and the right mouse button to drag key points vertically.

Because moving points interactively can be imprecise, you might want to have your keys snap to specific time or value intervals. For more information on snapping to the grid, see *Snapping Keys to Time, Value, and Frames* on page 75.



If you want to see the effect of your changes in the viewports while dragging key points in the graph editor, check **Active** in the Interactive Update section of the animation editor preferences Editor page. You can adjust the speed of the update by moving the slider.

Moving Points Precisely

There are several ways to move key points to a specific location. The fastest way to modify a single key point is by selecting it in the graph editor and entering its new frame location and/or value in the **Frame** and **Value** text boxes in the toolbar.

To quickly enter the new point coordinates for a series of neighboring key points, use the animation editor preferences:

1. Select the key point you want to move.
2. Display the animation editor preferences by clicking the Preferences icon on the animation-editor toolbar.
3. Click the **Keys** tab and enter the key point coordinates in the **Frame** and **Value** text boxes.
4. Move to the next or previous keys by clicking the **Previous Key** and **Next Key** buttons below and continue entering key point values until you are done.

In the **Frame** text box, you can specify the value in SMPTE time code format (i.e., HH:MM:SS:FF) or you can simply enter the position in terms of number of frames (at 30 fps, entering a value of 45 would be the same as entering a value of 00:00:01:15).

To offset key points

You can displace one or more key points' frames or values by a specified amount using the **Frame** and **Value** text boxes in the toolbar.

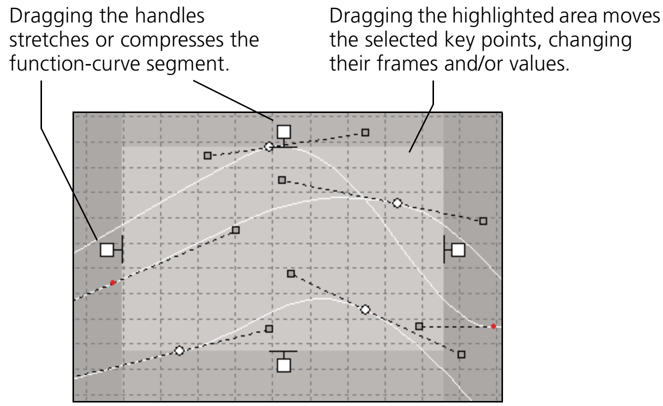
1. Select the key points you want to offset. To offset the entire function curve, select the curve and leave the key points unselected.
2. Enter the amount of the offset in the **Frame** or **Value** text box, followed by a + (addition), - (subtraction), * (multiplication), or / (division). For instance, to advance several key points by two frames, you would select the key points and enter 2+ in the Frame text box. To double the value of several key points, you would select the key points and enter 2* in the Value text box.

To move key points by region

1. Select the function curve or curves you wish to modify.
2. Click the Region icon in the toolbar, then drag the mouse pointer horizontally across a section of the function curve you wish to modify. Hold down the Alt key while dragging to snap the left and right edges of the region to keyframes. The selected area is highlighted in light gray.

If you want to extend or shrink the region you have selected, activate Region mode and press the Shift key while you drag. You can also toggle curves and individual keys within the region by Ctrl+clicking them.

Inside the region, between the maximum and minimum values of the selected key points and the first and last keys of the selected function curve, is a highlighted selection area. This area is bounded by four handles.



- To stretch or compress the function-curve segment, drag one of the handles. Dragging horizontally shortens or extends the function-curve segment over time. Dragging vertically increases or decreases the range of values in the segment.

If the Ripple icon is activated on the toolbar, stretching the curve segment horizontally will shift unselected keys in the graph editor. If Ripple is off, the curve segment cannot be stretched any farther than neighboring keys. For more information on the Ripple icon, see *Paste Options* on page 80.

- To change the frame position or values of the function-curve segment, click anywhere within the highlighted region and drag the region. Dragging the region horizontally adjusts the frame position of the function-curve segment (the animation occurs earlier or later in the timeline). Dragging the region vertically changes the values of the entire segment. All neighboring key points remain in their original positions.

Snapping Keys to Time, Value, and Frames

You can make your key points to snap to the nearest time and value coordinate on the grid's X and Y axes, or to the nearest frame.

To snap a key point to time and value

To snap a key point to both the grid's X (time) and Y (value) coordinates, choose **Edit** > **Snap to Grid** from the animation-editor menu bar.

To snap a key point to just one of the grid's two coordinates, open the animation editor preferences and click **Editor**. In the Snap Grid controls, select the **Snap** option below **X** (**Time**) if you want to snap the key points to the grid's time coordinate; select the **Snap** option below **Y** (**Value**) if you want to snap the key points to the grid's value coordinate.

If more than one key is selected and translated, just the translation amount is snapped to the grid. The keys that were on the grid remain on the grid and the others will remain outside of it. For example, if you have set the X (Time) grid to 5, you would only be able to translate keys horizontally in increments of 5 frames.

To snap a key point to its nearest frame

You can make your selected key points snap to the nearest frame on the graph editor's X axis. From the animation-editor menu bar, choose **Edit** > **Snap to Frame**.

If you want your keys to always snap to the nearest frame while you are working, choose **Edit** > **Discrete Time**.

Locking Key Point Position

You can freeze a key point's values as well as its location in time by locking the key's position on the graph editor's X and Y axis.

If the function curve uses spline interpolation to calculate the transition to the next key point, you can also freeze the key point's speed of transition and influence on the function curve via the animation editor's slope handle locking controls. For more information, see *Controlling the Shape of Spline Interpolations* on page 83.

To lock a key point's position on the graph editor

1. In the graph editor, select the key point whose value or location in time you want to freeze.

2. Do one or both of the following:

   - Choose **Keys** > **Lock in X** (**Time**) to lock the key's location in time on the X axis.

   - Choose **Keys** > **Lock in Y** (**Value**) to lock the key's value on the Y axis.

   The key point is frozen in place on the graph editor and cannot move in the locked axis until **Keys** > **Lock in X** (**Time**) or **Keys** > **Lock in Y** (**Value**) is selected again. You can still manipulate the key-point slope handles to change the shape of the slope, unless they have also been locked (see *Controlling the Shape of Spline Interpolations* on page 83).

Adding Key Points

You can also add key points interactively using the mouse. When you add a key point, it is the same as adding a keyframe to your animation. SOFTIMAGE|XSI recalculates the interpolation and draws new curve segments connecting it.

If you add a point on the same frame as an existing key point, the old point is replaced.

To add key points interactively

1. Select the function curve to which you want to add points.

2. Activate Add Key mode by clicking on the Add Keyframe icon in the toolbar or by pressing **a.**
3. In the graph editor, click where you want a key point.

Because moving points interactively can be imprecise, you might want to have your keys snap to specific time or value intervals. For more information on snapping to the grid, see *Snapping Keys to Time, Value, and Frames* on page 75.

Adding Key Points to Multiple Curves

You can automatically add key points to all selected curves on a given frame. While a new key is added for each curve, the curves retain their shape. This is useful when you want to insert a key for a group of parameters (e.g., X, Y, and Z) so that each key occurs at the exact same point in time.

To add key points to selected curves

1. Select the function curve or curves to which you want to add a key point.
2. Position the playback cursor to the frame where you want the key point(s) inserted.
3. Choose **Keys** > **Insert Key at Current Time** from the animation-editor menu bar.

   One key is inserted on each selected curve at the frame indicated by the playback cursor.

Deleting Key Points

When you delete key points, it is the same as deleting a keyframe from your animation. SOFTIMAGE|XSI recalculates the interpolation and draws a new curve segment connecting the remaining key points.

To delete specific key points

1. Select the function curves from which you want to delete points.

2. Activate delete mode by clicking the Remove Keyframe icon on the toolbar or by pressing **d**.
3. Click on the key points you want to delete.

To delete multiple key points

1. Select the curve or curves from which you want to delete points.
2. Using the Select or Select Region tools, select a region that includes all they key points you want to delete.
3. Press Delete.

To delete multiple key points on the same frame

1. Select the curves from which you want to delete points.
2. Move the playback cursor to the frame from which you want to delete points.
3. Choose **Keys** > **Delete Key at Current Time** from the animation-editor menu bar.

To delete all key points on a function curve

In the animation-editor menu bar, choose **Keys** > **Remove Animation**, or press Backspace.

When you delete all the key points on a function curve, you remove the animation of the associated property and the property's function curve reverts to a flat line in the graph editor.

Flipping Function Curves

You can flip a function curve or curve segment horizontally or vertically from the **Curves** menu, using either the **Flip X** (**Time**) or **Flip Y** (**Value**) command. Flipping a curve or curve segment transposes the key points across a dividing line determined by the midpoint between the highest and lowest key values on the selected axis.

If you flip a curve segment that does not start or end on a keyframe, SOFTIMAGE|XSI will create new keyframes at the start or endpoint.

You cannot flip a curve or curve segment that has keys locked in the same axis. For example, if a key is locked in time, **Flip Y** (**Value**) will function; if it is locked in value, **Flip Y** (**Value**) will not function.

Cutting, Copying, and Pasting in the Graph Editor

You can copy and paste a key point, a curve segment, or an entire curve. You can also set paste options to control how copied key points are pasted—whether they replace the selection or are added to it.

When you copy a curve segment, SOFTIMAGE|XSI can copy just the selected keys or the curve segment itself. This is determined by the method used to select the curve segment before copying or cutting.

- To cut or copy the curve segment's key points, use the Select Region tool or select a series of keys on the curve.
- To cut or copy the curve segment, use the Select tool and drag a rectangular selection over the curve.

To cut, copy, and paste a function curve

1. Select the function curve you want to copy.
2. Copy the curve to the clipboard by choosing **Edit** > **Copy** from the animation-editor menu bar, or cut it by choosing **Edit** > **Cut**.
3. Select a parameter (or its function curve) as the target that you want to paste onto.
4. Paste from the clipboard by choosing **Edit** > **Paste**.
5. You can paste the clipboard contents to other curves or segments by repeating steps 3 and 4.

You can use keyboard shortcuts for the cut, copy, and paste functions: Ctrl+c (**Edit** > **Cut**), Ctrl+x (**Edit** > **Copy**), and Ctrl+v (**Edit** > **Paste**).

To cut, copy, and paste selected key points

1. Select the key point or curve segment you want to copy. If you are selecting a curve segment, use the Region tool.
2. Copy them to the clipboard by choosing **Edit** > **Copy** from the animation-editor menu bar, or cut them by choosing **Edit** > **Cut**. A black line appears in the graph editor.
3. Select a function curve or curve segment as the target that you want to paste onto.
   - To paste the new keys starting at a specific frame on the function curve, click in the graph editor to move the black line to the frame.
   - To paste the new keys in a specific segment of the function curve, select the curve segment using the Select Region tool.

4. Paste from the clipboard by choosing **Edit** > **Paste**.
   - If the target is a curve segment, the clipboard contents will be scaled to fit the number of frames in the selected area.
5. You can paste the clipboard contents to other curves or segments by repeating steps 3 and 4.

If you want to paste the curve segment after performing other functions in the animation editor, you can use the playback cursor to determine where the segment will be pasted.

Paste Options

The Paste Options property page in the animation editor preferences determines how the clipboard contents will be pasted. To open the Paste Options property editor, choose **File** > **Preferences** > **Paste Options** from the animation-editor menu bar.

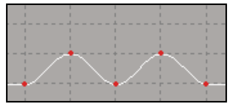
- **Insert**: The curve segment on the clipboard is added to the selected function curve to the left of the current selection. The rest of the function curve is moved to the right of the inserted segment.
- **Replace**: The curve segment on the clipboard is overlaid on the equivalent area on the curve.

When using the Replace option, you can paste either absolute or relative values.

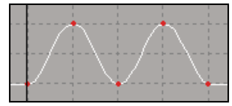
- **Absolute:** The curve segment on the clipboard completely overwrites the equivalent area on the curve.
- **Relative:** The curve segment on the clipboard is pasted as an offset from the first key point in the pasted area.

The Continuity Options determine how the first and last keys of the pasted curve segment will be integrated into the curve.

- **Left Key: Use Average:** The first key of the pasted region is the average of the values of the copied and original curves at that point.
- **Left Key: Use Left Value:** The first key of the pasted region uses the value of the original curve at that point.
- **Left Key: Use Right Value:** The first key of the pasted region uses the value of the copied curve at that point.
- **Right Key: Use Average:** The last key of the pasted region is the average of the values of the copied and original curves at that point.
- **Right Key: Use Left Value:** The last key of the pasted region uses the value of the copied curve at that point.
- **Right Key: Use Right Value:** The last key of the pasted region uses the value of the original curve at that point.

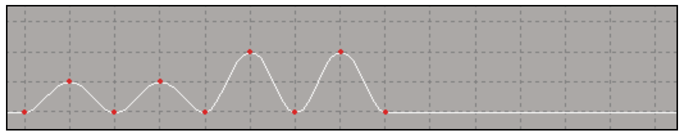


Curve A

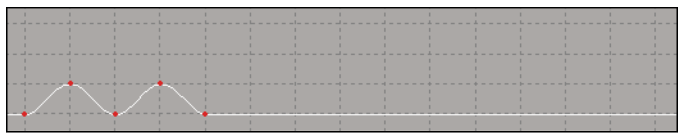


Curve B

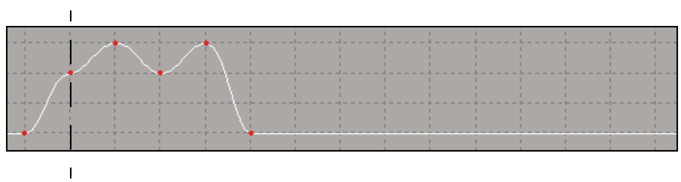
Curve A's key points have been copied and will be pasted on curve B. Note the black bar on curve B, which is where curve A's key points will be pasted.



When using Insert, the new key points push the existing keys to the right.



When using Replace with the Absolute option, curve A's key points are placed on top of the original keys.



The black bar has been moved to a new location (indicated by the dashed line for clarity) before pasting. Curve A's key points have been pasted using Replace with the Relative option. Since the Left Key continuity option is set to Use Left Value, the first key point is set to curve B's value at that point. The rest of curve A's key points are pasted relative to the first key point. The exception is the last key; the Right Key continuity option is set to Use Right Value, so it is also set to curve B's value at that point.

The Ripple Option

The Ripple icon on the animation-editor toolbar provides a quick way to paste a curve segment using the Insert option.

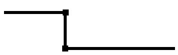
To activate Ripple, click the Ripple icon before you paste the curve segment.

Choosing a Function Curve Interpolation Type

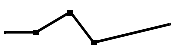
Function-curve interpolation is a way to estimate how animation progresses between key points. There are several ways to calculate interpolation. For example, linear interpolation creates sudden changes at certain keyframes, while spline interpolation creates smooth animation with an ease-in (deceleration) and ease-out (acceleration) between keyframes.

To set the interpolation type

1. Select a key point or a curve segment.
2. Choose one of the interpolation options from the animation editor's **Curves** menu:



- **Constant Interpolation** repeats the value of a key point until the next one. The movement is characterized by sudden changes at keyframes and static positions between keyframes.



- **Linear Interpolation** connects key points by straight line segments. The movement is characterized by constant speed with sudden changes at each keyframe.

- **Spline Interpolation** is the default. It uses a curved spline interpolation. The interpolation accelerates and decelerates to ease into and out of each keyframe, resulting in a smooth transition. The degree of acceleration and deceleration before and after the keyframe is determined by the slope handles associated with the key point.

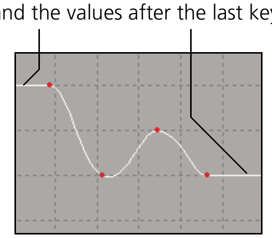
Editing a Function Curve's Slope

Controlling Slope Shape Before and After the First or Last Key Points

The animation editor extrapolation controls let you modify the shape of a function curve before the first key and after the last key.

Constant extrapolation extrapolates the value of the Y axis from the first frame to the first keyframe and the value of the Y axis from the last keyframe to the last frame. This results in a hold of motion before and after the movement.

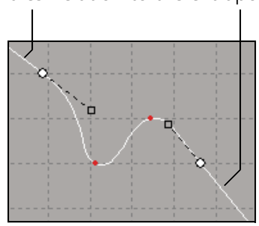
With constant extrapolation, the values before the first keyframe and the values after the last keyframe remain constant.



Gradient extrapolation extrapolates an increment based on the first key's slope orientation and applies it from the first frame to the first keyframe.

Next, it extrapolates the increment calculated between the last and previous key points and applies it from the last key frame to the last frame.

With gradient extrapolation, the values before the first keyframe and the values after the last keyframe are determined by the slope. Note the slope handles' relation to the extrapolated values.



To apply extrapolation between to a curve

1. Select a function curve in the animation editor.
2. Choose **Curves** > **Constant Extrapolation** or **Curves** > **Gradient Extrapolation** from the animation-editor menu bar.

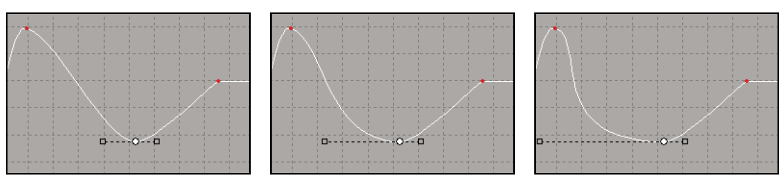
For spline interpolation, the slope of the first or last key is used. This results in a continuation of the motion at the same speed.

Controlling the Shape of Spline Interpolations

When a key point with spline interpolation is selected two handles, called *slope handles*, extend out from the key point. By modifying their length and direction, you can define the way the curve moves toward the previous and next key points.

In terms of the slope's shape, the steeper the slope's orientation, the faster the values change.

The length of the slope handles determines the weight of the key point's influence on the function curve. The longer the handle, the more the key point influences the connecting curve segment.



The same function curve, with short, medium, and long slope handles.

As you change the orientation and length of the slope handles, you can also set how the handles move: the left and right handles can be manipulated independently or together, or locked in place.

To change the orientation or length of the slope handles

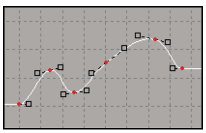
1. Select a key point in the graphics editor; slope handles appear.
2. Click on the end of a handle and drag it to a new location.

If you are having trouble selecting a slope handle because it lies too close to a point, press **h** to give priority to handle selection.

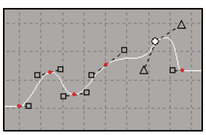
To set slope-handle options

The Keys menu on the animation-editor menu bar contains slope options that control how slope handles move as you manipulate them.

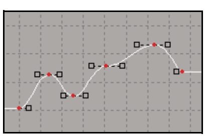
From Keys menu in the animation-editor menu bar, select one of the slope options:



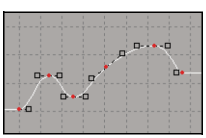
- **Unify Slope** keeps the slopes of the selected keys together as they move, keeping a constant angle between them. Unified slopes provide smooth, unbroken function-curve segments, as values ease into and out of key points. In the actual animation, the rate of the parameter's change decelerates as the timeline approaches the keyframe, and accelerates as the timeline leaves the keyframe.



- **Break Slope** lets you control the slope's animation on either side of the key point individually. Depending on the slopes you define, the transition in the rate of the parameter's change at the keyframe can be as smooth or as abrupt as you like.
- **Automatic Slope** computes the orientation and length of a key point's slope based on the interpolation of the function curve between the previous key and the next key.



- **Zero Slope** constrains key points to a slope of zero. The slope handles remain horizontal so that only the length can be modified. This is useful to prevent the automatic interpolations from going above or below your keyframed values as they approach key points.



- **Plateau Slope** automatically sets the slope of a key point to zero if the next key has the same value, or if that key is a local minimum or maximum. A local minimum/maximum is a key point where both previous and next keys have either smaller (maximum) or larger (minimum) values. This is useful to hold key point values to prevent the automatic interpolation from going above or below keyframed values that define extremes. When Plateau Slope is active, it affects all points on selected curve.

To lock a key point's slope handles

If the function curve uses a spline interpolation, you can freeze a key point's speed of transition and influence on the function curve by locking the orientation and length of its slope handles:

1. In the graph editor, select the key point whose handles you want to freeze.
2. Do one or both of the following:
   - Choose **Keys** > **Lock Slope Orientation** to lock the slope handles' orientation.
   - Choose **Keys** > **Lock Slope Length** to lock the length of each slope handle.

   The slope handles of the selected key point are frozen in place on the graph editor and cannot be moved until **Keys** > **Lock Slope Orientation** or **Keys** > **Lock Slope Length** is selected again. You can still manipulate the position of the key point itself, unless its position has also been locked (see *Locking Key Point Position* on page 76).

You can use the same commands to unlock key points or slope handles.

Repeating Function Curves (Creating Cycles)

The Cycle commands available from the animation-editor menu bar provide you with several quick ways to repeat animation. By using these controls, you can create repetitive motion including bouncing and oscillation effects.

Creating a Basic Cycle

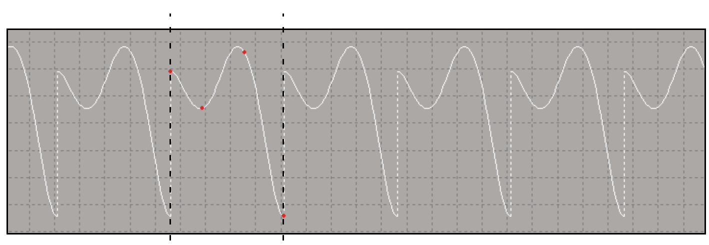
The **Curves** > **Cycle** command in the animation-editor menu repeats the function curve shape/pattern defined by the key points. This results in the cycling of that pattern from the first keyframe to the last frame.

When a cycle is created, the repeated information is not actually added to the function curve, which provides a few benefits: you can switch between different types of cycles at any time, and changes to the original keyframes are reflected in the cycle as well.

If you have one or more key points on your function curve that are not on a frame, you will get odd effects because the cycle is evaluated at each frame and not on an intermediate time. Choose **Edit** > **Move Keys to Nearest Frames** to remedy this.

To create a basic cycle

1. In the graph editor, select the function curve whose shape/pattern you want to repeat.
2. In the animation-editor menu bar, choose **Curves** > **Cycle**.
3. The function curve is repeated continuously in both directions in time.



The original key points are indicated between the dashed lines. Note how the resulting cycle has no key points.

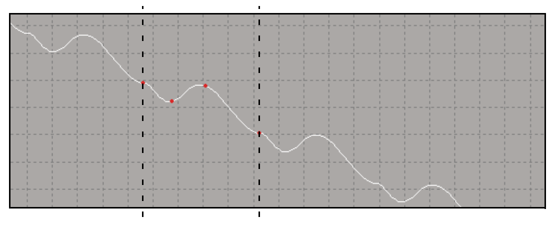
Creating a Relative Cycle

The **Curves** > **Relative Cycle** command in the animation-editor menu repeats the function curve shape by making each cycle start with the same value as the last key of the previous value. This results in a curve cycle with a progressive offset.

Setting a progressive offset is particularly important when you want to create a simple walk cycle. Animating foot translation in a repeated cycle on the Y axis generally requires no offset, but cycling animation on the X axis without an offset will make the character look like he's walking without moving forward.

To create a relative cycle

1. In the graph editor, select the function curve whose shape/pattern you want to repeat.
2. In the animation-editor menu bar, choose **Curves** > **Relative Cycle.**



The original key points are indicated between the dashed lines.
Again, note how the resulting cycle has no key points.

Deleting Function-Curve Cycles

Since a cycle's repeated information is not actually added to the function curve, you can delete the cycle completely while leaving the original keyframes intact.

To delete the cycle information, select **Curves** > **Constant Extrapolation** or **Curve** > **Gradient Extrapolation.**

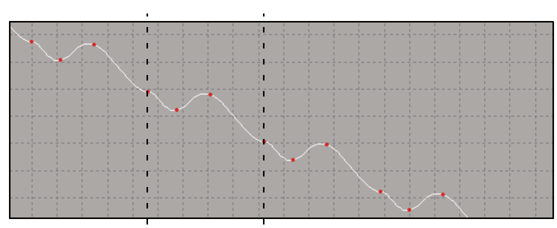
Freezing Function-Curve Cycles

The **Curves** > **Freeze Cycle** command in the animation-editor menu freezes the selected function curve by creating (or sampling) the keys represented on the cycled part of the function curve. The function curve can then be edited.

To freeze a function-curve cycle

1. In the graph editor, select the function curve whose shape/pattern you want to freeze.
2. In the animation-editor menu bar, choose **Curves** > **Freeze Cycle.** If no keys on the curve are selected and no region is highlighted, a dialog box appears. Enter the start and end frame that you want to use to freeze the cycle.

   The cycle is frozen with new key points on the function curve.



The original key points are indicated between the dashed lines.
The frozen curve now has key points, which can be modified.

Saving and Retrieving Function Curves

The function curves you create in the animation editor can be saved and subsequently applied to other parameters. You can also import and export SOFTIMAGE|3D function curves.

To save a function curve

1. In the animation tree or graph editor, select the function curve you want to save.
2. Choose **File** > **Save** from the animation-editor menu bar to save the function curve for use in SOFTIMAGE|XSI. Choose **File** > **Export** (**.fraw**) to export the function curve for use in SOFTIMAGE|3D.
3. Navigate to the folder where you want to save your curve, give it a name, and click OK.

   Your curve is now saved as a function-curve file, indicated by its **.fraw2** (SOFTIMAGE|XSI) or **.fraw** (SOFTIMAGE|3D) extension.

To load function a curve

1. In the animation tree or graph editor, select the function curve you want to replace.
2. Choose **File** > **Open** from the animation-editor menu bar to load an .fraw2 function curve. Choose **File** > **Import** (**.fraw**) to import an .fraw function curve.

The loaded function curve replaces the currently selected curve.

Processing Function Curves

Usually, function curves created within SOFTIMAGE|XSI have key points that are placed at various frames throughout your animation. SOFTIMAGE|XSI evaluates the function curve and interpolates between the key points.

Function curves generated from external sources such as motion capture may have many more key frames than necessary (commonly one key point per frame), or they may contain “noise”—sharp spikes and jags in the function curve. You may want to remove some of the key points or reduce the noise to edit the animation.

Conversely, if you keyframe an animation in SOFTIMAGE|XSI and you want to transfer the animation data to an external device that requires frame-by-frame information (for example, a camera), you may want to increase the number of keyframes and preserve the motion as best as possible.

To increase the number of keyframes on a function curve

1. Open the animation editor preferences.
2. Choose **Curve Processing** > **Time Step**. The time step determines the frequency of the generated keyframes. For instance, a time step of 1 will create key frames on every frame, and a time step of 2 will create key frames on every other frame. You can also choose to keep or discard your original keys by selecting **Keep Existing Keys** in this dialog box.
3. Select a function curve in the animation editor.
4. Choose **Curves** > **Resample** from the menu bar.

To reduce noise in a function curve

1. Open the animation editor preferences.
2. Choose the smoothing method.
   - Use the **Average Filter** setting to apply a moving average where the weights of the averages are uniformly distributed. You can define the number of keys to average at a time using **Filter Size**. A higher number of keys will result in a straighter curve.
   - Use the **Gaussian Filter** to apply a moving average where the weights of the averages are distributed as a bell curve. Use the **Variance** setting to control the degree of smoothing. A higher variance results in a smoother curve. The Gaussian filter usually provides better results than the average filter.
3. Select a function curve in the animation editor.
4. Choose **Curve Processing** > **Smooth**.

To reduce the number of keyframes on a function curve

1. Open the animation editor preferences.
2. Choose **Curve Processing** > **Tolerance** to set the processing accuracy. Smaller values yield a more accurate end result, but can create more keys.
3. Select a function curve in the animation editor.
4. Choose **Curves** > **Fit** from the menu bar.

Chapter 3 Animating along Paths and Trajectories

A path provides a route in global space for an object to take in order to get from one point to another, and the object's center is constrained to the curve for the duration of the animation. A path is a specific type of constraint, which is discussed more fully in the next chapter.

You can create path animation by:

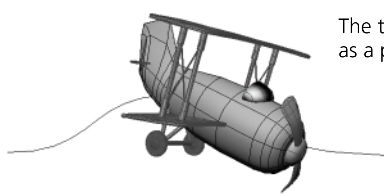
- Picking an existing curve as the path for an object. The keys will represent percentages of the path length.
- Moving an object about your scene and saving key-path positions at different frames. The path curve is created automatically as you go. The keys can either represent a percentage of the path length or be locked to a particular point along the path.

After you have created path animation, you can modify it like any other constraint. You can also modify the path curve to change the animation.

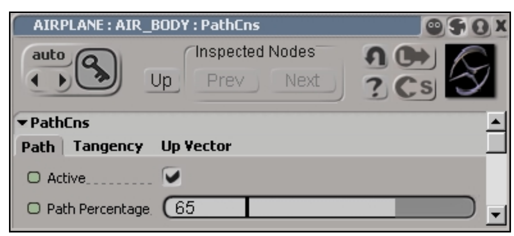
Trajectories are similar to paths, in that they are specialized constraints that control an animated object's translation. Unlike a path, which interpolates an object's position between keyframes, a trajectory's knots serve as indicators of the object's position at each frame. A trajectory requires no keys because the object jumps to the next knot on the curve at each frame. This means you have absolute control as to where your object will be from frame to frame.

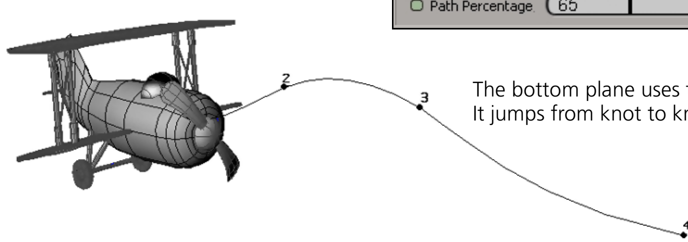
You can create trajectory animation by picking an existing curve as the trajectory for an object and selecting the starting frame for the animation.

The trajectory constraint is almost identical in its properties to the path constraint. Most of the features outlined in this chapter are applicable to both paths and trajectories.



The top plane uses path animation. Its position is measured as a percentage along the curve, as seen below.

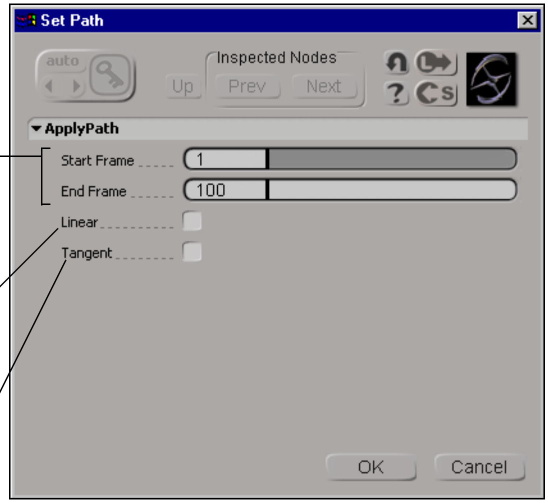




The bottom plane uses trajectory animation. It jumps from knot to knot at each frame.

Picking Paths

You can use the Set Path dialog box to quickly animate an object along a path.



Start Frame and End Frame determine when the object will begin traveling along the path and when it will stop. By default, the Set Path dialog box sets the object to travel the full length of the path.

If Linear is checked, the object will travel along the path at a constant speed; otherwise, the object will ease into and out of the animation.

If Tangent is checked, the object's X axis will be aligned with the path's slope throughout the animation.

To animate an object along a path

1. Create a curve using any of the available tools.
2. Select the object you want to animate.

Make sure the selected object does not have compensation on when creating a path: this would key the offsets rather than path animation. For more information on compensation, see *Introducing Offsets between Constrained and Constraining Objects* on page 112.

3. Choose **Create** > **Path** > **Set Path** from the Animate toolbar. The Set Path dialog box appears.
4. Enter the frame at which the object starts moving (Start Frame) and the frame at which the object reaches the end of the path (End Frame).
5. If you want the object to move along the path at a constant speed, check **Linear**. If you want the object's X axis to stay aligned with the path's slope, check **Tangent**.

   For more information on constraining an object's tangency, see *Constraining an Object by Tangency* on page 103.
6. Click OK.
7. Pick the curve. Keys will be created for each end of the path and the object's intermediate positions along the path will be interpolated.

8. The PathCns property editor will appear, allowing you to manually adjust your constraint settings.

You can also create path animation for a selected object by using **Constraint** > **Curve** (**Path**) from the Constraint panel and picking the curve. This opens the PathCns property editor, where you can set keys to animate the path percentages manually.

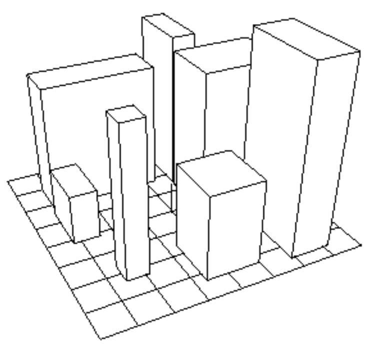
The parameterization of the curve has no effect on timing.

Example: Path through a City

This example shows you how to apply path animation to the camera and its interest. The **Create** > **Path** > **Set Path** command is used instead of interactively moving the camera and saving translation keyframes.

Create a Cityscape Using Cubes and a Grid

1. Choose **Get** > **Primitive** > **Polygon Mesh** > **Grid** and accept the default settings. Choose **Get** > **Primitive** > **Polygon Mesh** > **Cube** and accept the default settings. Align the bottom of the cube with the grid.
2. Click **Center** in the Selection panel and translate the cube's center in the Y axis to the bottom of the cube. Click **Object** in the Selection panel.
3. Choose **Edit** > **Duplicate Multiple** to create seven copies of the cube. Scale the cubes so that they are all different sizes and place them on the grid. Use the figure below as a reference.



Create a Path for the Camera and Its Interest

4. Choose **Create** > **Curve** > **Sketch** from the Model or Animate toolbar and draw a curve path that goes through the city.

Animate the Camera along the Path

5. Change the timeline's End Frame to 300.

6. To select the camera, you can do any one of the following:
   - Open the explorer and select the camera.
   - Click the camera in a viewport. If the camera is hidden in a viewport, choose **Show** > **Cameras** from the viewport and choose the camera from the list.

7. Choose **Create** > **Path** > **Set Path** from the Animate toolbar. Use the default values for the start and end frames. Check **Linear**, and leave Tangent unchecked. Click OK, then pick the path.

8. Play back the animation. You will notice that the camera is following the path but the camera interest stays fixed.

Animate the Camera's Point of Interest on the Path

9. Select the camera interest by clicking it in a viewport or the explorer.

10. Choose **Create** > **Path** > **Set Path**, and set the start and end frame values to –20 and 280. Pick the curve path.

11. Play back the animation. The camera interest is now traveling along the path slightly ahead of the camera.

12. Experiment with moving points along the path (the **m** supra key) to change its slope and with translating the path along the Y axis.

To see more of the city scene while flying the camera, open the camera's property editor and change the field of view, or interactively set the field of view by zooming in the Camera viewport.

Picking Trajectories

You can animate an object along a trajectory by picking a curve as the object's trajectory.

1. Create a curve using any of the available tools.

2. Select the object you want to animate.

3. Choose **Create** > **Path** > **Set Trajectory** from the Animate toolbar.

4. Pick the curve. The Trajectory Constraint property editor opens, allowing you to change the animation start frame setting from its default of 1.

Creating Paths from Translation Keys

You can define path animation by moving an object within the scene and saving keys at different frames. The path curve is created automatically as you go.

1. Move the playback cursor to the frame where you want the animation to begin, then select the object you want to animate and translate it to where you want it to start.

2. Choose one of the following commands from the **Path** menu on the Animate toolbar:

   - **Save Key on Path** sets a key corresponding to a percentage of the path's length. If you later modify the path by adding, deleting, or moving its points, the timing may change because the same percentage would correspond to a different location on the curve.

   - **Save Locked Key on Path** sets a key corresponding to a specific location on the curve. The object will always pass through that specific location on the curve at that frame, even if the shape or length of the path changes.

3. Change frames, move the object, and repeat step 2. SOFTIMAGE|XSI automatically updates the curve and timing information. Continue until you have completely defined the path.

Converting Existing Animation to a Path

Converting an Object's Translation to a Path

You can use the **Create > Path > Convert Position Fcurves to Path** command to convert part or all of an object's existing translation animation into path animation.

1. Select the animated object.

2. Choose **Create > Path > Convert Position Fcurves to Path** from the Animate toolbar. The Convert Position Animation to Path dialog box appears.

3. Enter the Start Frame and End Frame to set which segment of the function curve will be used to create the path. Any animation before the Start Frame or after the End frame will be discarded.

4. Enter the desired interval between keyframes as the Step. SOFTIMAGE|XSI will create the path with knots based on the object's location at the Step interval. For example, if the start and end frames are 10 and 20 and the Step is 5, the curve will be created with knots based on the object's location at frames 10, 15, and 20.

Because the path is a curve, an extra knot will be created after the first and another will be created before the last.

SOFTIMAGE|XSI will replay the chosen segment of the animation and create the new path.

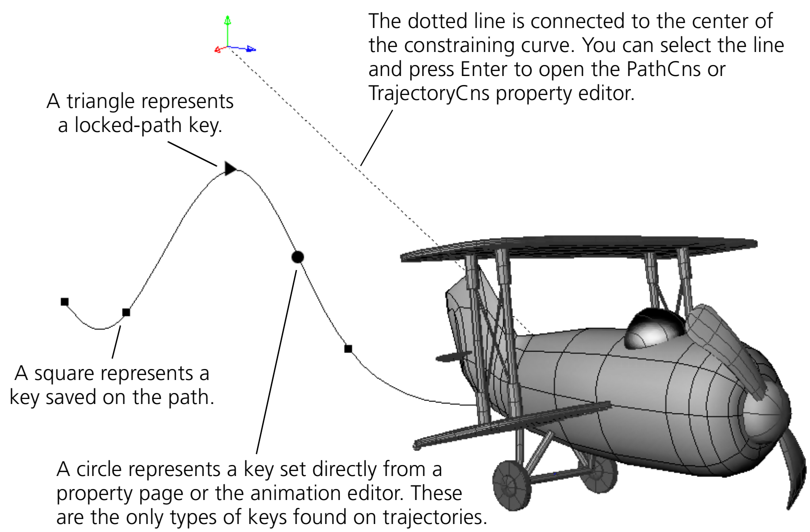
Viewing Path and Trajectory Information

SOFTIMAGE|XSI can provide a visual reference to path and trajectory information such as key points, types of constraints, and constraint options.

To display constraint information in a viewport, choose **Show** > **Relations** from (the (Show menu of the) viewport title bar. A dashed line appears, connecting the center of the selected object and its constraining path. Key points and their corresponding frame numbers are also displayed:

- A square represents a key saved on the path.
- A triangle represents a locked path key.
- A circle represents a key set directly from a property page or the animation editor. These are the only types of keys found on trajectories.

If there is an up vector, it will be indicated by an arrow.



For finer control of constraint-information display, open the Visibility Settings property editor by choosing **Show** > **Visibility Options** from the viewport title bar, then click the **Attributes** tab. You can then display constraint information independently for selected and unselected objects using the **Relations** (links and key points) and **Relations Info.** (frame numbers and other labels) options.

The constraint information can also provide easy access to the PathCns property editor. To open the property editor, click the dotted line representing the constraint then press Enter.

Viewing Path Information in the Animation Editor

You can view your path animation's function curve in the animation editor by opening the PathCns property editor, right-clicking on the Path Percentage animation icon, and choosing **Animation Editor**. Alternatively, if you are displaying constraint information in a viewport, you can click the constraint line and open the animation editor.

Once the animation editor is open, the path percentage function curve is displayed, with key points representing keys on the path. Locked keys on the path are blue, locked in value (but not time) in the animation editor. You can unlock the key by selecting it and choosing **Keys** >**Lock in Y** (**Value**) from the animation editor menu.

Modifying Paths and Trajectories

A path is a specific type of constraint. Once you have created path animation, you can modify how the constraint is applied.

Constraint property editors are described in more detail in *Chapter 4: Animating with Constraints*.

Directly Modifying a Path or Trajectory

You can modify the paths directly by using one of three methods:

- You can move, add, or remove points by selecting the curve and choosing **Point** from the Selection panel.
- If the path was created using **Create** > **Curve** from the Model or Animate toolbars, you can open its Geometry property page to modify its shape. You can also change the U subdivisions to increase or decrease the number of points in the curve.
- You can translate the constrained object and use its new location to simultaneously create a new knot and keyframe on the path. To do this, translate the object at the appropriate frame, then choose **Create** > **Path** > **Save Key on Path** or **Save Locked Key on Path**.

Displaying Path and Trajectory Constraint Property Editors

The PathCns property editor lets you modify the properties of the path animation.

To display the PathCns property editor from the Selection panel

1. Select the object on the path.
2. Click the **Property** button on the Selection panel. A pop-up explorer opens.
3. Choose **Kinematics** > **Constraints** > **PathCns** or **TrajectoryCns**.

To display the PathCns property editor from a viewport

1. From the viewport title bar, choose **Show** > **Relations**.
2. Select the object on the path. A dashed line indicating a path constraint link appears.
3. Click on the constraint link and press Enter.

For information about the Tangency and Up Vector property pages, refer to *Constraining an Object by Tangency* on page 103.

Modifying Path Timing

To modify the path timing, adjust the **Path Percentage** in the Path property page and set keys at different frames. You can also right-click the parameter's animation icon to view and modify the Path Percentage function curve in the animation editor.

Unlike paths, a trajectory constraint does not create a function curve.

From the PathCns property editor or the animation editor, you can set keys with values below 0% or higher than 100% of the path. SOFTIMAGE|XSI extrapolates path percentages outside of the 0–100% range, depending on the type of curve used as a path:

- On a closed curve, such as a circle or a square, using path percentages less than 0% or greater than 100% will continue to translate the object on the curve, with increasing percentages resulting in clockwise movement (relative to the local front of the object). For example, if an object were keyed to start at 0% of a circular path and end at 150%, the object would make one complete clockwise revolution of the circle and continue for another half a revolution. If the end were at –150%, the object would travel the same distance counterclockwise.
- On an open curve, such as a spiral or a NURBS curve, keys below 0% place the object at the curve's starting point; keys above 100% are place it at the curve's endpoint.

Removing Path Animation

To remove path or trajectory animation from an object

1. Select the object.
2. Choose **Constraint** > **Remove Constraint** from the Constraint panel.

   If the object has more than one type of constraint, the status bar will prompt you to pick the constraint to remove. If this happens, pick the constraining curve.

Alternatively, you can delete the constraint from the explorer:

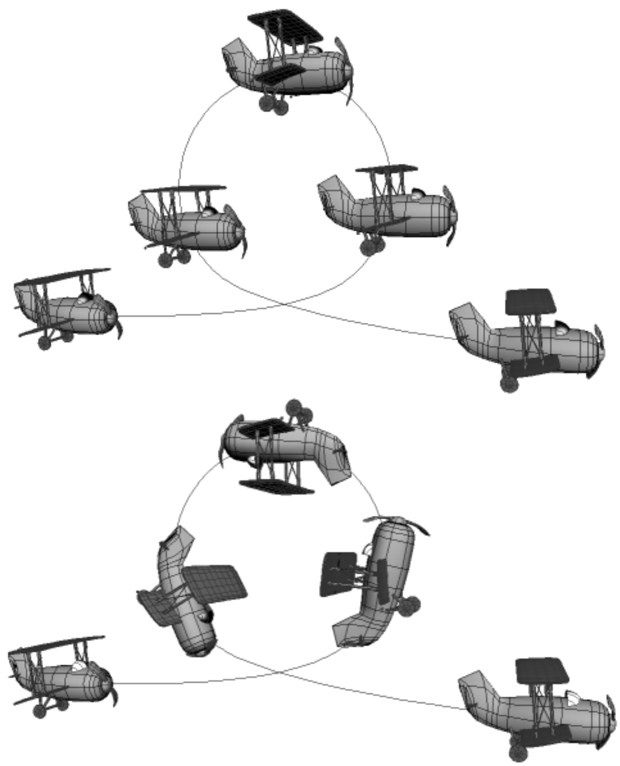
1. Make sure that the explorer is set to show **All Nodes**, **All Except Parameters** or **Clusters and Properties**.
2. Expand the object's Kinematics branch.
3. Select the PathCns or TrajectoryCns node and press the Delete key.

If constraints are visible in a viewport, you can also click the dotted line representing the constraint and press Delete.

Constraining an Object by Tangency

The tangency constraint constrains the orientation of an object to be tangent to a curve (path); that is, the object's selected axis (the X axis by default) points in the direction of the curve's slope at all times.

This constraint is particularly useful for having an object follow a path's direction, such as a roller coaster car following the tracks.



The plane in the top figure has been animated on a path, but its tangency has not been constrained. It will always be oriented the same way, regardless of its position on the path. In the bottom figure, the plane's Y axis has been constrained to be tangent to the path. Note that the nose of the constrained plane does not always rest on the path; the constraint uses an object's center to set tangency.

You can also use path constraints in conjunction with the up-vector constraint to define the rotation of an object around its tangent. For more information on the up-vector constraint, see *Up-Vector Constraints* on page 128.

To constrain an object's tangency

1. Create the curve to be used as the path.
2. Select the object you want to constrain to the path.
3. Choose **Create** > **Path** > **Set Path** from the Animate toolbar. Use the default values for the start and end frames and pick the path.
4. Open the PathCns property editor by clicking the **Property** button on the Selection panel and choosing **Kinematics** > **Constraints** > **PathCns** from the pop-up explorer.
5. Click the **Tangency** tab.
6. Check the **Active** check box.

The constrained object's X axis is repositioned in the direction of the curve's slope. As the object moves along the path, its X axis follows the curve's slope.

The **X, Y,** and **Z** sliders define a vector along which the object's center will be aligned.

Example: Rocket

In this example, you will animate a rocket by drawing a curve and then attaching it to that curve. You will use the Tangency property page to ensure that the rocket follows the path with the appropriate orientation.

Create the Rocket

1. Choose **Get** > **Primitive** > **Polygon Mesh** > **Cone** to create the rocket cone. In the cone's property editor, set Height to 2.
2. Choose **Get** > **Primitive** > **Polygon Mesh** > **Cylinder** to create the rocket fuselage. In the cylinder's property editor, set Height to 8.
3. Translate the cylinder so that it is flush with the cone.
4. Choose **Get** > **Primitive** > **Polygon Mesh** > **Cube** to create a rocket fin. In the cube's property editor, set the Length to 2.
5. Scale the cube to about 0.75 on the X axis, and 0.05 on the Z axis.
6. Translate the fin down to the bottom of the fuselage, flush against its side.
7. Choose **Point** from the Selection panel. Using the rectangle selection tool, select the cube's upper right points in the Front viewport.
8. Translate the selected points down and to the left, until you get a pleasing fin shape.
9. Use the rectangle selection tool with the middle mouse button to deselect the cube's points.
10. Choose **Object** from the Selection panel to exit point selection mode, then choose **Center**.

11. Translate the fin's center to the center of the fuselage's bottom in the X and Z axes. Deselect **Center** from the Selection panel.
12. Choose **Edit** > **Duplicate/Instantiate Options** from the Edit panel to open the Duplicate Options property editor.
13. In the Options property page, set the Placement to Apply/Repeat Xform.
14. Click the **Transform** tab. Set the Y rotation to 90, then close the Duplicate Options property editor.
15. Choose **Edit** > **Duplicate Multiple.** Set the Number of Copies to 3, then click OK. This will create the other three fins along the base of the fuselage.
16. Parent the fuselage to the cone and fins.

Set a New Time Frame and Draw Your Path

17. Change the last frame of the timeline to 200 in order to extend the playing time of your scene.
18. Draw a curve by choosing **Create** > **Curve** > **Draw CV NURBS** from the Model or Animate toolbar. To modify your points, use the **m** supra key with the left mouse button.

Attach the Rocket to the Curve

19. Select the missile by right-clicking on the fuselage. Choose **Create** > **Path** > **Set Path** from the Animate toolbar. Use the default values for the start and end frames, click OK, and pick the path. The PathCns property editor appears.
20. Click the **Tangency** tab.
21. Check the **Active** checkbox.
22. Set X to 0 and Y to 1. This will align the rocket along its local Y axis so that it points in the direction of its flight.

You can also accomplish this by rotating the rocket's center so that the X axis points toward its tip and leaving the tangency at its default settings.

23. Play back your animation.

As an additional experiment, try giving the rocket a rolling effect by animating the up vector constraint as the rocket makes a turn.

Chapter 4 Animating with Constraints

Constraints allow you to constrain objects, including lights and cameras, to other objects. This is very useful for creating complex behavior by having one or more objects automatically react to another's animation.

In terms of time savings, the value of constraining objects is considerable. Suppose you want to animate a group of people at an air show watching a plane fly overhead. To do this efficiently, you would constrain each spectator's head to the plane and then animate only the plane itself.

If you used keyframes to animate the same scene, you would have had to generate function curves for each head movement as well as the plane. With the Direction constraint, the only object you have to keyframe is the plane.

Constraint Package

There are many ways to constrain objects. You can constrain them by virtually any animatable parameter, including position, rotation, and scaling. SOFTIMAGE|XSI has packaged a number of constraints into predefined commands. For example, the **Constraint** > **Direction** command constrains the direction of a selected, or "aligned," axis (the default is the X axis) of one or more objects to another object. The location of the constrained objects do not change, but their aligned axes always face the center of the constraining object, no matter where the constraining object is.

Each constraint command is described in detail in this chapter, except for **Constraint** > **Curve** (**Path**), which is dealt with in *Chapter 3: Animating along Paths and Trajectories* on page 91.

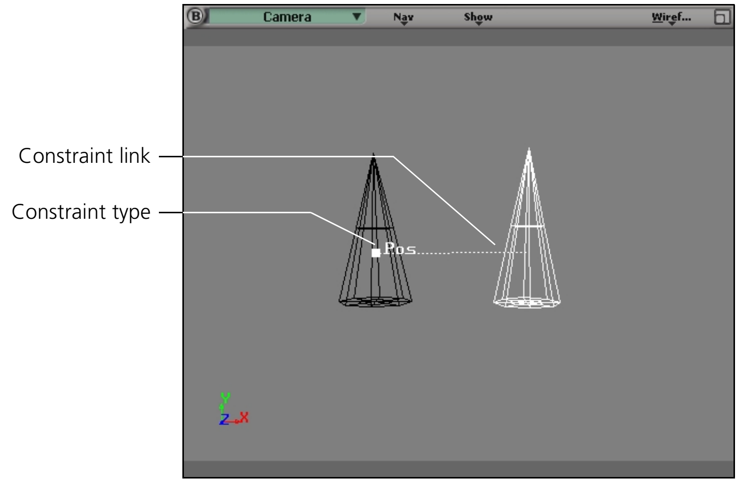
Customizing Your Constraints

In addition to the predefined commands available in SOFTIMAGE|XSI, you can use mathematical expressions to develop more sophisticated constraints between objects, such as setting a cube's color according to the average position of two models and a light. For more information on how to use expressions to define constraints, refer to *Chapter 5: Animating with Expressions* on page 133.

Viewing Constraint Information

In any viewport title bar, choose **Show** > **Relations**.

This shows you the type of constraint in effect plus a dashed line between the constrained and constraining objects.



For more control over how to view constraint information, go to the viewport menu bar and choose **Show** > **Visibility Options** and click the **Attributes** tab. From the list of options that displays, you can choose to show constraint icons or constraint information for selected or unselected objects, or both.

Activating and Deactivating Constraints

You can temporarily deactivate a constraint and activate it later.

1. Select object(s) whose constraints you want to activate/deactivate.
2. From the Selection panel, choose **Explore** > **Constraints**.
3. In the Constraints folder, choose the property node whose constraints you want to activate/deactivate.
4. In the property page that displays, toggle the **Activeness** checkbox to activate/deactivate the constraint.

Activating Multiple Constraints

It is possible to have more than one constraint applied to a particular object at the same time. When more than one constraint is active at the same time on an object, the latest constraint to be applied is used.

Avoiding Constraint-Dependency Cycles

Unless you are careful, building complex constraint relationships between objects can eventually lead to a cycle of dependencies. As a simple example, constraining **a** to **b** (such as with an orientation constraint) then trying to constrain **b** to **a** will create a cycle of dependencies.

Setting Coupling Behavior Between Constraints

The coupling behavior between constrained objects can be either **soft** or **rigid**.

Soft is where objects are allowed to rotate somewhat independently of one another, much like a ball joint. Soft constraints are represented by a small circle in the centers of the constrained and constraining objects.

Rigid is where the constrained object mimics the rotation of the constraining object. Rigid constraints are represented by a small square in the centers of the constraining objects.

Imposing Limits to Constraints

When constraining objects, you can set maximum and minimum limits to their position and its rotation. You can, for example, specify that a sphere can rotate no more than 90 degrees from its current rotational position. By setting limits to its transformations, the constrained object is prevented from exhibiting awkward or unnatural movement as it follows the constraining object.

For information on how to impose limits to an object's position and rotation, refer to *Imposing Limits to Transformations* in Chapter 6 of the *Fundamentals* guide.

Editing Constraints

When you constrain an object to another object, you cannot usually edit the constrained object's parameters (such as position, orientation, scaling, etc.), because it is subordinate to its constraining object. There are, however, several ways to change a constrained object's parameters while maintaining the constraint.

Global versus Local

Constrained objects are transformed in global space. As such, any edits you make to the object's local transform property editor will have no effect on the constrained object.

Introducing Offsets between Constrained and Constraining Objects

If you constrain one object to another without any offset, both objects will end up sharing the same coordinate points and will seem to be on top of one another. In most cases, you will want to separate the constrained and constraining objects with an offset of some sort. The example that follows shows you how to use the Constraint panel's Compensation controls to achieve this.

Example: Creating an Offset

Let's say you want to constrain a cylinder to a cone using the Position constraint, but you introduce a slight offset to the constrained cylinder.

Create the Constraint

1. Create a cone and a cylinder.
2. Select the cone. (This will be the object to be constrained.)
3. Choose **Constraint** > **Position** from the Constraint panel and pick the cylinder. (This will be the constraining object.)

   The cone's position coordinates are now constrained to those of the cylinder.

Apply an Offset between the Cone and the Cylinder

4. Select the Compensation feature by choosing **Constraint** > **Compensation** or by clicking the **Comp.** button in the Constraint panel.
5. Move the cone to where you would like it to be in relation to the cylinder.
6. Deactivate Compensation by choosing **Constraint** > **Compensation** again or clicking the **Comp.** button again.
7. Now move the cylinder. You will find that the constrained cone maintains the offset.

If you wanted to use the original distance that separated the cone from the cylinder as the offset, you would have clicked the **Comp.** button after Step 1 then deselected **Comp.** after the position constraint was created.

Creating or Modifying Offsets Interactively

Even after you have created a constraint between two or more objects, you can create or modify their offsets at any time by means of the constraint's property editor.

Editing a Constraint from Its Property Editor

You can introduce offsets as well as set a number of other options from your constraint's property page.

To open a constraint's property editor from its icon in the viewport

1. If you have not already done so, go to a viewport menu bar, choose **Show** > **Visibility Options** (from the Show menu), click the **Attributes** tab and, in the Selected Objects area, toggle **Relation Info** on.

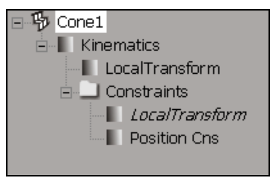
   Any constraints in your scene should appear in the viewport.

2. Select the dashed constraint link and press Enter.

To open a constraint's property editor from the explorer

1. Select the constrained object, then choose **Explore** > **Constraints** from the Selection panel.

   An explorer window opens, showing the constrained object's root and the constraints folder below it. The constraints folder contains all the constraints attached to the object.

2. Select a constraint node to display its associated property page.

   From this page you can:

   - Edit the constraint's offsets (whenever they are modified, the offset control's XYZ axis properties are automatically marked so you can quickly animate them).
   - Activate/deactivate the constraint.
   - Set coupling behavior (see *Setting Coupling Behavior* below).
   - Set additional constraints, such as tangency and up-vector constraints (see **Combining Constraints** below).

Combining Constraints

In many cases, you can set a combination of constraints to objects. For example, the **Up Vector** controls in the Direction constraint's property page can be used to control the "roll," or up direction, of the constrained object.

In all cases, the most recently applied constraint overrides other, older constraints should a conflict between their properties arise.

Removing Constraints

As opposed to deactivating constraints, where the constraint is still present but has no effect until it is reactivated, you can also remove constraints permanently between objects.

To remove a constraint

There are two ways:

- If the constraint link is visible between the constrained and constraining objects, select it and press the Delete key.

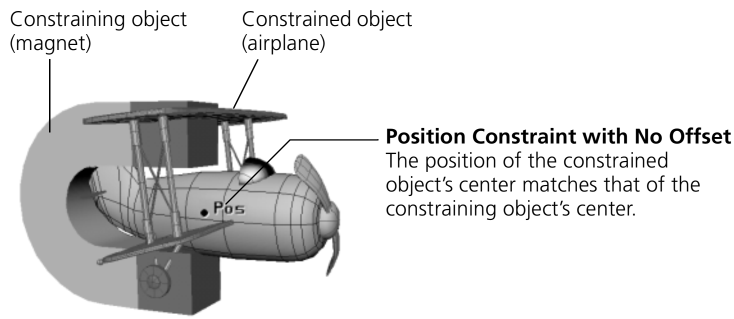
  or

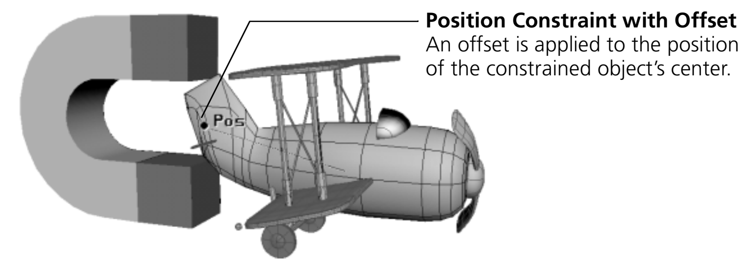
- Select the constrained object and choose **Constraint** > **Remove Constraint**. If the object is constrained to more than one object, you will be prompted to pick the object from which you want to remove the constraint.

If you want to remove all constraints from an object, select the object and choose **Constraint** > **Remove All Constraints**.

Position Constraints

You can take the center of one or more objects and constrain it to another object or a null.





To constrain an object's position

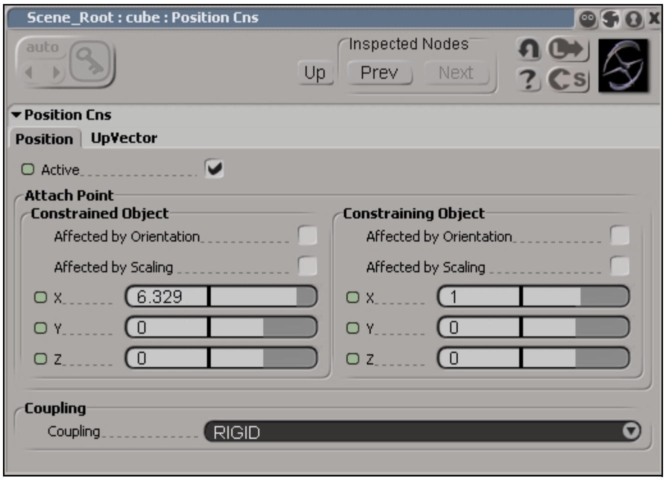
1. Select the object to be constrained.
2. Choose **Constraint** > **Position**.
3. Pick the object that is to act as the constraining force.

   The constrained object repositions its center to the center of the constraining object, less any offset you may have specified. Now, when a transformation is performed on the constraining object, the constrained object's center remains aligned.
4. If you switch off the **Affected by Orientation** control in the Position Constraint property editor for either the constrained or constraining object, object rotation is not included as part of the constraint for that object. The **Affected by Scaling** control operates in a similar manner.

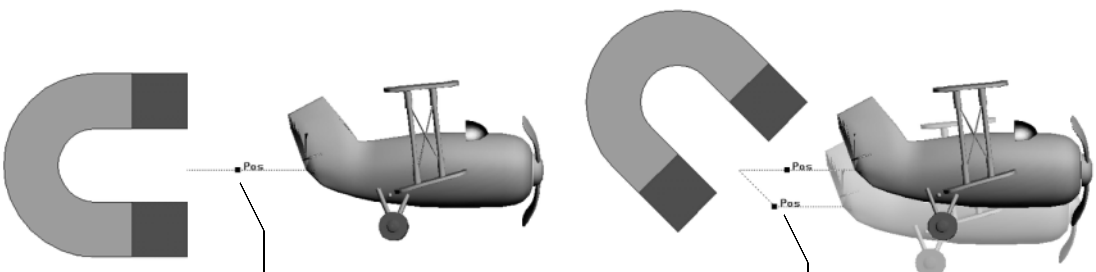
Adding Offsets to the Constraining/Constrained Object’s Center

By default, constraints are applied to the centers of the constrained and constraining objects. You can offset the centers, if required.

Example: Introducing an Offset to the Center of a Constraining Object



In the **Attach Point - Constraining Object** axis controls of the Position Constraint property editor, enter the axis offset of the constraining object’s centre.



The position constraint is now calculated according to the constraining object’s center, which in this example is offset on the X axis by 1.

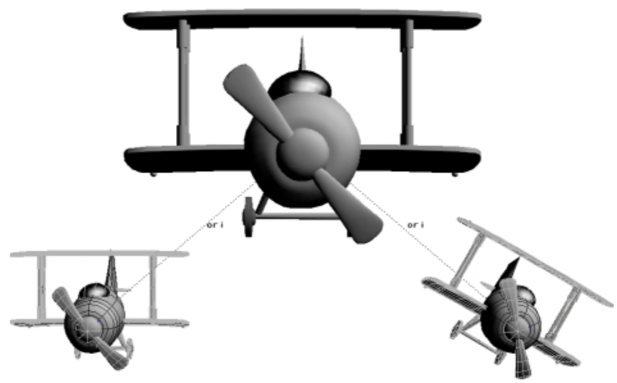
(An offset has already been applied to the position of the constrained object’s center.)

If **Affected by Orientation** is selected, the position constraint is influenced by the orientation (rotation) of the constraining object.

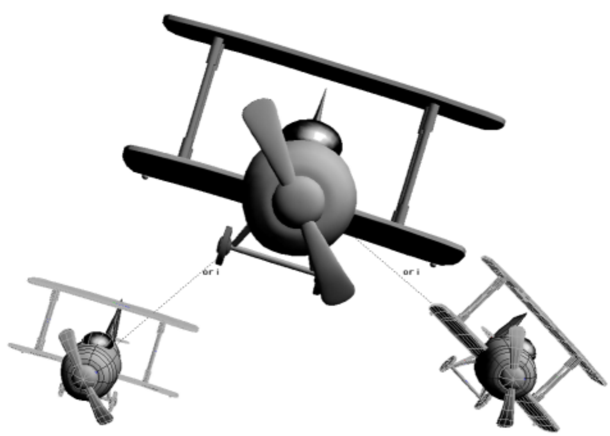
Orientation (Rotation) Constraints

You can constrain the rotation values of one or more selected objects to another object without changing the location of the constrained objects. A rotation performed on the constraining object is also performed on the constrained objects.

For example, you can make a group of people's heads all move in sync by animating one head and then constraining the rotation of all the others to this head.



Two small planes (Plane A and Plane B) are constrained by orientation to the large plane. Plane B (lower right) has been offset by 30 degrees.



When the large plane rotates, the smaller planes rotate as well.

To constrain an object's orientation

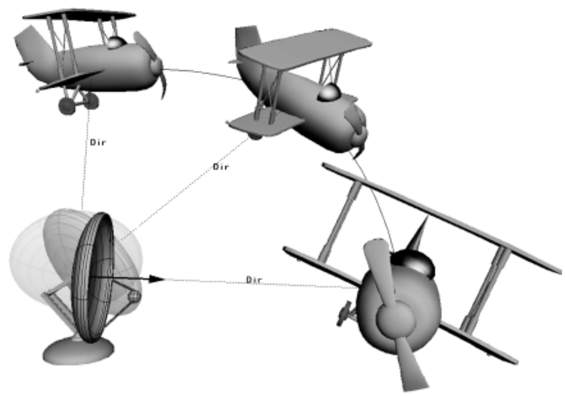
1. Select the object or objects to be constrained.
2. Choose **Constraint** > **Orientation**.
3. Pick the object that is to act as a constraint.
4. Select the constraining object and rotate it as desired.

   The constrained objects also rotate.

You can set rotation limits for inverse-kinematic chains. You set these limits in the Skeleton Joint property page. For more information on setting rotation limits see *Imposing Limits to Transformations* in Chapter 6 of the *Fundamentals* guide.

Direction Constraints

You can constrain an axis of one or more selected objects to another object without changing the location of the constrained objects. The aligned axis (the X axis by default) of each constrained object always remains facing the center of the constraining object.



Radar constrained by direction to the plane
The X axis of the radar dish continually points in the direction of the plane's center.

To constrain an object by direction

1. Select the object to be constrained.
2. Choose **Constraint** > **Direction**.
3. Select the object that is to act as a target.

   The constrained object rotates so that its X axis faces the constraining object's center.
4. Translate the constrained or constraining object as desired.

   If the constraining object is translated, the constrained object responds by rotating so that its specified axis remains facing the constraining object. If the constrained object is moved, its specified axis continues to face the constraining object from its new location.

The rotation of the constrained object around its axis can be modified and keyframed by changing its axis-rotation value (known as the "roll" angle of the object). The commands in the Transform panel, however, continue to display the rotation values of the object in global or local coordinates.

The **Up Vector** controls in the Direction constraint's property page can be used on the constrained object to control the "roll," or up direction, of the object.

Pose Constraints

Pose is a combination of scaling, orientation, and position constraints. There's no attachment type associated with this feature. This constraint allows fine control over which value (per degree of freedom) is carried from the constraint to the constrained item. Furthermore, it offers per-transform control over the hierarchical style of manipulation, which involves making sure two objects always appear to remain connected even if a rotation or scaling is applied to the parent node. (This involves selective "compensation": When one rotates or scales the parent object, the rotation or scaling on the child is compensated, but not its translation).

To constrain an object to a pose

1. Select the object to be constrained.
2. Choose **Constraint** > **Pose**.
3. Select the object that is to act as a constraint.
4. Select either a constrained object or the constraining object, and perform a transformation as desired.

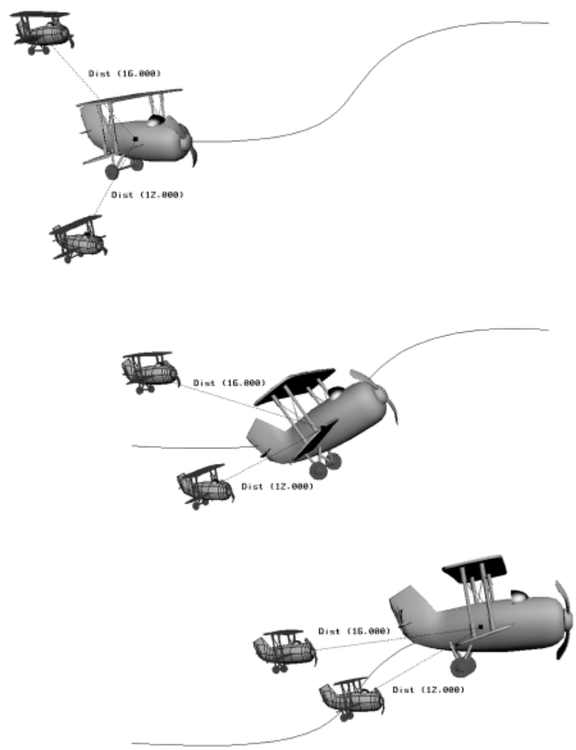
   The constrained object will remain constrained to the constraining object's position, scaling, and orientation.

Modifying Pose Constraints

1. Select the constrained object, then in the explorer select the object's constraint node.
2. In the **Pose Constraint** property page, set any offsets in the controls under the **Pose** tab.
3. Under the **Options** tab, you can turn on or off the position, scaling and orientation constraints.

Distance Constraints

You can constrain an object so that its center maintains a constant distance from that of the constraining object.



The centers of the smaller planes maintain a constant distance from that of the constraining object (the large plane).

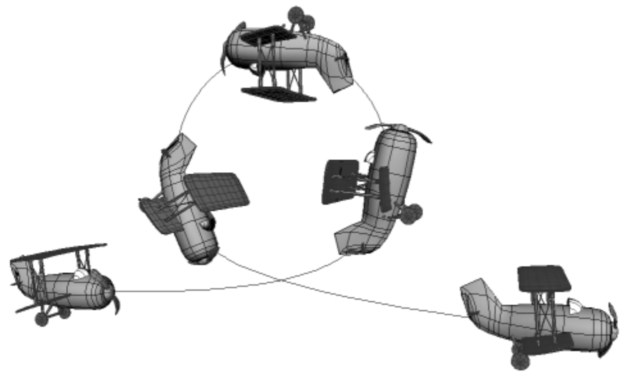
To constrain an object by distance

1. Position the object to be constrained at the required distance from the constraining object.
2. Click the **Comp.** button to set the distance offset.
3. Choose **Constraint** > **Distance**.
4. Select the object that is to act as the constraint.
5. Deselect the **Comp.** button.

This constraint uses soft coupling only.

Curve (Path) Constraints

You can constrain an object to a curve and animate its position on that curve. This constraint is particularly useful for having an object follow a path's direction, such as a roller coaster car following the tracks.



You can also use this constraint in conjunction with the **Up Vector** controls in the Path constraint's property page to define the rotation of an object around a chosen axis.

To constrain an object to a curve

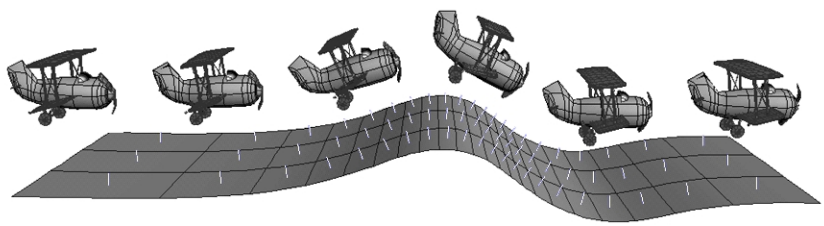
1. Select an object.
2. Choose **Constraint** > **Curve** (**Path**).
3. Select the path. By default, the constrained object's X axis is repositioned in the direction of the slope of the curve. As the object moves along the path, its X axis follows the curve's slope.
4. Use the path's percentage slider to animate the object along the path.

For more information on techniques related to path animation, see *Chapter 3: Animating along Paths and Trajectories* on page 91.

Surface Constraints

You can constrain the displacement of an object based on the surface U/V values of the constraining object. Much like a surface-based path, this type of constraint orients the constrained object according to the normal of the closest point on a surface.

The **Constraint** > **Surface** command constrains the Y axis of an object to the direction of the nearest surface normal of the constraining object. This constraint works only with NURBS objects as the surface object.



Surface constraint
Plane is rotated so that its Y axis is aligned with the normal of the surface polygon.

This constraint is useful for keeping an object perpendicular to a surface—usually when the surface is deformed, such as a ship sailing over a section of choppy water.

To constrain an object to a surface

1. Select the object to be constrained.

   You may want to choose **Show** > **Normals** from the viewport menu bar to see the object's normals. If the normals are pointing in the wrong direction, choose **Modify** > **Surface** > **Inverse** in the Model toolbar.

2. Choose **Constraint** > **Surface**.

3. Select the constraining object. It must be a NURBS surface object.

4. In the Surface Constraint property editor, click the **Normal** tab and toggle **Up Vector Active** on. Make sure the Y axis is set to 1 in the Affected Axis controls below.

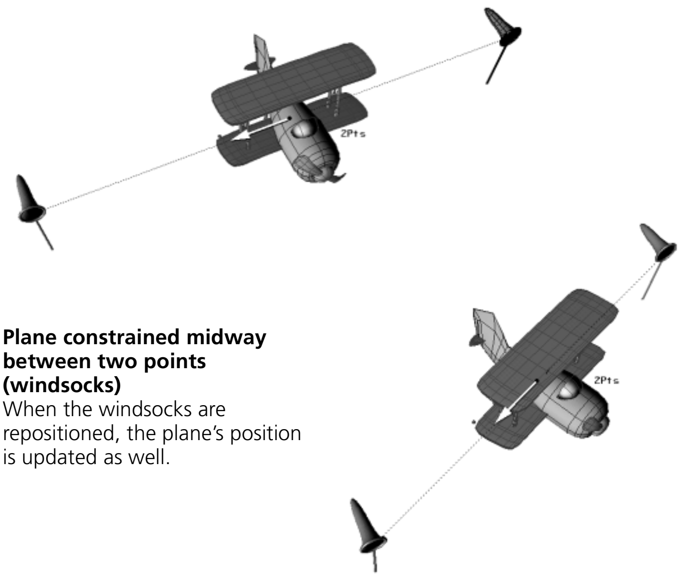
   The constrained object's Y axis is repositioned in the direction of the surface normals of the constraining object.

5. If required, animate the constrained object using the U/V position sliders in the Surface Constraint property page.

Constraints between Points

Constraining between Two Points

You can position an object to be constrained centrally between two other objects or nulls. By default, the constrained object's X axis is oriented along the vector that goes from the first-selected constraint to the second one. You can, however, select the object to be constrained by its Y or Z axis.



Plane constrained midway between two points (windsocks)
When the windsocks are repositioned, the plane's position is updated as well.

This constraint is useful for keeping an object between two points pointing along their direction, such as a muscle between two arm joints.

It also provides an easy way to convert two positions into an orientation for motion capture. When you have motion curve data without orientation information, you can infer rotations from position data by using this constraint.

To constrain an object between two points

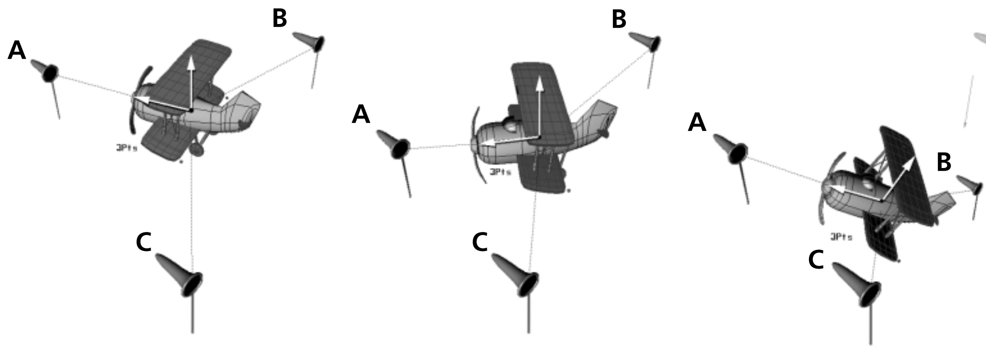
1. Select the object to be constrained.
2. Choose **Constraint > 2 Points**.
3. Pick two constraining objects.

The order in which you select your constraining objects is important to get the correct orientation of the constrained object.

The constrained object is repositioned to the mid-point between the two constraining objects, and by default its X axis is repositioned in the direction of the second constraining object's center. As you move the constraining objects (one or both), the constrained object readjusts to keep between them.

Constraining an Object between Three Points

You can also position an object to be constrained centrally between three constraining objects or nulls in a type of invisible triangle. By default, this orients the constrained object's Y axis to the normal of the plane defined by the three constraining objects and its X axis toward the center of the first constraint object you pick.



Three-point constraint
Constrain plane to windsocks A, B, and C. Pick A first.

Move windsock A
Plane remains at midpoint.

Note the orientation
X axis of plane is oriented to A. Y axis is oriented to normal of plane defined by windsocks.

This constraint is useful for converting three positions from motion capture data into an orientation. When you have motion curve data for positions only, such as three points on the hips and trunk of an actor, you can infer the rotation of its body by using this constraint.

To constrain an object between three points

1. Select the object to be constrained.
2. Choose **Constraint** > **3 Points**.
3. Pick three constraining objects.

The order in which you select your constraining objects is important so you can get the correct orientation of the constrained object. By default, the first constraint defines the X axis and the two remaining constraints together define the orientation of the plane for the Y and Z axis.

The constrained object is repositioned to a central point between the three constraining objects. Its X axis is repositioned in the direction of the first constraining object's center, and its Y axis is oriented to be perpendicular to the plane formed by the three constraining objects. As you move the constraining objects (one or all), the constrained object readjusts to keep properly positioned and oriented.

If required, set the coupling type from the constraint's property page (see *Combining Constraints* on page 114).

Constraints between Multiple Points

You can also position an object to be constrained centrally between multiple constraining objects or nulls. This constraint calculates the average position of all the points and applies the resulting value as a constraint.

To constrain an object between multiple points

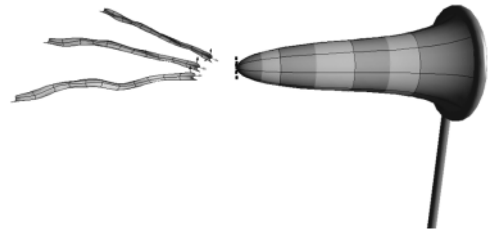
1. Select the object to be constrained.
2. Choose **Constraint** > **N Points**.
3. Pick all the constraining objects you require.

The order in which you select your constraining objects is important so you can get the correct orientation of the constrained object.

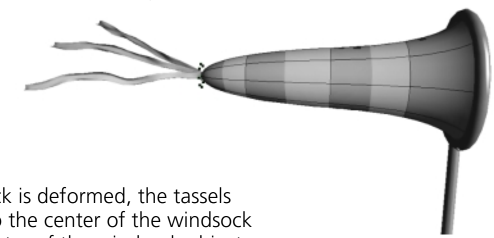
Object to Cluster Constraints

You can constrain an object to the center of a cluster.

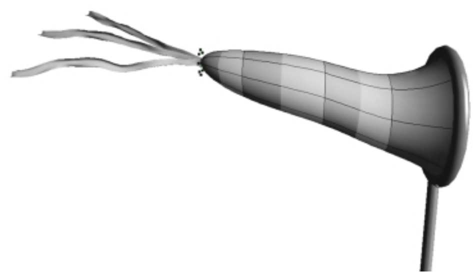
Three tassels constrained to the end of a windsock



Each tassel object is constrained to a cluster
that forms part of the windsock object.



When the windsock is deformed, the tassels
move in relation to the center of the windsock
cluster, not the center of the windsock object.

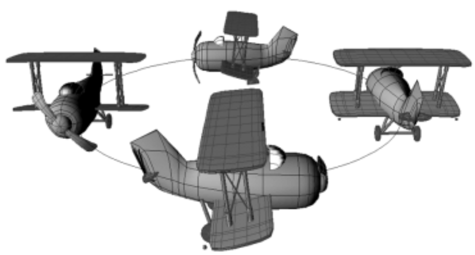


To constrain an object to a cluster

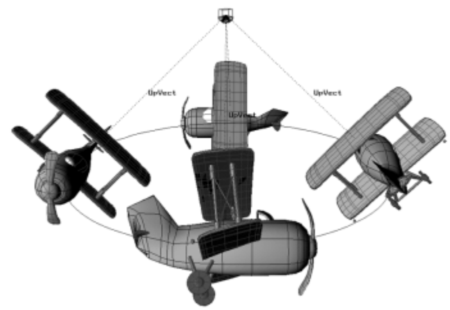
1. Select the object to be constrained.
2. Choose **Constraint** > **Object to Cluster.**
3. In point-selection mode, select the points of the cluster. The object will be constrained to the center of the selected points.

Up-Vector Constraints

In the case of models, the up-vector constraint has an effect only if the model has a Direction, Position, Surface, 2 Point, 3 Point or a Curve (Path) constraint. By default, these constraints make the X axis point to a constraining object. The up-vector constraint makes the Y axis point to another constraining object.



Curve constraint only
Plane's X axis is oriented to the curve's X-axis tangent.



Curve with up-vector constraint
Plane banks as Y axis is oriented toward the up-vector cone. X axis remains oriented to curve tangent.

You can also control the up direction (or Y axis) of a camera or an object by using a secondary constraint object or null.

There are two reasons why you would want to do this:

- To prevent the camera from flipping when it reaches a vertical axis with the interest. For example, you can produce a proper roller-coaster–like action with the camera using an up-vector constraint.
- To control the banking of the camera using a separate control path.

Creating an Up-Vector Constraint

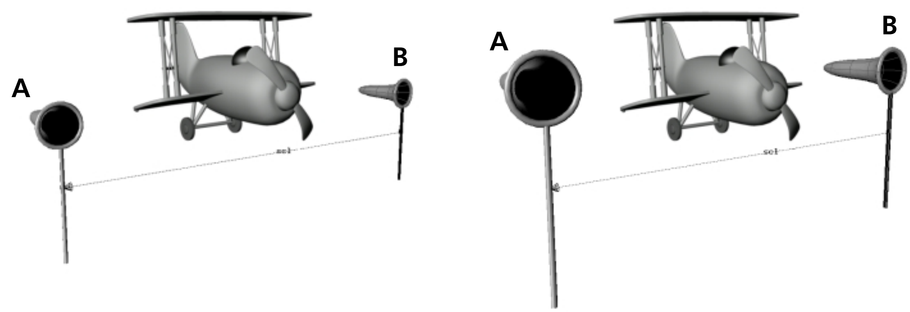
To use the up-vector constraint in its simplest form, follow these steps—this uses a camera as the example object:

1. Select the direction constraint of a camera from the explorer.
2. Choose **Constraint** > **Up Vector** and select the object to which you want to constrain the camera's up vector.

If you wish to remove an up-vector constraint from an object that also has a position constraint applied to it, you must first remove the position constraint from that object.

Scaling Constraints

You can constrain the scaling values of one or more selected objects so that they are the same as the scaling values of a constraining object. This command is useful for scaling all objects in a hierarchy at once and in accordance to the scaling changes applied to one object.



Scaling Constraint
Scaling of windsock A is constrained to scaling of windsock B.

When windsock A is scaled in all axes, windsock B is scaled by the same factor.

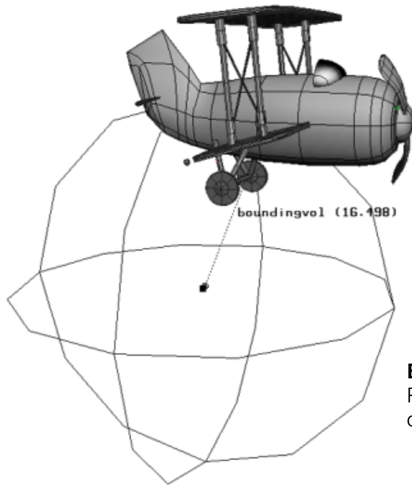
To constrain an object's scaling

1. Select the objects to be constrained.
2. Choose **Constraint** > **Scaling**.
3. Select the object that is to act as a constraint.
4. Select the constraining object and scale as desired.

The objects constrained to it are also scaled, using the same scaling values as the constraining object.

Bounding-Volume Constraints

This type of constraint makes sure that one or more objects remain within, outside, or on the surface of its constraining object's boundaries.



Bounding-Volume Constraint
Plane is constrained to the surface of a sphere's bounding volume.

To use the bounding-volume constraint

The constraining object can be either an implicit cube or an implicit sphere.

1. Select the object to be constrained.
2. Choose **Constraint** > **Bounding Volume**.
3. Select the implicit cube or sphere that is to act as a constraint.
4. Select either a constrained object or the constraining object, and perform a transformation as desired. By default, the constrained object will remain confined within the object's volumetric area.

To modify bounding-volume constraints

1. Select the constrained object, then in the explorer select the object's constraint node.
2. In the **Bounding Volume Constraint** property page, select the bounding type from the Bounding Type drop-down list:
   - **Inside**—the constrained object remains inside the constraining object's boundary limits.
   - **Outside—**the constrained object remains outside the constraining object's boundary limits.

- **On Surface**—the constrained object remains constrained to the constraining object's outer surface.
- **Monitor Only**—the constrained object exhibits no constrain behavior. If Constraint Information has been selected in the viewport's Visibility Setting property editor, only the distance between the centers of the constrained and constraining objects is displayed.

3. In the **Constraint Shape Type** drop-down list, select whether the object is constrained within the bounding volume by its points, by its surface, or by bounding box points/surface.

Chapter 5 Animating with Expressions

Expressions are mathematical formulas that you can use to control any parameter. This includes parameters such as translation, rotation, scaling, material, and texture.

You cannot directly use more than one animation source (for example, keyframes and an expression) on the same parameter. To combine animation sources, you must use the animation mixer as described in *Chapter 11: Actions* on page 253.

An expression is simply a string of characters that may include object and parameter names, mathematical operators, and tokens representing functions or constants. For example, to constrain object A's Y rotation to object B's X translation, you would set an expression on **A.kine.local.roty** consisting of the string **B.kine.local.posx**.

Expressions have the flexibility to let you create almost any connection you like between parameters. You can apply an expression on any parameter that can be animated and use any other parameter as well as predefined variables, standard math functions, random generators, and more.

You'll find expressions to be very powerful. They allow you the freedom to animate using exact specifications.

If you're a novice to math, there are some basic expressions you can perform without having to understand math functions well. In fact, many common expressions contain little or no math. Some of them just make one parameter equal to another to create a constraint.

For example, you can easily create a simple constraint, such as constraining the color of one object to the rotation of another. In this case, all you have to know are the names of the objects and their appropriate parameters.

Once there is an expression on a parameter, you can no longer manipulate that parameter interactively. For example, if you apply an expression on an object's local X position, you can no longer use the Translate tool to move the object in X. Linked parameters are an exception—you can still change the driven parameter, even though there is an **l_fcv**() expression on it.

Resulting Value of an Expression

The value of an expression is always a floating-point value (meaning it can be a number like 12.345 instead of an integer like 12). The expression editor always shows the mathematically correct value. However, some parameters can only accept integers or values in a specific range. In these case, the value returned by the expression is automatically rounded off or clamped before being used.

The SOFTIMAGE|3D function **has_relation**() is not supported.

The modifiers .L and .G are not used to specify local and global transformations. SOFTIMAGE|XSI specifies these parameters differently:

```
sphere.kine.local.posx
sphere.kine.global.posx
```

You cannot combine expressions and function curves directly on the same parameter. To do this, you must use the animation mixer—see *Combining Function Curves and Expressions* on page 287.

SOFTIMAGE|XSI does not support local variables. If you import a scene from SOFTIMAGE|3D containing local variables, they will be expanded. You can use custom parameters to mimic local variables, as described in *Using Custom Parameters as Variables* on page 151.

Opening the Expression Editor & Creating Expressions

There are several ways to display the expression editor. Each way does something slightly different.

Creating a Default Expression

To open the expression editor and create a default expression, use the **Set Expression** command in any one of the following ways:

- In a property editor, right-click on the icon of an animatable parameter and choose **Set Expression**.

  or

- In the explorer, right-click on the icon of an animatable parameter and choose **Set Expression**. Mark a parameter, then choose **Create** > **Parameter** > **Set Expression** on the Animate toolbar. You can set the same expression on multiple parameters by marking more than one.

  or

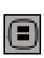
- Mark a parameter, then choose **Animation** > **Set Expression** from the Animation panel. Again, you can set the same expression on multiple parameters by marking more than one.

In each case, the default expression is the parameter's current value.

You cannot create an expression on a parameter that already has other animation, such as a function curve or another expression. You must remove the existing animation first using the Remove Animation command on the Animation menu. Alternatively, you can combine expressions and other animation using the animation mixer—see *Combining Function Curves and Expressions* on page 287.

Editing an Existing Expression


Constant value


Equal to another parameter


Any other expression

To open the expression editor and edit an existing expression, use the **Edit Expression** command in any one of the following ways:

- In a property editor, right-click on the icon of an parameter with an expression and choose **Edit Expression**.

  or

- In the explorer, right-click on the icon of an parameter with an expression and choose **Edit Expression**.

  or

- Mark a parameter, then choose **Create** > **Parameter** > **Edit Expression** on the Animate toolbar.

  or

- Mark a parameter, then choose **Animation** > **Edit Expression** from the Animation panel.

If the parameter does not already have an expression, it is set as the target parameter but no default expression is created.

You can also open and exit an existing expression from other views:

- In a 3D view, make sure that **Show** > **Relations** is on. If the selected object has an expression relating it to another object, they are joined by a white dotted line labeled **Expr.** Click the line to see a list of expressions; select an expression from the list to edit it.
- In the Schematic view, make sure that **Show** > **Expression Links** is on. If there is an expression connecting two objects, it is shown as a green line labeled **Expr.** Select the line to see a list of expressions; select an expression from the list to edit it.

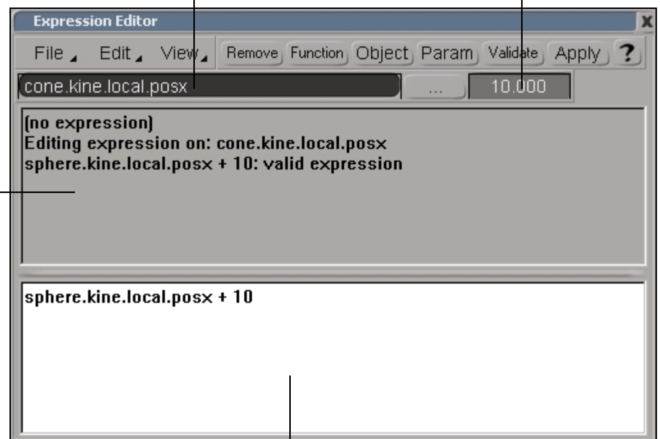
Displaying the Expression Editor

To simply open the expression editor without specifying a target parameter or setting a default expression, choose **View** > **Views** > **Expression Editor**. You can then select a target as described in *Selecting the Target* on page 140.

Overview of the Expression Editor

The target shows which parameter is controlled by the expression.

The number box shows the value of the expression at the current frame



The message pane updates as you work, letting you know whether the expression is valid. Press Ctrl-g to switch between this pane and a graph of the resulting expression values.

The expression pane is a text editor where you enter an expression by typing or choosing items from the menus.

Writing Expressions

To write an expression:

1. Select the target parameter. This step is only necessary if you opened the expression editor from the **View** menu.
2. Enter the expression. For a description of expression syntax, see page 142.
3. Validate and apply the expression.

You can preview the result of an expression as a graph. To switch between displaying the message pane and the graph view, press **Ctrl+g** or choose **View Graph** from the expression editor's **View** menu.

Selecting the Target

If you opened the expression editor with the **View** > **Views** > **Expression Editor** command on the main-menu bar, you first need to select a target parameter (also called the *affected element*).

If you opened the expression editor using the **Set Expression** or **Edit Expression** commands on the Animate toolbar or Animation menu, the target parameter is already selected (whatever parameter was marked when you chose the command). However, you can change the target as described below.

To select or change the target parameter

1. Do any one of the following:

Browse

   - Click the Browse (**...**) button.

   or

   - Choose **Edit** > **Change Target** from the expression editor's command bar.

   or

   - Press **Ctrl+t**.

   A pop-up explorer opens.

2. Navigate the pop-up explorer by expanding nodes until you see the desired parameter, then click on its name.

Alternatively, if you know the scripting name of the parameter, you can type it directly into the target parameter box. To determine the scripting name of a parameter, modify the parameter in a property editor and check the name that gets logged in the SetValue command in the Command box or Script Editor. For more information about commands and scripts in general, see *Chapter 7: Commands & Scripts* in the *Fundamentals* guide.

Entering Expressions

You can enter an expression by typing in the expression pane. In addition, you can:

- Use the **Function** button or press **Ctrl+f** to add a function or constant at the insertion point.
- Use the **Object** button to add an object or parameter name at the insertion point.
- If you have already typed part of a branch name followed by a period, use the **Param** button to choose a child node to add at the insertion point.
- On Windows NT, you can cut, copy, and paste as well as drag and drop text from another editor such as Notepad.

Entering Functions & Constants

SOFTIMAGE|XSI supports several tokens representing functions and constants in expressions. You can enter these directly by typing in the editing pane, or you can select them using the **Function** button or by pressing **Ctrl+f.**

For some items you choose, only the first part of the string is entered. For example if you choose **Function** > **Trigonometry** > **Sine**, the following is entered:

```
sin(
```

You can then type any expression after it, then type a closing parenthesis:

```
sin(Fc * 10 + 5)
```

For other items you choose, dummy strings are entered to help you remember which parameters are required. You must replace the dummy strings with expressions. For example, if you choose **Function** > **Condition**, the following is entered:

```
cond( <cond> , <true_expr> , <false_expr> )
```

Replace <**cond**> with a condition to evaluate as true or false, for example:

```
(cone.kine.global.posx <= 0)
```

Replace <**true_expr**> with an expression to use if the condition is true, for example:

```
0
```

Similarly, replace <**false_expr**> with an expression to evaluate if the condition is false.

For a complete description of all the functions and constants available, see Online Help.

If you are using the camera distance functions To Camera(**ctr_dist_cam**) or To Camera Interest (**ctr_dist_cam_int**) and you move the camera or interest, you may need to change the current frame to update the scene.

Entering Object Names

You can enter object names by typing directly in the editing pane or clicking the **Object** button and selecting one from a pop-up explorer. You can expand nodes to select children, properties, and parameters.

You can use the **this** token in an expression to have an object reference itself. This is useful when you to use the same expression on many different objects. For example, the following always refers to the local Y position of the object whose parameter is the target:

```
this.kine.local.posy
```

Entering Parameter Names

You can also enter parameter names by typing directly, or use the **Param** button or **F12** key. These commands prompt you with a list of possible parameters in context. For example:

1. Enter the following:

   ```
   Camera_Root.
   ```

2. Press F12. A list of possible nodes appears; choose **kine.**
3. Type **l** (lowercase *L*) and press F12 again. The node **local** is automatically filled in (it's the only available node beginning with **l**).
4. Type **p** and press F12 one last time. A list of available nodes that begin with **p** appears; choose **posx.**

Expression Syntax

Here are a few things to remember when creating expressions:

- All object names must be followed by a parameter scripting name.

  The exceptions are the distance functions: Center to Center (**ctr_dist**), to Camera(**ctr_dist_cam**), and to Camera Interest (**ctr_dist_cam_int**). When you specify an object name with these functions, remember to use a period at the end of the name.
- Parentheses must always be closed.
- Spacing is irrelevant to how the expression is parsed. Spaces, tabs, and carriage returns may make your expression easier for humans to read, but they are ignored by SOFTIMAGE|XSI.

Operators

You can use the following operators in an expression:

Operator	Meaning
*	multiply
+	add
–	subtract
/	divide
%	modulo (remainder)

Operators follow the standard of precedence: First, items enclosed in brackets, then * and /, followed by + and –.

You can use the following Boolean operators in a condition:

Operator	Meaning
=	equals
>	greater than
=>	greater than or equal to
<	less than
<=	less than or equal to
!=	not equal to
*	Boolean AND
+	Boolean OR

Using Another Text Editor

You can use a different text editor to edit expressions. When you choose **View** > **Text Editor** or press **Ctrl+e**, the current contents of the editing pane open in a separate text-editor window (Notepad on Windows NT; jot on IRIX). You can type the expression; when you close the text editor, you are asked whether you want to update the editing pane of the expression editor.

Validating & Applying Expressions

After you have entered an expression, you can validate and apply it.

Validating Expressions

Validation checks whether the contents of the editing pane are a proper expression. It provides a way of checking your expression's syntax without applying it.

To validate your expression, click the **Validate** button on the expression editor's command bar. A message appears in the message pane, informing you whether the expression is valid or not.

Applying Expression

Applying an expression attaches it to the target parameters. Changes you make in the editing pane do not take effect until you apply the expression.

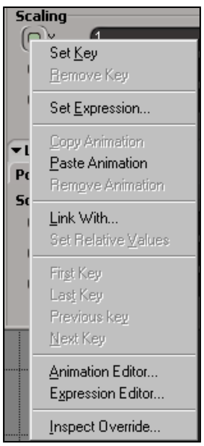
Once you have entered an expression in the bottom pane, click **Apply**. The expression is automatically validated first; if it is not valid, a message informs you and the expression is not applied.

Copying Expressions

Expressions can be duplicated between different parameters for the same object or for different objects. For example, you can copy an expression from the X rotation of one object and paste it onto the Y scaling of another object.

The **this** token is useful when using the same expression on many different objects. It always stands for the object with the expression on it. For example, the following always refers to the local Y position of the object whose parameter is the target:

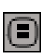
```
this.kine.local.posy
```



To copy expressions in a property page

1. Open the property page containing the animated parameter you want to copy.
2. Right-click on the parameter's animation icon and choose **Copy Animation** from the pop-up menu.
3. If necessary, open the property page for the parameter to which you want to copy the function curve.
4. Right-click on the parameter's animation icon and choose **Paste Animation** from the pop-up menu.


Constant value


Equal to another parameter


Any other expression

To copy expressions from an explorer window

1. Open an explorer window. If necessary, choose **Show** > **All Nodes** from the explorer menu to display animatable properties.
2. Expand the nodes until the animated parameter you want to copy is displayed.
3. Right-click on the parameter's animation icon and choose **Copy Animation** from the pop-up menu.
4. Expand the nodes until the animated parameter you want to copy to is displayed.
5. Right-click on the parameter's animation icon and choose **Paste Animation** from the pop-up menu.

You can also copy animation in the scene explorer by dragging one animated parameter node on top of another parameter.

Removing Expressions


Constant value


Equal to another parameter


Any other expression

You can remove expressions using the expression editor, the Animation menu, the explorer, or a property editor.

To remove expressions using the expression editor

To remove the expressions from the target parameters in the expression editor, click **Remove** on the command bar.

To remove expressions using the Animation menu

To remove an expression from a parameter, mark it and choose **Animation** > **Remove Animation.**

To remove expressions using the explorer or a property editor

To remove an expression from a parameter in the explorer or on a property editor, right-click on the parameter's animation icon and choose **Remove Animation.**

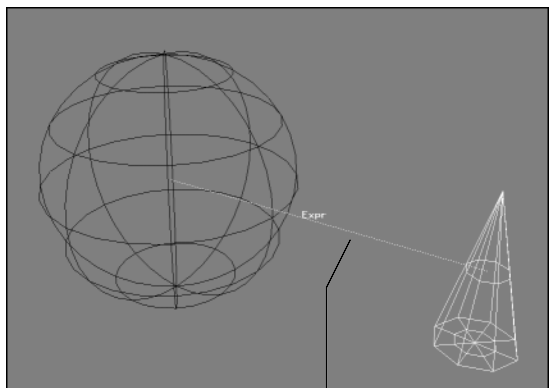
Viewing Expression Information

You can identify the objects that have expressions on their parameters in the 3D views as well as in the schematic view.

Expression Information in 3D Views

To see expression information in a 3D view, make sure that **Show** > **Relations** is on.

- If the selected object has parameters with expressions on its transformations, **Expr** is displayed in white.
- If the selected object has parameters with expressions that involve other objects, those objects are linked to the selected object by dotted white lines. You can click on a dotted line to see a list of expressions that involve the linked object. You can also click on one of the expressions in the list to open it in the expression editor.



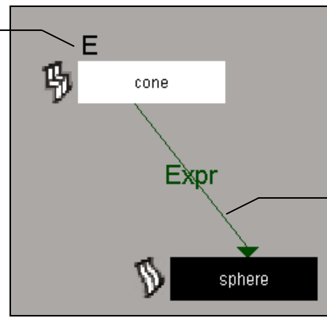
Dotted line indicates an expression involving the other object. Click to see a list of expressions, then select an expression from the list to edit it.

Expression Information in the Schematic View

In the schematic view:

- If an object has parameters with expressions on its transformations, the letter **E** is displayed above it.
- If **Show** > **Expressions** is on, an object with an expression is linked to objects whose parameters are involved by solid green arrows labelled **Expr**. You can select these arrows by clicking and dragging slightly—a list of expressions appear. Click on an expression in the list to open it in the expression editor.

Label indicates an expression on the cone.



Arrow indicates an expression involving the sphere. Select the arrow for a list of expressions, then select an expression in the list to edit it.

Saving and Loading Expressions

You can save the current contents of the expression editing pane as a file then load it back later on other parameters. Expressions are saved as plain text files with an extension of **expr2**. By default, they are saved in the Expressions subdirectory of the current project.

Expressions are not validated when saving—the contents of the editing pane are written as is to file. Similarly, expressions are not validated when opening—the contents of the editing pane are simply replaced with the contents of the .expr2 file.

Saving Expression Files

To save the contents of the editing pane, choose one of the following commands from the expression editor's **File** menu:

- **Save.** If you have already opened or saved an expression file, it is overwritten on disk with the current contents of the editing pane. Otherwise, you are prompted for a file name as with Save As.
- **Save As.** A browser opens, prompting you for a file name and directory.

Opening Expression Files

To open a saved expression file, choose **Open** from the expression editor's **File** menu then use the browser to select the desired .expr2 file.

You cannot open a binary **.exp** format expression file from SOFTIMAGE|3D directly. Instead, import a scene or model with the expression in it.

Converting Expressions to Function Curves

You can convert an expression to a raw function curve file (**.fraw2** format). SOFTIMAGE|XSI plots the value of the expression and saves a function curve file with a key at every frame. You can then open the saved .fraw2 file in the animation editor. The original expression applied on the selected parameter is left unchanged

To convert an expression to a function-curve file

1. Make sure that you have applied the expression. SOFTIMAGE|XSI will only plot an expression if it has been applied; it will not plot the contents of the expression editing pane.
2. In the expression editor, choose **File** > **Save as FCurve**. A browser opens.
3. Use the browser to select a directory and file name, then click OK.

For information about opening function curves in the animation editor, see *Editing Function-Curve Animation Using the Animation Editor* on page 62.

Tips & Tricks with Expressions

This section provides some information about using expressions in specific situations.

Using Expressions on Components

You cannot use expressions to directly manipulate components like point, polygons, edges, or subsurfaces. However, you can create a cluster, assign the cluster center to a null or other object, and use expressions on the object to manipulate the components indirectly. For more information about cluster centers, see *Deforming by Cluster Centers* in Chapter 8 of the *Modeling & Deformations* guide.

Using Expressions with Velocity and Acceleration

Normally, velocity and acceleration values are not computed for objects. However, you can force these values to be computed by using a Quickstretch deformation. You can then mute the Quickstretch operator so that the object is not deformed but its velocity and acceleration values are available for use in expressions. For more information about Quickstretch in general, see *Chapter 12: Quickstretch* of the *Modeling & Deformations* guide.

Using Expressions on Preferred Angles

If you set an expression on the preferred angle of a bone, the expression's values will be overridden if you are using the SOFTIMAGE|3D solver. For more information, see *Inverse-Kinematics Behavior* on page 191.

Using Custom Parameters as Variables

If you find yourself using the same sub-expressions over and over, you can create a custom parameter to hold the sub-expression. You can change the custom parameter to update all expressions that reference it.

Chapter 6 Linked Parameters

Linked parameters are parameters that are connected in a way defined by a function curve. They are similar to expressions but the value of one parameter drives the other in a way dictated by the curve, rather than by a mathematical formula.

First you create a link relationship between the parameters, then you set relative values. You can also edit the function curve directly.

For a brief tutorial on linked parameters, see *Tutorial 7: Linked Parameters* in the *Tutorials* guide.

Linked parameters actually use a special type of expression: **l_fcv**(). For more information about expressions in general, see *Chapter 5: Animating with Expressions* on page 133.

Linking Parameters

When you link a parameter to another, you specify that its value is to be controlled by another parameter.

To link one parameter to another

1. Do one of the following:

   - Mark the parameters that you want to be controlled by another and choose **Link With** from the Animation menu on the Animation panel.

   or

   - In the explorer, right-click on the parameter that you want to be controlled by another and choose **Link With**.

   or

   - In a property editor, right-click on the animation icon of the parameter that you want to be controlled by another and choose **Link With**.

   A pop-up explorer opens.

2. Navigate through the pop-up explorer by expanding nodes until you see the parameter that you want to act as a controller and click on its name (not its icon).

Linked parameter

   A link relationship is established. You can now set relative values—"keys" that define the relationship between the parameters—or edit the linking function curve in the animation editor.

Setting Relative Values

Relative values are keys that map a particular value of the controlling parameter to a particular value of the controlled parameter, in the same way that an animation key maps a particular frame to a particular value.

To set relative values

1. First set a value for the controlling parameter. This is the equivalent of first setting the frame for an animation key.
2. Set a value for the controlled parameter.

You can lock a property editor in order to keep more than one open at the same time.

3. Right-click on the animation icon of the controlled parameter, or right-click on the controlled parameter in the explorer, and choose **Set Relative Values**.

Editing the Link Function Curve

You can use the animation editor to modify the function curve that maps the values of the linked parameters. Right-click on the animation icon of the controlled parameter in a property editor or the explorer, and choose **Animation Editor**. The animation editor opens with the linking function curve loaded. For information about using the animation editor to edit function curves, see *Editing Function-Curve Animation Using the Animation Editor* on page 62.

You cannot open the linked-parameter function curves using the Animation menu below the timeline.

When editing the linking curve, remember that the X axis does not represent time—it represents the controlling parameter.

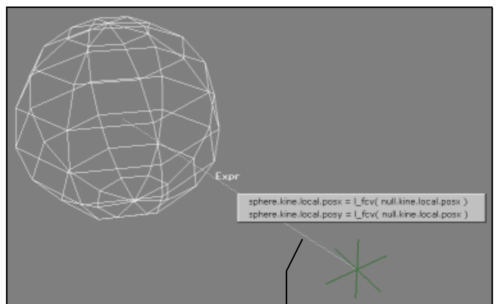
You can edit only one link function curve at a time in the animation editor.

Viewing Link Information

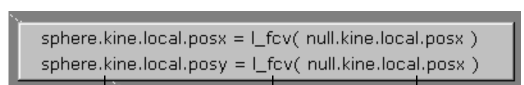
Because linked parameters are a special type of expression, you can identify the objects that have linked parameters in the 3D views as well as in the schematic view.

Link Information in 3D Views

To see link information in a 3D view, make sure that **Show** > **Relations** is on. If the selected object has a parameters on its transformations that are linked to parameters on other objects, those objects are linked to the selected object by dotted white lines. You can click on a dotted line to see a list of expressions that involve the linked object. You can identify the linked parameters by the **l_fcv**() expression.



Dotted line indicates an expression involving the other object. Click to see a list of expressions.



l_fcv indicates a linked parameter.

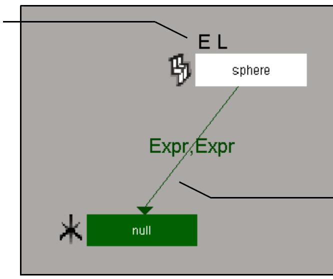
Controlled parameter is on the left.

Controlling parameter is on the right.

Link Information in the Schematic View

You can also see information about linked parameters in the schematic view. As is the case with all expressions, if an object has linked parameters on its transformations, the letter **E** is displayed above it; and, in addition, for linked parameters there is also the label **L**. If **Show** > **Expressions** is on, an object with an expression is linked to objects whose parameters are involved by solid-green arrows labelled **Expr**. You can select these arrows by clicking and dragging slightly—a list of expressions appear. Once again, **l_fcv**() indicates a linked parameter.

Labels indicates an expression and a linked parameter on the cone.

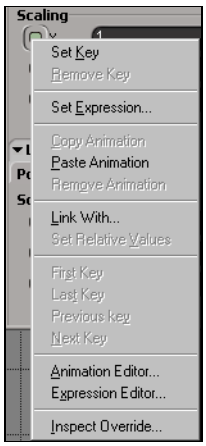


Arrow indicates an expression involving the sphere. Select the arrow for a list of expressions.

Copying Link Animation

Link animation can be duplicated between different parameters for the same object or for different objects. For example, you could copy link animation of the X rotation of one object and paste it onto the Y scaling of another object.

When you duplicate link animation, the function curves are copied but not shared. If you change the relative values of one parameter, the other parameters are not affected.



To copy animated parameters in a property page

1. Open the property page containing the animated parameter you want to copy.
2. Right-click on the parameter's animation icon and choose the **Copy Animation** command from the pop-up menu.
3. If necessary, open the property page for the parameter to which you want to copy the function curve.
4. Right-click on the parameter's animation icon and choose the **Paste Animation** command from the pop-up menu.

To copy animated parameters from an explorer window


Linked parameter

1. Open an explorer window. If necessary, choose **Show** > **All Nodes** from the explorer menu to display animatable properties.
2. Expand the nodes until the animated parameter you want to copy is displayed.
3. Right-click on the parameter's animation icon and choose the **Copy Animation** command from the pop-up menu.
4. Expand the nodes until the animated parameter you want to copy to is displayed.
5. Right-click on the parameter's animation icon and choose the **Paste Animation** command from the pop-up menu.

You can also copy animation in the scene explorer by dragging one animated parameter node on top of another parameter.

Removing Link Animation

You can remove link animation using the Animation menu, the explorer, or a property editor.

To remove link animation using the animation menu

To remove link animation from a parameter, mark it and choose **Animation** > **Remove Animation**.

To remove link animation using the explorer or a property editor


Linked parameter

To remove link animation from a parameter in the explorer or on a property editor, right-click on the parameter's animation icon and choose **Remove Animation.**

Tips & Tricks with Linked Parameters

This section provides some information about using linked parameters in specific situations.

Using Linked Parameters on Components

You cannot use linked parameters to directly manipulate components like point, polygons, edges, or subsurfaces. However, you can create a cluster, assign the cluster center to a null or other object, and use linked parameters with the object to manipulate the components indirectly. For more information about cluster centers, see *Deforming by Cluster Centers* in Chapter 8 of the *Modeling & Deformations* guide.

Using Linked Parameters on Preferred Angles

If you use a linked parameter to control the preferred angle of a bone, the values will be overridden if you are using the SOFTIMAGE|3D solver. For more information, see *Inverse-Kinematics Behavior* on page 191.

Chapter 7 Custom Parameters

Custom parameters are parameters that you create for any purpose you want. You typically create a custom parameter then connect it to other parameters using expressions or linked parameters. You can then use the sliders on the custom parameter set's property editor to drive the connected parameters in your scene.

For example, you can use a set of sliders on a property editor to drive the pose of a character, instead of creating a virtual control panel using 3D objects.

Defining Custom-Parameter Sets

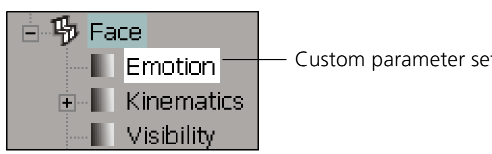
You can organize your custom parameters into sets with meaningful names. For example, you can have one custom-parameter set for facial expressions, another for skeleton poses, and so on. You can have as many custom parameter sets as you like for any object.

Creating Custom Parameter Sets

To create a custom parameter set

1. Select the object for which you want to create a parameter. If you want to propagate the custom-parameter set to the object's children, select it in branch mode.
2. Choose **Create** > **Parameter** > **Custom Parameter Set** from the **Animate** toolbar. The custom parameter set's property editor opens, and a node is created under the object in the explorer.
3. Enter a **Name** for your custom parameter set.

In the explorer, custom parameters are displayed similarly to other property sets.



Renaming Custom Parameter Sets

You can rename a custom-parameter set after you have created it. In the explorer, make sure that **Show** > **Properties** is on, then right-click on the custom-parameter set and choose **Rename**.

Defining Custom Parameters

Custom parameters are parameters that you create to hold data you define. You can specify the data type, the default value, and other options. Once you have created a custom parameter, you can connect it to other parameters in your scene to create custom controls.

To create a custom parameter

1. Select an object or a custom parameter set.
   - If the object has no custom parameter set, a new one is created named CustomPset by default.
   - If your object has multiple custom parameter sets, select the parameter set to which you want to add the parameter instead of selecting the object. If you select the object, no parameter will be created.
2. Choose **Create** > **Parameter** > **Custom Parameter** from the Animate toolbar. The New Custom Parameter dialog box opens.
3. Set the options as desired. For information, click the help icon.

You cannot change any of these options after you have created a custom parameter.

Parameter names must be unique. If there is already a parameter with the same name in the same custom parameter set, no new parameter will be created.

4. Click OK. A new custom parameter is created.

Use text parameters to store annotations.

Connecting Custom Parameters

Typically, an unconnected custom parameter is not very useful. By connecting it to other parameters, you can control various properties in your scene. For example, you can link custom parameters to mixer weight values of shapes or skeleton poses then use the custom parameters to quickly blend them.

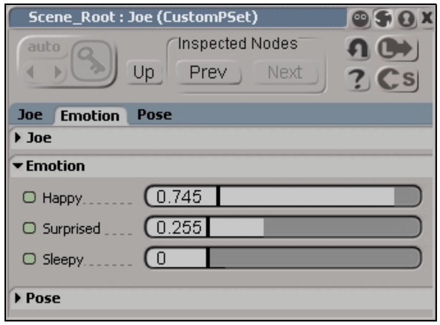
You can connect a custom parameter in several ways:

- **Using linked parameters.** Linking maps one parameter to another in a non-linear way using a function curve. For more information, see *Chapter 6: Linked Parameters* on page 153.
- **Using expressions.** Create an expression for a parameter in your scene that references your custom parameter. Custom parameters use the same naming conventions as other parameters; for example, if you have a custom parameter called **Happy** in a set called **Emotion** on an object called **Face**, you would use the string **Face.Emotion.Happy** to refer to it in expressions. For more information about expressions in general, see *Chapter 5: Animating with Expressions* on page 133.
- **Using scripts.** The scripting name of a custom parameter is the same as its name in expressions. For more information about scripting in general, see *Chapter 7: Commands & Scripts* in the *Fundamentals* guide.

You can leave a custom parameter unconnected and use it as a global variable to share data between scripts.

Displaying & Editing Custom Parameters

You can modify the values of custom parameters in the same way as any other parameter in SOFTIMAGE|XSI. For example, you can set values with the sliders, save keys, apply expressions, store actions, plot, and so on. Each set of custom parameters can be displayed on its own property page.



To display custom parameters

Select the object and do one of the following:

- Choose **Create >Parameter >Edit Custom Parameter Sets** from the Animate toolbar. A property editor opens with pages for the object's name as well as all custom parameter sets.

  or

- Choose **Edit** > **Custom Properties** from the Edit panel. Again, a property editor opens with pages for the object's name as well as all custom parameter sets.

  or

- Click on the icon of a custom-parameter set in an explorer. In this case, a property editor opens with only one page for the chosen custom parameter set.

Copying Custom-Parameter Sets

You can copy a custom-parameter set using the commands on the Edit menu or by Ctrl+dragging in an explorer.

When you copy a custom parameter set, its current parameters and values are duplicated on another object. You can then edit the values of the two parameter sets independently of each other. The two parameter sets do not share a data structure; if you change the name of one custom-parameter set or add new parameters, the other set is not affected.

To copy a custom-parameter set using Edit commands

1. Select the custom-parameter set by clicking on its name in an explorer.
2. Choose **Edit** > **Copy** from the main-menu bar.
3. Select the object to which you want to apply the custom parameter set. To apply the custom parameter in branch mode, select the object with the middle mouse button.
4. Choose **Edit** > **Paste** from the main-menu bar.

To copy a custom-parameter set by Ctrl+dragging

1. Select the custom-parameter set by clicking on its name in an explorer.
2. Hold the Ctrl key down while you click and drag the custom-parameter set onto another object. To apply the custom parameter in branch mode, drag with the middle mouse button.

Deleting Custom Parameters & Sets

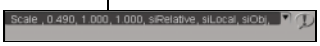
You can delete custom parameters individually, as well as delete entire custom parameter sets.

To delete animatable custom parameters

Mark the custom parameters to delete, then choose **Create** > **Parameter** > **Remove Custom Parameter** from the Animate toolbar.

To delete read-only and non-animatable custom parameters

Command box



Use the **RemoveCustomParam**() command in the command box or script editor. For example:

```
RemoveCustomParam(Joe.Emotion.Angry)
```

To delete a custom parameter set

Select it in an explorer and press the Delete key. The set and all custom parameters in it are deleted.

Section II • Character Animation

Chapter 8 Skeletons & Inverse Kinematics

This chapter describes how to create skeletons for your models. A skeleton provides a framework for a model and can be used to pose or deform it intuitively.

The human skeleton provides a framework for the body. The structure of your skeleton determines every aspect of how you move, from picking up a pencil to running. Similarly, a SOFTIMAGE|XSI skeleton provides an interior framework for a model, and how you build a skeleton will determine how your model will move when animated.

The Purpose of Skeletons

Skeletons provide an intuitive way to pose and animate your model. Most models are static objects, and intricate movement of individual parts can be time consuming. A well-constructed skeleton can be used for a wide variety of poses and actions, in much the same way as the all-purpose skeletons in our bodies can. On the other hand, a skeleton might also be used to provide *less* flexibility—to ensure that a model cannot move beyond certain restricted conditions. Look at your wrist: you can easily bend it straight up or down, but side-to-side motion is very limited.

Like real skeletons, skeletons in SOFTIMAGE|XSI are usually made up of *articulated chains*, which are made up of *bones,* which are connected by rotating *joints*. You can animate chains using conventional keyframing of transformations or by constraining a chain to another animated object that you transform. You can also animate chains by using animation data from a motion capture session.

How parts of the skeleton move relative to each other is determined by the way your skeleton hierarchy is built, whether and how objects are constrained to each other, and if you have created any expressions to govern their motion.

Skeletons are usually covered with an enveloping model or object (generically referred to as *envelopes*), which can be deformed (stretched, twisted, or compressed), depending on how the skeleton components are moved, rotated, or scaled. This can be used to simulate the muscles in flexing biceps or facial expressions.

How Skeletons Work

Skeletons are built using objects, chains, or a combination of both. The objects and chains of a skeleton are usually connected in a hierarchy. For instance, the hips of your model would be connected to the spine, which would in turn be connected to the shoulders. The shoulders would then be connected to the arms, each of which has an elbow and a wrist.

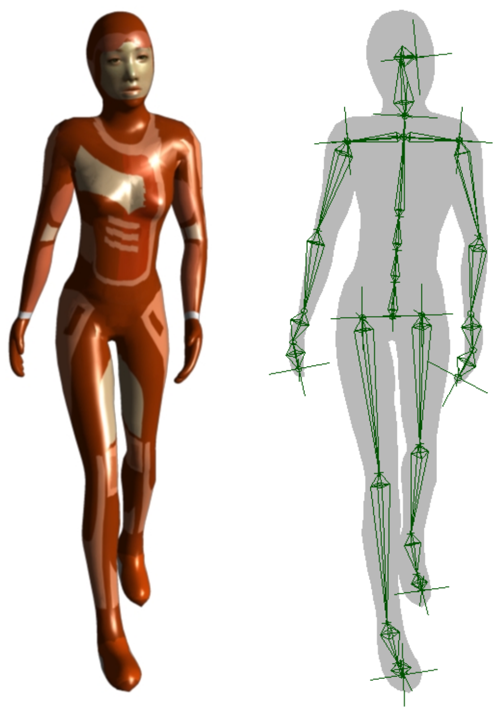
Although it is possible to build an enveloping model around a skeleton, it is more common (and usually more practical) to build or place a skeleton inside a finished model. The structure of a skeleton determines how you can manipulate the envelope. Objects in skeletons can be rotated or positioned to deform envelopes, while chains are used for positioning different parts of envelopes.

Articulated Chains and Kinematics

Chains are manipulated using *inverse kinematics* (IK) and *forward kinematics* (FK). A chain can use either IK or FK at any time. The method used depends on what part of the chain you manipulate.

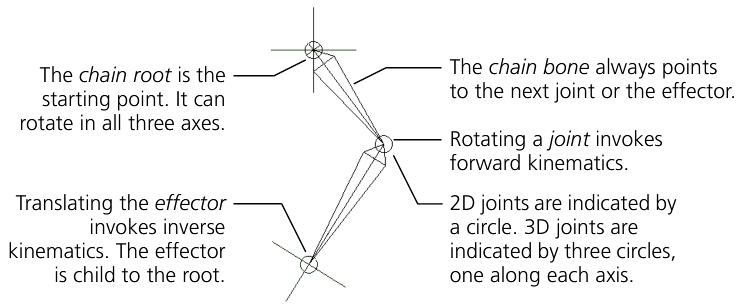
IK is invoked by translating a bone or the end of a chain. SOFTIMAGE|XSI calculates the angles at which the previous joints in the chain must rotate in order for the bone or end of the chain to reach its goal. This process is called *solving*.

FK is invoked by rotating a joint in the chain, allowing for complete control of the chain's behavior. Only the angle of the selected joint is affected; all other joint angles are preserved. With FK, positioning a skeleton's foot means rotating each joint in the leg, from the hip to the ankle. This method is more tedious to execute but the chain bends exactly as intended. Using FK allows you to create many types of movements that may not be possible to animate with IK alone.



Introduction to Articulated Chains

A chain is a series of bones connected by articulated joints. A chain has three basic components: a root, bones, and an effector.



There is no limit to the number of bones a chain can have, but there must be at least one bone in a chain.

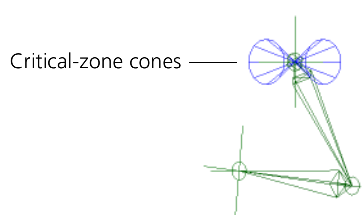
Chains can be 2D or 3D. When using IK, a 2D chain's joints can only rotate on the Z axis of the root, like hinge joints. A 3D chain's joints can rotate freely on any axis, like ball joints.

The first joint connecting the root to the first bone on either a 2D or 3D chain is always a ball joint, and it can rotate on any axis.

Chain Root

The chain root is a null that acts as the starting point on the chain. Because the first joint is local to the root, the root's position and rotation determine the position and rotation of the rest of the chain. In a 2D chain, the entire chain extends on the root's XY plane, called the *plane of resolution*. A chain root can be the child of another chain, root, or effector.

Around the root is an area called the *critical zone*. If the effector is translated into the critical zone, the general orientation of the chain will flip. To visualize the critical zone, choose **Show** > **Visibility Options** > **Attributes** from the viewport and select **Chain Critical Zone** in the Selected Objects and Unselected Objects sections.



Chain Bone

The chain bone is a straight line that connects roots, joints, and effectors. A bone is always rotated about its joint, which is at the end of the previous bone. The first bone rotates around the root.

The first bone in the chain is a child to the chain root. All other bones are children to the previous bones in the chain.

The first bone in the chain is considered to be position constrained to the root. As a result, the bone's position constraint to another object will be superseded by its constraint to the root.

Joint

A joint is the connection between two elements in a chain: there is a joint between the root and the first bone, between bones in the chain, and between the last bone and the effector.

A joint can be either 2D or 3D. In a 2D chain, only the first bone can be rotated with three degrees of freedom using IK. Its joint, the chain root, acts as a ball joint. All other joints in a 2D chain act like hinge joints. A 2D joint is indicated on a chain by a circle. The angle of the circle indicates the plane along which the chain will unfold, or the *plane of resolution*.

A 3D chain's joints all act like ball joints; they can rotate with three degrees of freedom. A 3D joint is indicated by three circles along the joint's X, Y, and Z axes.

Effector

The effector is the last point of a chain. Moving the effector invokes IK, which modifies the angles of all the joints in the chain between the root and the effector.

The effector is the child of the chain root. This allows you to create local animation on the effector that can be translated with the root. However, you can place the effector anywhere in the hierarchy or even make it the child of an object outside of the chain.

Chains loaded from a SOFTIMAGE|3D scene will retain their original hierarchical structure. The effectors' orientation and position will also remain global, not local.

Creating Articulated Chains

To draw a chain, choose **Create** > **Skeleton** > **Draw 2D Chain** or **Draw 3D Chain** from either the Model or Animate toolbars. The chain is defined by a series of points you create in a viewport. The first point defined is the root, the last point is the effector, and all the points in between are joints.

To create an articulated chain

1. Choose **Create** > **Skeleton** > **Draw 2D Chain** or **Draw 3D Chain** from either the Model or Animate toolbars.

2. To create the first joint, click in any window.

The chain root is created. If the chain is 2D, its local Z axis will point “out” from the window; otherwise, the root will be aligned with the scene root.

3. Move the pointer to a new position and click again.

The first chain bone is created.

- To add joints, continue clicking to create as many joints as desired.
- To start a new chain, middle-click to end construction of the current chain, then move your pointer to where you want to start the new chain. Repeat steps 2 to 4.
- To end the drawing mode, right-click.

When placing a root, joint, or effector, SOFTIMAGE|XSI does not place the point until you release the left mouse button. If you hold the mouse button down while moving the point, you can see exactly how the chain will look.

When a 2D chain is created, it is created in the local XY plane of the root and its joints rotate about their local Z axes.

A bone’s local X axis always points along the bone—that is, in the direction of the next joint (or, if it is the last bone, the effector).

Using Single-Bone Chains

You can also use single-bone chains by constraining them to other objects in your scene, such as constraining the effector to point in a particular direction. This allows you to benefit from the control the chain provides. For example, you could use single-bone chains for creating eye movement. To move an eye in the direction of an object it’s looking at, constrain the eye’s pupil to the single-bone chain and then constrain the chain to the object at which the eye is looking.

For more information about constraints, see *Chapter 4: Animating with Constraints*.

Preferred Angles

When you draw a chain, the angle of each joint is to be considered to be its "natural" state, called the *preferred angle*. A chain's preferred angles determine how it will bend, because when you move the effector SOFTIMAGE|XSI tries to preserve the preferred angles as much as possible.

Bones are prevented from rotating backward against their joints' preferred angles when using IK. This keeps a character's elbow, for instance, from bending unnaturally.

Because preferred angles allow you to predict how a chain will bend, draw chain bones as close to their "natural" angles as possible. Drawing bones in a straight line can result in unpredictable bending.

Editing Bones and Joints

You may need to edit a chain to achieve the exact structure you are trying to create. For instance, you may find that your skeleton's forearms are too long for your envelope, or a joint's preferred angle may need adjusting.

Resizing Bones

Choosing the **Property** > **Chain Bone** property from the Selection panel opens the Chain Bone property editor for the selected bone. You can change the length of the bone interactively by dragging the slider or entering a value directly in the Length text box.

Resizing the bone using the Chain Bone property editor preserves the local coordinate system of its children in the chain. This is especially important if you are modifying a skeleton for use with a different envelope.

Changing Preferred Angles

When using 2D inverse-kinematic chains, only the preferred Z angle of a joint is used (though the preferred X angle has some meaning, as described in *Changing the Joint's Preferred Angle* on page 200). Each joint is constrained to remain on a single plane and can rotate only around its local Z axis when the effector is moved.

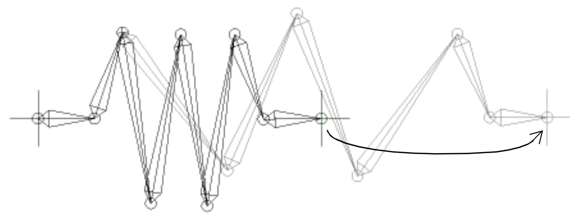
When using 3D inverse kinematic chains, the preferred X, Y, and Z angles of the joints are used, since 3D joints work like ball joints.

To change the preferred angles

1. Select a chain bone.
2. Choose **Property** > **Kinematic Joint** to open the Kinematic Joint property editor for the selected bone.
3. Set the preferred angles using the **X**, **Y**, and **Z** sliders.

Creating Subchains

A chain can be divided into several subchains, each of which can be 2D or 3D, independent of the rest of the chain. When a chain is divided into subchains, IK will only work up to the subchain root.



This chain has a 2D subchain that starts at the second joint after the root. When the effector is translated, the first two bones don't move, as IK only works up to the subchain root.

You can create a subchain by explicitly defining a joint's chain behavior. By default, every joint is implicitly defined as 2D or 3D, based on the previous joint's behavior.

The first joint of each subchain is always a 3D joint, regardless of the subchain's behavior.

To create a subchain

1. Select the bone with which to start the subchain.
2. Choose **Property** > **Kinematic Joint** to open the Kinematic Joint property editor for the selected bone. Change the **Chain/Subchain Type** setting to **2D** or **3D**.

You can display the Kinematic Joint property editor for a 2D joint by clicking on the joint (the joint and bone will turn gray) and pressing Enter.

The new subchain will be defined from the selected bone to the effector or to the next defined subchain, if any. Each subchain will have its own plane of resolution.

To remove a subchain

1. Select the first bone in the subchain.
2. Choose **Property** > **Kinematic Joint** to open the Kinematic Joint property editor for the selected bone. Change the **Chain/Subchain Type** setting to **Default**.

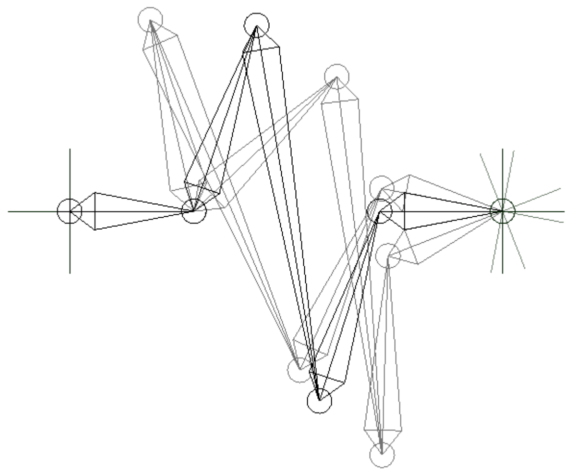
The joints in the subchain will become part of the previous subchain.

Pseudo-roots

You can also create a subchain by making the joint of the selected bone or effector a pseudo-root. Pseudo-roots have the added feature of remaining fixed in local space when IK or FK are invoked. Only rotating the root can affect a pseudo-root's position.

You can also move the pseudo-root by position constraining its bone to another object and translating the object. For more information on constraints, see *Chapter 4: Animating with Constraints*.

To create a pseudo-root, select a chain bone and open its Kinematic Joint property editor. Select **Joint is Pseudo-Root**. If desired, change the **Chain/ Subchain Type** setting to **2D** or **3D**.



In this example, the effector's joint has been set as a pseudo-root, so it remains pinned to its location. As the second bone in the chain is rotated, the joints after and after it are automatically recalculated.

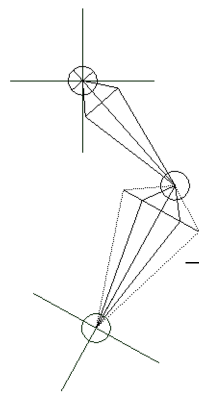
Note that in this situation, if the chain is 2D and the bone is translated out of the resolution plane, the effector will move as the resolution plane is modified.

Displaying Articulated Chains

The Shadow Radius

The shadow radius is a visual reference that simplifies visualizing and selecting bones that are too small or too difficult to see or select. A bone's shadow radius can be resized without affecting the skeleton's structure or the envelope's shape.

For more information on resizing the shadow radius, see the next section.

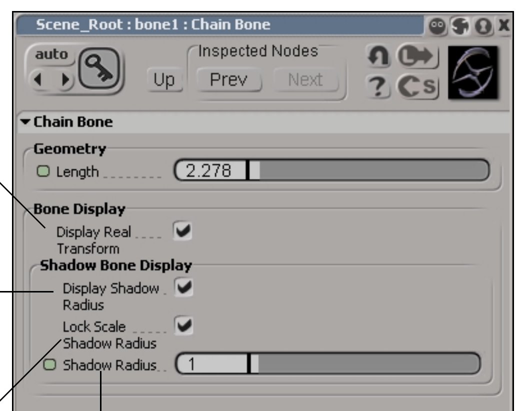


The second bone in this chain has a shadow radius. Clicking on the shadow radius selects the bone.

Modifying a Chain's Appearance

You can control the appearance of a chain on a bone-by-bone basis by using the Chain Bone property editor. To open the Chain Bone property editor, select a bone in the chain and choose the **Property** > **Chain Bone** property from the Selection panel.

Modifying a bone's appearance from the Chain Bone property editor



If Display Real Transform is unchecked, the bone will not be displayed. This parameter is checked by default.

The Display Shadow Radius parameter toggles shadow radius display for the selected bone. This parameter is unchecked by default.

If Lock Scale Shadow Radius is checked, the shadow radius remains at a constant size even if the bone is scaled in the Y or Z axes. This parameter is unchecked by default.

The Shadow Radius value is the scale of the shadow radius in relation to the bone. The default Shadow Radius is 1, the same size as the bone.

Creating a Skeleton

From Hierarchies to Skeletons

Your final objective is not just to build a collection of chains or objects and play with them. Typically, you want to apply an envelope and take advantage of IK to move your character around and have it deform naturally—or unnaturally, depending on your project!

An ordinary collection of chains or objects is considered a skeleton as soon as you assign an envelope to it. An envelope is any model or object that you assign to the skeleton. In most cases, an envelope moves with and is deformed by the skeleton once it is assigned.

It usually simplifies your workflow if the chains and objects that make up your skeleton are organized into a hierarchy before you apply an envelope.

To create the skeleton

1. Draw one or more chains. Optionally, you may get or create any other object you wish to include in the skeleton.

   Normally, a skeleton has multiple chains, in which case you can make the effector of one chain the parent of the chain roots of other chains, and so on.

To assign the envelope

2. Select the object you want to use as an envelope.
3. Choose **Deform** > **Envelope** > **Set Envelope** from the Animate toolbar. The Envelope dialog box appears.
4. Click OK.
5. Pick the first chain or object to include in the skeleton.
   - To pick a single object or one bone from a chain, click on the object or bone.
   - To pick a part of a hierarchy, middle-click on the parent object within the hierarchy.
6. Repeat step 5 for every chain or object to include in the skeleton. To end the envelope assignment, right-click.

For details on assigning and editing envelopes, see *Chapter 9: Envelopes.*

Levels of Hierarchy

Skeleton hierarchies have multiple levels. Each chain root, joint, and effector, as well as any objects used, are all separate entities and “levels” in the skeleton. Thus, every component corresponds to a separate level in the hierarchy.

Animating with Forward Kinematics

When you use FK, you move the skeleton into position, specifying the angle of each joint and then keyframing its rotational position. Each movement needs to be carefully planned to create the resulting animation. For example, to bend a leg, you start from the “top” and move down by rotating the hip, then the thigh, calf, and finally the foot.

When to Use Forward Kinematics

You have complete control over the orientation of each joint with FK, allowing you to create many types of movements that may not be possible to animate with IK alone. This control, however, comes at the price of slow and often tedious work.

The advantages of using FK are:

- You can key the exact orientation of a joint. This prevents any surprises from occurring when 2D chains flatten on their resolution plane.
- Three function curves are generated for each joint orientation, which gives you precise control over your animation.
- You can control certain joints that are difficult to animate, such as shoulders.

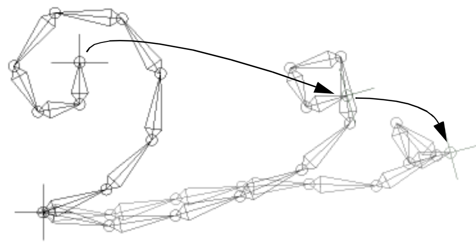
Because the chain’s joints are calculated from the root to the effector, rotating a joint using FK can sometimes “pull” another bone away from a position constraint.

To animate a chain using FK

1. Create a chain by choosing **Create** > **Skeleton** >**Draw 2D Chain** or **Draw 3D Chain** from the Model or Animate toolbars.
2. Select the bone you want to keyframe, and click the Rotate button.
3. Choose **Animation** > **Set Key** from the Animation panel to record a rotation keyframe, or click the keyframe icon.
4. To animate your object, move the timeline pointer and continue to create rotation keys for the joint.

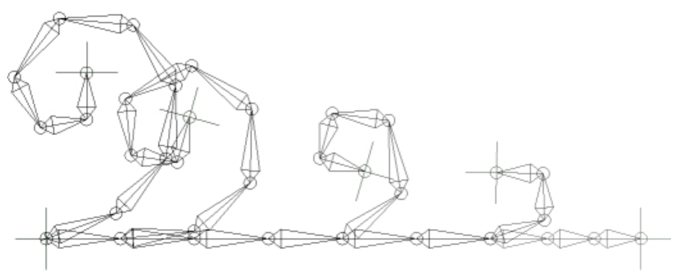
Example

Let’s say you want to animate one of the tentacles on Kraken, the Evil Sea Monster. Currently, the tentacle is coiled, and you want to animate it straightening out. If you were to use IK by keyframing the effector translation, parts of the tentacle would pass through each other.

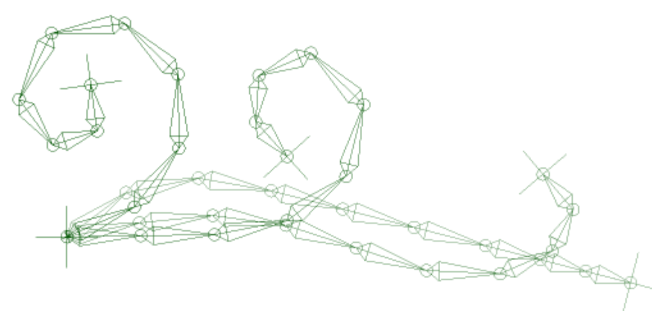


Using IK, the tentacle twists strangely when the effector is keyframed.

Using FK, you would rotate each joint in the chain, starting from the chain root. Although more time-consuming than using IK, the tentacle will not twist into itself. FK also gives you the freedom to choose how you want the tentacle to uncoil—from a lazy extension (Kraken is swimming peacefully) to a whip-like snap (Kraken is attacking a submarine).



Using forward kinematics, the tentacle uncoils without strange twists. In this figure, the joints start rotating at regular intervals, with each joint completely straightening out in a constant number of frames. The result is a slow, lazy uncoiling.



This time, the joints are keyframed with some recoil on the first four joints, creating a whip-like effect. Notice how the second, third, and fourth joints bend backward compared to their original angles. This creates the whip effect, which would be impossible with inverse kinematics.

Animating with Inverse Kinematics

IK provides a quick and easy way to create animation: you define the goal of the action by positioning the effector of a chain, and SOFTIMAGE|XSI calculates how to position the rest of the chain to reach that goal. You save a keyframe for the translation of the effector at different times, and SOFTIMAGE|XSI solves the joint rotations, from the root to the effector, based on the effector's position.

You can keyframe the translation of an effector as you would any other object. When you animate the translation of an effector, three function curves (X, Y, and Z) are generated.

When to Use Inverse Kinematics

IK makes it simple and fast to move characters, but you often need to tweak the movements later to compensate for any unwanted movements. These problems occur because there is often more than one way for the chain to reach the goal set for it by the effector. Luckily, there are many tools to help you fix these problems.

These are the advantages of using IK:

- You can easily try out different movements.
- Animating 2D chains that have a limited range of movement is very fast.
- Simple movements are quick to animate.

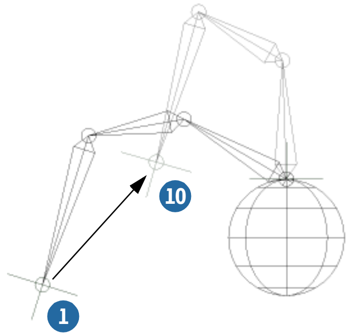
Animating Effector Translation

Animating the translation of the effectors is the most basic way of animating chains with IK.

Transformations on effectors of chains created in SOFTIMAGE|XSI are local to the effector's parent (by default, the root). For global transformations (as in SOFTIMAGE|3D), you must explicitly mark and key the global transform parameters.

To animate the translation

1. Create a chain by choosing **Create** > **Skeleton** > **Draw 2D Chain** or **Draw 3D Chain** from the Model or Animate toolbars.
2. Select the effector and click the Translate button.
3. Choose **Animation** > **Set Key** from the animation toolbar to record a translation keyframe, or click the keyframe icon.
4. To animate your chain, move the timeline pointer and invoke IK by moving the effector to another position. Choose **Animation** > **Set Key** again.
5. To translate the entire chain, select the chain root as a tree and move it to its new position.



In this example, the chain is animated by keyframing the effector's translation at frames 1 and 10. The joint angles for each frame is solved.

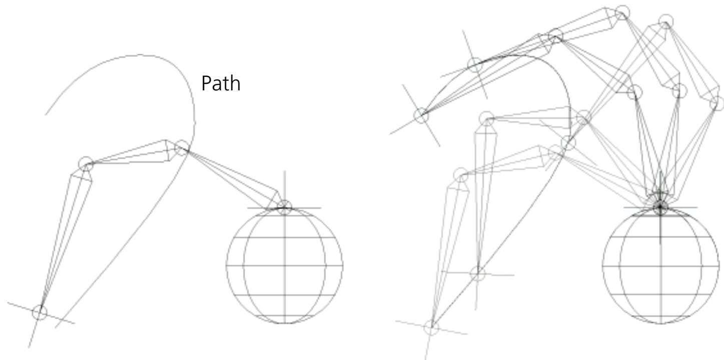
Invoking IK (animating the effector) after invoking FK (keyframing a joint's rotation) or vice versa can result in unpredictable behavior when you play back the animation.

Animating on a Path

You can also translate chains by constraining them to a path or a trajectory following the steps described below:

1. Create a chain by choosing **Create** > **Skeleton** > **Draw 2D Chain** or **Draw 3D Chain** from the Model or Animate toolbars.
2. Create a curve of any type using one of the **Create** > **Curve** commands in the Model or Animate toolbars.
3. Select the bone or effector of the chain to be constrained to the path.
4. Choose **Create** > **Path** > **Set Path** in the Animate toolbar and pick the curve as the path. The Set Path dialog box appears.
5. Enter the frame at which the bone or effector starts moving (Start Frame) and the frame at which the bone or effector reaches the end of the path (End Frame).
6. If you want the bone or effector to move along the path at a constant speed, activate **Linear**.
7. Click OK.
8. Pick the path. SOFTIMAGE|XSI will create keys for each end of the path and interpolate the bone or effector's intermediate positions along the path.
9. The PathCns property editor will appear, allowing you to manually adjust your constraint settings.

The bone or effector joint is constrained to the path. When the sequence is played back, the joint translates along the path just like any other object. Since the joint is part of a chain, however, the principles of IK are still applied to the previous joints in the chain.



Draw path. Animate effector on the path.

For information on path animation, see *Chapter 3: Animating along Paths and Trajectories* on page 91.

Inverse-Kinematics Behavior

When a chain's effector is translated, SOFTIMAGE|XSI uses a *solver*—an algorithm that "solves" how the effector will reach its target by calculating how the joint angles should change. You can use one of two solvers for a chain:

- **SOFTIMAGE|3D Solver** (the default) uses the same algorithm as SOFTIMAGE|3D, but may not be able to solve the effector's translation properly if the chain has both stiffness and rotation limits.
- **SOFTIMAGE|XSI Solver** uses a more efficient algorithm than the SOFTIMAGE|3D solver, providing quicker results when translating the effector.

The SOFTIMAGE|XSI solver is used for compatibility with SOFTIMAGE|3D scenes, but both solvers generally give similar results. If you experience solving problems with a one solver, you can easily switch to another.

If you find that different solvers work best for different situations on the same chain, you can set keys to switch from one solver to another at specific times.

One of three methods for calculating joint rotations when using IK is used:

- **Use Preferred Rotations from Joint Properties** calculates rotation values based on the joints' preferred angles. This means that the joint rotations are always predictable; a given effector position will always result in the same rotation values. This is SOFTIMAGE|XSI's default setting, unless you are using a chain imported from SOFTIMAGE|3D.
- **Use Joint Rotations** (**SOFTIMAGE|3D Behavior**) will change the preferred angles if the joint angles are changed by any means other than IK, which will affect further IK joint calculations. This is the same behavior as in SOFTIMAGE|3D, and chains imported from SOFTIMAGE|3D will automatically assume this behavior.

Because joint rotations can change the preferred angles when this method is used, expressions which modify joint rotations can cause unpredictable behavior.

- **Reuse Joint Rotation at Every Resolution Step** ignores the preferred angles and calculates rotation values based on the joints' current angles. When the chain is extended to its full length, translating the effector toward the root will cause the chain to bend in one direction. As a result, this setting can cause unpredictable behavior.

To change a chain's behavior

1. Select the chain's first bone.
2. Choose **Property** > **Kinematic Chain**. Use the Solver and Solver Angles drop-down lists to change the chain's behavior.

Rotation Limits

When you create a skeleton with many chains, you will find that each chain doesn't always move the way you would like it to. Another problem is that, unlike real limbs, chains don't have that many restrictions in their ability to move. This can cause problems such as a leg bending in an alarmingly unnatural manner.

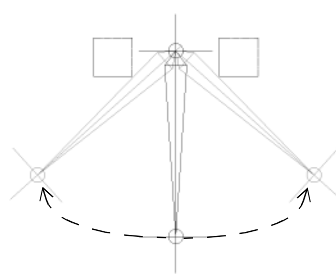
The way to solve these problems is to set "rules" that determine how much each joint should rotate. There are a number of ways you can constrain joints and effectors to limit their movements:

- Limit the movements of joints by using the Rotation Limits and Stiffness properties—see *Setting Rotation Limits* on page 193 and *Setting Stiffness* on page 195.
- Constrain the effector's orientation—see *Changing the Effector's Orientation* on page 196.
- Constrain the chain's orientation using the Up Vector and/or Preferred Axis constraints—see *Controlling the Chain's Orientation* on page 202.

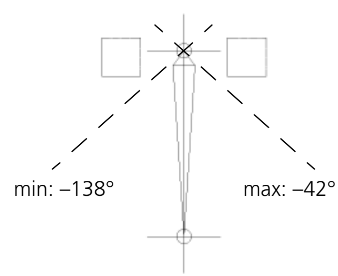
Setting Rotation Limits

The **Rotation Limits** option restricts the range of rotation of one or more chain elements. For example, setting rotation limits could prevent a head from rotating 360 degrees. Local coordinates are always used for the restriction unless the element is not parented, in which case global coordinates are used.

Regardless of the rotation limits settings, IK will continue to prevent joints from bending backward against their preferred angles. However, if both the rotation limits are on the opposite end of the joint's preferred angle, the preferred angle (and the joint) will flip.



Rotation limits
The bone can only rotate about its Z axis and not beyond the surface of the blocks.



Settings
Determine minimum and maximum allowable values for Z-axis rotation. Restrict rotations in X and Y.

To apply rotation limits interactively

1. Create a chain by choosing **Create** > **Skeleton** > **Draw 2D Chain** or **Draw 3D Chain** from the Model or Animate toolbars.
2. Select the joint to be restricted. It's usually best to start with the first joint in the chain. Because each bone is parent to the next bone in the chain, you need to consider all three axes and see how changes affect the subsequent joints.
3. Click the Rotate icon in the Transform panel.

   If you are working with a 2D chain, remember that the axis for a 2D joint is Z. Rotating only in the Z axis keeps the joint properly in the resolution plane.
4. Rotate the joint to its minimum position, which is the farthest clockwise point (assuming the Z axis points toward you) at which you want it to stop.
5. Choose **Create** > **Skeleton** > **Set Minimum Rotation Limit** from the Animate toolbar.
6. Rotate the joint to its maximum position, which is the furthest counterclockwise point at which you want it to stop.
7. Choose **Create** > **Skeleton** > **Set Maximum Rotation Limit** from the Animate toolbar.
8. Repeat steps 2 to 7 for the next joint, if necessary.

To apply rotation limits from the Kinematic Joint property editor

1. Create a chain by choosing **Create** > **Skeleton** > **Draw 2D Chain** or **Draw 3D Chain** from the Model or Animate toolbars.
2. Select the joint to be restricted. It's usually best to start with the first joint in the chain. Because each bone is parent to the next bone in the chain, you need to consider all three axes and see how changes affect the subsequent joints.
3. Choose **Property** > **Kinematic Joint** from the Selection panel to open the Kinematic Joint property editor for the selected bone. Click the **Rotation Limits** tab.
4. Click the Rotate icon in the Transform panel.

   If you are working with a 2D chain, remember that the axis for a 2D joint is Z. Rotating only in the Z axis keeps the joint properly in the resolution plane.
5. Rotate the joint to its minimum position, which is the farthest clockwise point (assuming the Z axis points toward you) at which you want it to stop.
6. In the Rotation Limits property page, set the minimum angles for the appropriate axes.
7. Rotate the joint to its maximum position, which is the furthest counterclockwise point at which you want it to stop.

8. In the Rotation Limits property page, set the maximum angles for the appropriate axes.
9. Check the **Active** checkbox to activate the rotation limits for the joint.
10. Repeat steps 2 to 9 for the next joint, if necessary.

When you add rotation limits to a chain, its joint angles are not recalculated automatically. To update the chain, move the effector.

Setting Stiffness

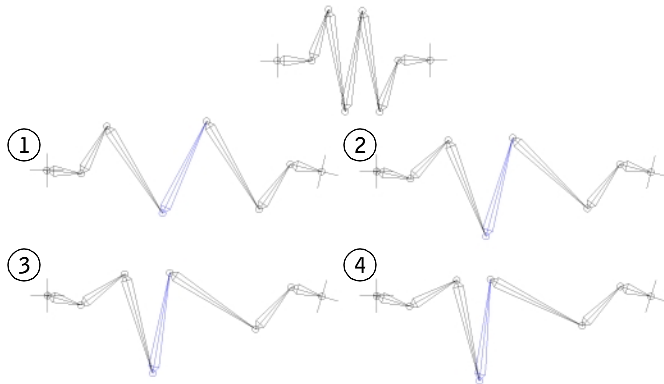
Another way to restrict a joint's range of motion is to set its stiffness. A joint's stiffness is a measure of its resistance to bending during while using IK.

To apply stiffness to a joint

1. Create a chain by choosing **Create** > **Skeleton** > **Draw 2D Chain** or **Draw 3D Chain** from the Model or Animate toolbars.
2. Select the joint to be restricted.
3. Choose **Property** > **Kinematic Joint** from the Selection panel to open the Kinematic Joint property editor for the selected bone.
4. In the Kinematic Joint property editor, set the stiffness. The stiffness value must be between 0 and 1, inclusive.
5. Check the **Use Stiffness** checkbox to activate the stiffness for the joint.

The restricted object never rotates beyond the minimum and maximum values set in the Kinematic Joint property editor.

When you add stiffness to a chain, its joint angles are not recalculated automatically. To update the chain, move the effector.

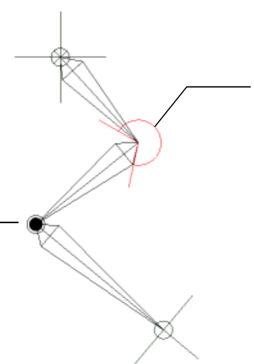


The chain at the top is the original chain. To illustrate how stiffness affects joint behavior, the stiffness of the fourth bone has been set to three different levels, and the effector translated to the same spot. On chain #1 the stiffness is 0, and the joints bend evenly, like an accordion. On chains #2 and #3, the stiffness is 0.25 and 0.5 respectively, and the other joints bend more than the fourth. On chain #4 the stiffness is 1, and the fourth joint is at the same angle as in the original chain.

Displaying Rotation Limits and Stiffness

To display rotation limits and stiffness in a viewport, choose **Show** > **Visibility Options** > **Attributes** from the viewport menu and select **Chain Joint Rotation Limits** from the Selected Objects or Unselected Objects sections.

The *stiffness* is indicated by filling in the joint. The degree to which the joint is filled in represents the degree of stiffness. In this image, the stiffness is at 0.75.



A joint's *rotation limits* are indicated by a circle with two lines. The space opened between the lines indicates the range of motion for the joint.

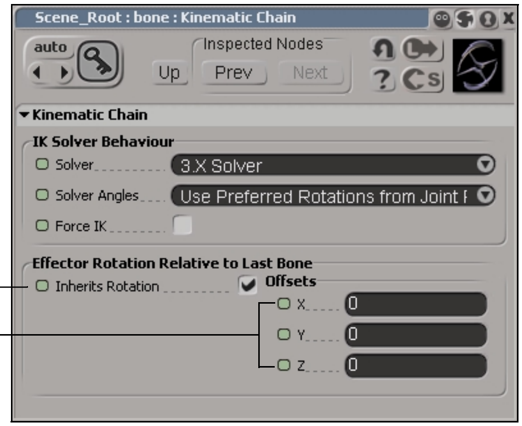
If the joint is 2D, the circle will be red. If the joint is 3D, there will be three circles in red, green, and blue for the X, Y, and Z axes, respectively.

The lines move interactively as you change the rotation limits in the Kinematic Joint property editor.

Changing the Effector's Orientation

The rotation of the effector is important to the workings of an IK chain because the effector can be used for parenting other objects to a chain. By default, the effector's orientation is the same as the last joint's. In some cases, you many not want this orientation to be the same.

The Effector Behavior property in the Kinematic Chain property editor provides two methods for changing the effector's orientation, allowing you to rotate the effector according to global coordinates, or local to the last bone. To display the Kinematic Chain property editor, select the chain's first bone and choose **Property** > **Kinematic Chain** from the Selection panel.



The Inherits Rotation box toggles the effector's dependency on the last bone. If the box is unchecked, you can rotate the effector according to global coordinates.

If the box is checked, you can use the Offsets settings to rotate the effector locally with respect to the last bone.

You can select which of the two approaches to use depending on your needs. For example, if you were using a two-bone chain for a character's leg, the effector would be acting as the foot. If you wanted to keep the foot parallel with the ground during a walk cycle, you would orient the effector globally by unchecking the Inherits Rotation box. If you would rather animate angle of the foot with respect to the bone, you would leave the flex the Inherits Rotation box checked and animate the Offsets settings.

Example: Snowboarder

A good example of constraining an effector's orientation is a snowboarder's feet. No matter which way the snowboard is angled, the snowboarder's feet must be flat against the board.

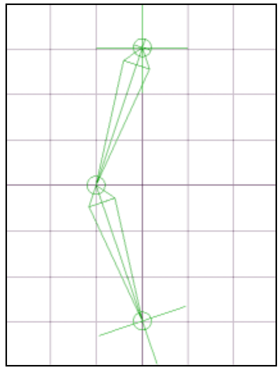
In this example, you'll create a skeleton to represent a snowboarder's legs, with the effectors' orientation determining the orientation of the feet. You will use constraints to keep the effectors attached to the snowboard so that the board's position and rotation drive the leg joints' angles of rotation by IK.

If the effector has the same orientation as the last joint, the snowboarder's feet will always point in the same direction, regardless of the snowboard's position or angle. The solution is to deactivate the Inherits Orientation from Last Bone constraint and constrain the effectors' rotations to the snowboard.

Create the snowboarder's legs and the snowboard

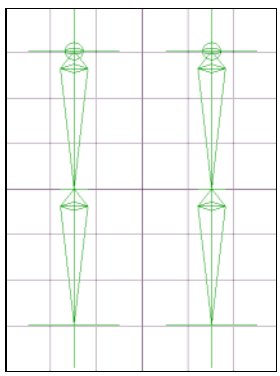
1. Choose **Create** > **Skeleton** > **Draw 2D Chain** from the Model or Animate toolbar.

2. In the Right viewport, draw a 2D chain as in the figure below.



3. In the Front window, translate the chain about 1.5 units to the left.

4. Press Ctrl+d to duplicate the chain. Translate the new chain to the right as in the figure below. Note that the feet would point in the negative direction of the effectors' Y axes.



5. Choose **Get** > **Primitive** > **Polygon Mesh** > **Cube**. Scale the cube to 0.02 in Y and 0.2 in Z.

Constrain the legs to the snowboard

6. Select the left chain's effector. Choose **Constraint** > **Position** from the Constraint panel and pick the snowboard. The effector will jump to the center of the snowboard.

7. In the Position Cns property editor, change the Attach Point - Constraining Object property to adjust the effector's position on the board. Enter a value of –1.5 for X and 5 for Y.

8. Repeat steps 6 and 7 for the right chain. Use a value of 1.5 for X in the Position Cns property editor.

Deactivate the effectors' inheritance constraints

9. Use multiple-select to select the first bone in both chains.
10. Choose **Property** > **Kinematic Chain**. Uncheck the Inherits Orientation from Last Bone check box.

Constrain the effectors' orientation to the snowboard

11. Select the left chain's effector. Choose **Constraint** > **Orientation** from the Constraint panel and pick the snowboard.

Experiment with translating and rotating the snowboard to see how the legs and feet will follow it. Note that if an envelope had already been applied to the skeleton, the feet would be pointing down through the board. You would have to adjust the orientation constraint properties to ensure that the feet stay level with the board. For more information on modifying constraint settings, refer to *Chapter 4: Animating with Constraints*.

Changing the Joint's Preferred Angle

You can change the preferred angle of a joint in a chain, which affects the movement of the chain when you animate it using inverse kinematics.

When you draw a chain, the preferred angles of a joint are the local rotation angles. These angles influence the behavior of the entire chain depending on the selected IK behavior. For more information on IK behavior, see *Inverse-Kinematics Behavior* on page 191.

When using 2D IK chains, only the preferred Z angle of a joint is used (though the preferred X angle has some meaning, as described below). Each joint is constrained to remain on a single plane, and can rotate only around its local Z axis when the effector is moved.

When using 3D IK chains, the preferred X, Y, and Z angles of the joints are used, since 3D joints work like ball joints.

For a 2D chain, the preferred X angle of each joint has a special meaning. The preferred X angle of the first joint of a 2D chain allows you to incline the resolution plane of the entire chain around its root-effector direction. When you change the preferred X angle of the first joint of your chain, the chain is oriented (using the preferred axis or up-vector constraint if you have applied one), then the chain's IK is solved to reach the goal (the translated effector), and finally the resolution plane is inclined based on the preferred X angle you have set. You can also add a rotation offset to the default resolution plane or to the resolution plane specified by the preferred axis constraint or up-vector constraint.

You can rotate a joint in a 2D chain outside of its resolution plane using FK. However, if you invoke IK the chain will snap back to the resolution plane.

Once you have changed the preferred angle of the first joint, the rotations are cumulative when you change the preferred angle of the second and subsequent joints. This allows you to gradually twist an envelope, and you can animate this effect by rotating subsequent joints of a 2D chain.

For example, if you are keyframing the motion of an arm and you want to rotate the arm from the shoulder to the wrist, then you would change the preferred angle of the first joint.

To change the preferred angles

1. Select a chain bone.
2. Choose **Property** > **Kinematic Joint** to open the Kinematic Joint property editor for the selected bone.
3. Set the preferred angles using the **X**, **Y**, and **Z** sliders.

You can animate the preferred angles of a joint by setting keyframes for the Preferred Rotation Angles settings in the Kinematic Joint property editor. When the animation is played back, the keyed values replace the preferred angles of the joint before the chain's IK is resolved. This allows you to change the behavior of a chain as the animation progresses.

Interactively Editing the Preferred Angle

By rotating a joint while the **Kinematic Joint** > **General** property page is open, you can interactively edit the preferred rotation angles of any chain's joint to see how this affects the way the chain reaches its goal. It is also easier to play with the preferred rotation in X of a 2D chain's joints and understand its effect. You can still rotate the joint and see where the effector will be positioned, such as for keyframing the position of the effector.

Preferred angles can only be edited interactively with the rotation tool when the IK is solved as in SOFTIMAGE|3D. For information on modifying IK behavior, see *Inverse-Kinematics Behavior* on page 191.

The preferred rotation angles of the chain's joints are used to determine the preferred pose for the whole chain. However, since IK is used for animation, the final rotations will rarely be the same as the preferred ones, even though the preferred rotation angles greatly influence the final rotations.

If you want to animate a chain using only FK (with no animation on the effector), you usually animate the joints of the chain by rotating them and keyframing those rotations. In this case, you want to animate the chain by interactively rotating the joints of the chain as a real 2D/3D chain with no constraints that would force the effector to stay at a fixed position. This is the default behavior when interactively rotating joints.

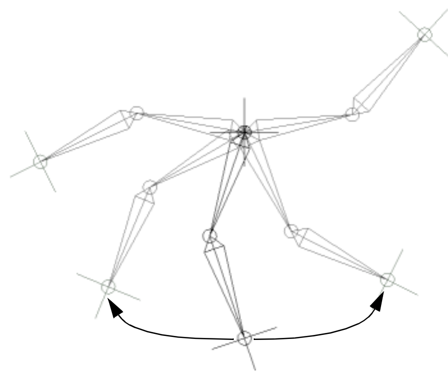
If you intend to animate your chain using IK, it is usually better to interactively edit the preferred rotation angles of the chain and immediately see how this influences the final pose of the chain when the effector reaches its goal. The effector then sticks to its goal and you immediately see the final pose of the chain for those new preferred rotation angles.

Controlling the Chain's Orientation

The Preferred Axis of Rotation

2D and 3D chains have a preferred axis of rotation around which the chain's joints try to rotate when you translate the effector. The default axis of rotation is the Z axis of the chain's root. In a 2D chain, the second and subsequent joints in the chain then rotate like a hinge to change positions in the plane perpendicular to the Z axis, which is the XY plane.

You can predict the behavior of a chain because of this preferred axis of rotation. For example, if you draw a 2D chain in the Front view, when you translate its effector around its root (in the Front view) the chain moves as illustrated below.



If you like, you can change the preferred axis from the default Z to either X or Y. See *Constraining the Chain's Preferred Axis* on page 204 for more information.

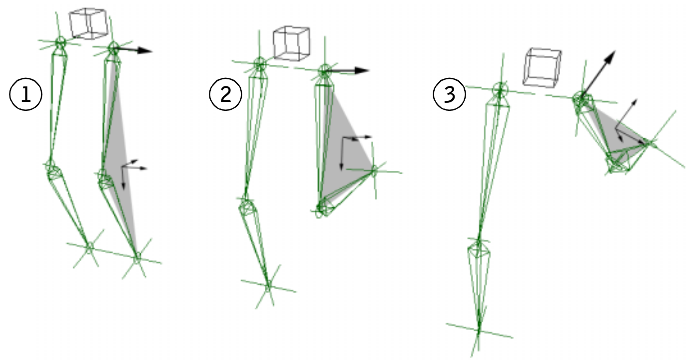
The Chain's Resolution Plane

With 2D chains, the preferred axis of a chain is perpendicular to the plane in which SOFTIMAGE|XSI tries to keep the chain when moving the effector in space. This plane is referred to as the *general orientation* or *resolution plane* of a chain because it is in the space of this plane that the IK system resolves the joints' rotations to get the effector where you want it.

The resolution plane of a 2D chain is the plane in which the entire chain lies. This plane corresponds to the local XY plane of the first joint (and all subsequent joints) of a 2D chain.

The general orientation of a 3D chain is also influenced by its preferred rotation axis. However, the joints of a 3D chain do not necessarily lie on the same plane, as the joints of a 2D chain do.

To get an idea of a resolution plane, extend your arm in front of you and do a biceps curl: as your arm bends, it forms an angle at the elbow. That bend occurs in the plane perpendicular to your body. When you do the biceps curl, that plane (your arm's resolution plane) remains static unless you change it by rotating your shoulder joint.



The resolution plane of this skeleton's leg is illustrated here by a gray triangle, connecting the root, the effector, and the knee. The resolution plane is defined by the first joint's XY plane, and any joint rotations stay aligned with this plane. When the first joint is rotated, the resolution plane rotates accordingly, and all joint rotations remain on the resolution plane.

Preventing Chain Flipping

You can control the orientation of a chain to help you avoid problems with the chain flipping. Chain flipping occurs when the effector of a 2D or 3D chain is moved close to the chain root. When this happens, the general orientation of the chain changes rapidly and dramatically (flips) as the effector crosses the preferred axis.

There are three things you can do to help with this problem:

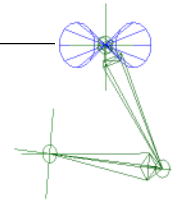
- Show the chain root's critical zones—see *Showing the Root's Critical Zones* on page 203.
- Constrain the chain's preferred axis—see *Constraining the Chain's Preferred Axis* on page 204.
- Constrain the chain's up vector (direction)—see *Constraining the Chain's Up Direction* on page 204.

Showing the Root's Critical Zones

Seeing where chain flipping will occur is the first step to preventing it. The critical zone is an area around the root in which the general orientation of a chain flips if the effector moves into it. The critical zone is represented by a double-cone shaped icon located at the chain's root.

You can use **Show** > **Visibility Options** > **Attributes** from the viewport to see where these zones are to avoid placing the effector in them. The chain will not flip in the plane perpendicular to the preferred axis of rotation.

The critical zone is blue when the default preferred axis of rotation is used or when a preferred axis constraint is applied to the chain. The critical zone is green when an up-vector constraint is applied to the chain.



Constraining the Chain's Preferred Axis

There are some constraints that allow you to control the preferred axis of the chain and its resolution plane.

When you use the preferred axis constraint, you can change the preferred axis of rotation to be X or Y instead of the default Z. When you change the preferred axis of rotation of a chain, SOFTIMAGE|XSI tries to rotate the chain around the new axis when you animate the movement of the effector.

Changing the Chain's Preferred Axis

If you need to change a chain's preferred axis, you can apply the constraint by changing the Preferred Axis property in the Kinematic Joint property editor. For example, if you want to change the preferred axis of a chain from the global Z axis to move around the chain's local Y axis, you would do the following:

1. Draw a 2D chain in the Front view (use **Create** > **Skeleton** > **Draw 2D Chain**).
2. Choose **Get** > **Primitive** > **Null** to create a null and position it in the direction of the local Y axis of the chain's root.

You can quickly create a null by choosing **Create** > **Skeleton** > **Draw 2D Chain** or **Draw 3D Chain**, placing a point, and ending the chain-creation mode by right-clicking.

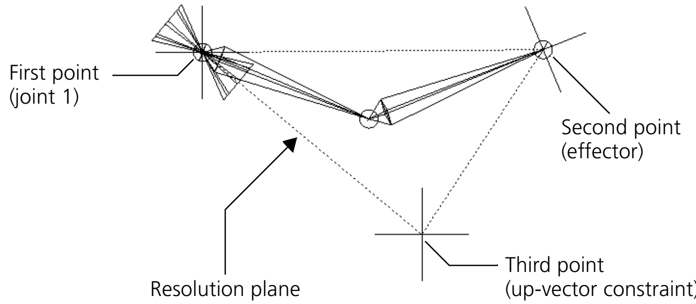
3. Select the first joint of the chain and choose **Create** > **Skeleton** > **Chain Preferred Axis** from the Animate toolbar.
4. Pick the null as the reference to constrain the 2D chain.

Constraining the Chain's Up Direction

You can use **Create** > **Skeleton** > **Chain Up Vector** from the Animate toolbar to constrain the orientation of a chain to prevent flipping. This constraint provides full and precise control of the general orientation of your chain's resolution plane and helps you control where flipping problems will occur (for information on the chain's resolution plane, see *The Chain's Resolution Plane* on page 202).

Using the up vector constraint forces the Y axis of a chain or subchain's first joint to point to a constraining object. Alternatively, you can set an up vector without a constraining object, in which case the up-vector settings define a vector in global space which will constrain the Y axis. The location of the chain's effector determines the X axis of the resolution plane, and the Z axis is perpendicular to the plane defined by the X and Y axes.

The up vector constraint specifies the third point of the resolution plane of the chain, while the first two points are the positions of the first joint and the effector.



The up-vector constraint has priority over the Preferred Axis constraint when both are applied to the first joint in a chain.

To create an up-vector constraint with a constraining object

1. Draw a 2D chain in the Front view (use **Create** > **Skeleton** > **Draw 2D Chain**).
2. Choose **Get** > **Primitive** > **Null** to create a null and position it next to the chain root.
3. Select the first bone of the chain and choose **Create** > **Skeleton** > **Chain Up Vector** from the Animate toolbar.
4. Pick the null as the reference to constrain the 2D chain.

To create an up-vector constraint without a constraining object

1. Draw a 2D chain in the Front view (use **Create** > **Skeleton** > **Draw 2D Chain**).
2. Select the first joint in the chain and press Enter to open the Kinematic Joint property editor. Click the **Resolution Plane** tab.
3. In the Resolution Plane drop box, choose Up Vector. Define the up vector using the Up Vector X, Y, and Z settings.

If you set an up vector for a bone that is not the first bone in the chain or subchain, a subchain will be created starting at that bone's joint.

Using Objects in Skeletons

Using objects in a skeleton framework offers greater control of local deformation. It lets biceps ripple under the skin, or lets a cartoon character's knee "throb" up and down in an exaggerated fashion, independently of the forward motion.

There are many techniques for shape deformation, such as lattices, cluster centers, deforming by surface or curve, and so on (see the *Modeling & Deformations* guide). Several of these techniques offer local control of deformation, but all points are affected equally by the deformation.

By contrast, when you use objects as skeleton elements and assign an envelope, you can take advantage of:

- The way envelope assignment and weighting allows deformation over an area with subtler, diminishing effects.
- The ability to create and combine bounding models on skeleton elements.
- Using expressions to define relationships between parts of skeletons and parts of their envelopes, such as their related transformations and interpolations. For more information on using expressions, see *Chapter 5: Animating with Expressions*.
- The ability to combine them with IK (if skeletons combine objects and chains). You can use expressions to relate the chains' IK with the local deformation of the objects.

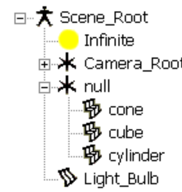
What Kind of Objects?

You can use any kind of object in a skeleton. The geometry of the skeleton object is really not significant because the vertices of the envelope will be assigned to the skeleton object's center. So, whether you use a cube, a sphere, or a null in the skeleton, all transformations of the surrounding envelope are relative in size, orientation, and position to the deforming object's center, not its geometry. The results will look exactly the same.

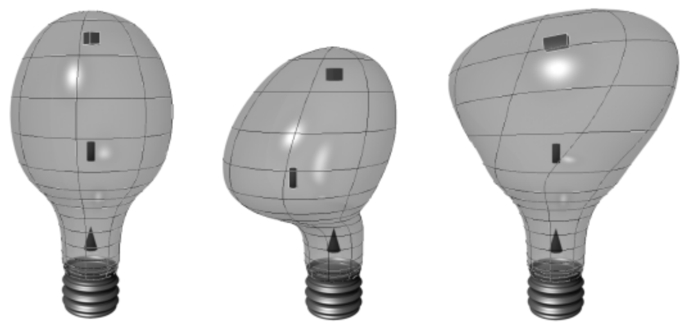
Null objects do not introduce extra geometry into the scene, but objects with geometry are sometimes easier to visualize and interact with. Be aware that, if the skeleton object somehow overlaps the envelope, it shows in the rendering.

Because implicit objects aren't rendered, consider using implicit geometry for skeleton objects.

Light bulb as envelope
The light bulb is an object defined as an envelope to the hierarchy of a cube, a cylinder, and a cone.



Hierarchy of objects as skeleton
The cube, cylinder, and cone were used to illustrate the example. Nulls would produce equivalent results, but geometric objects are easier to visualize.



Deform the light bulb by altering the skeleton
Translate cube (left); translate cylinder (center); scale and rotate cube (right).

A skeleton of an arm could be composed of a chain and a sphere, scaled to the shape of a biceps muscle, positioned, and then constrained to the "forearm" bone of the chain. To animate the deformation, a simple expression could relate the scaling of the sphere (biceps) with the rotation of the forearm joint: as the arm flexes, the sphere is scaled to make a swelling biceps under the skin.

Restrictions

There are some restrictions on these skeleton structures:

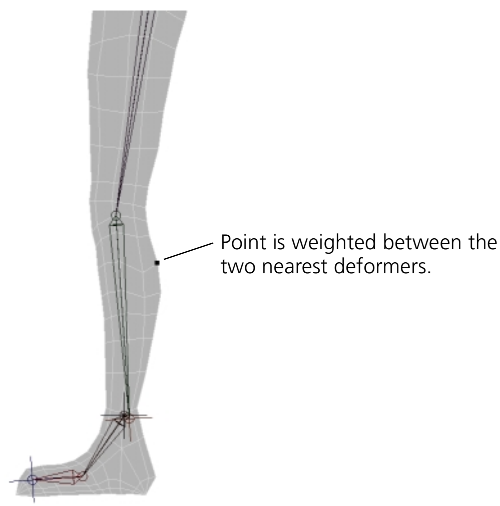
- No vertices can be assigned to the null displayed as the chain root.

Chapter 9 Envelopes

An envelope is an object or hierarchy that is assigned as a “skin” to a set of deformers such as IK chains. Envelopes move and deform in response to the movements of their deformers.

Every point in an envelope is assigned to one or more deformers. For each point, weights control the relative influence of its deformers.

Each point on an envelope has a total weight of 1.0, which is divided between the deformers to which it is assigned. For example, if a point is weighted by 0.75 to deformer A and 0.25 to deformer B, then A pulls on the point three times more strongly than B.



Deformers	Wghts
femur	11.5
shin	88.5

You have several levels of control over how points are assigned and weighted:

- Deformers and weights are initially assigned to points when you apply an envelope deformation. There are several options for controlling how this initial assignment is performed.
- You can use bounding volumes to fine-tune the automatic assignment of deformers.
- You can then select points and reassign them to specific deformers or to no deformers at all.
- You can edit the weighting of individual points manually or by painting.

After you have created an envelope, you can still add and remove deformers. You can also change the reference pose that is used to calculate the initial weight assignment.

When you make changes that cause the initial weight assignment to be recalculated—for example, if you add deformers, reset the reference pose, or add subdivisions to the envelope—all modifications you have made to weights are preserved.

There are several important differences between envelopes in SOFTIMAGE|XSI and SOFTIMAGE|3D:

- In SOFTIMAGE|XSI, envelopes are treated like any other deformation.
- Any object can act as an envelope deformer. There is no need to group objects as a skeleton first.
- There is no longer any distinction between local and global envelopes.
- Rigid envelopes are not supported.
- Because SOFTIMAGE|XSI supports only polygon meshes and NURBS surfaces, envelopes on other types of patches are lost when they are imported and converted. However, you can convert the patches to NURBS and use the **Skin** > **weightCopy** and **weightPaste** commands before importing the converted NURBS envelope.
- Envelopes and deformers do not need to be in a special hierarchy, but you can arrange them in a hierarchy if you wish.

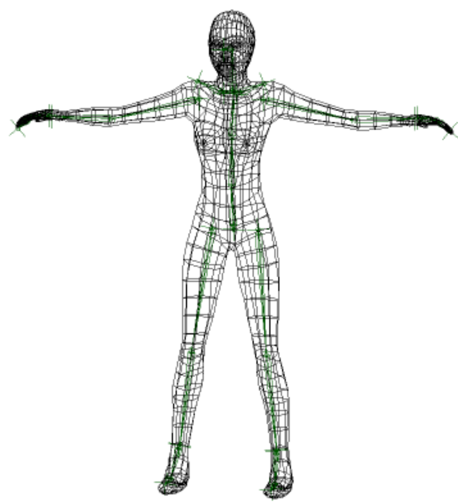
Envelope Assignment

When you define an object as an envelope on a set of deformers, two things happen:

- Each point on the envelope is assigned to one or more deformers. This determines which deformers influence that point.
- Each point's weight is distributed between its deformers. This determines how much each of the deformers influence the point.

Reference Poses

To begin with, your envelope and deformers should be in a *reference pose* (sometimes called a *bind pose*). At the moment you set the envelope, the relative positions of the envelope and deformers determine how points are initially assigned and weighted. It's best to choose a reference pose that makes it easy to see and control how points will be assigned.



In the reference pose, the envelope is not deformed; however as soon as you start to move the deformers, the envelope will begin to stretch and move.

Returning to the Reference Pose

You can return the envelope to its reference pose by muting the envelope deformation:

1. Select the envelope.
2. Click the **Property** button on the Selection panel.
3. Click the Envelope Operator icon to open its property editor.
4. Turn **Mute** on.

Turn Mute off again to restore the envelope deformation.

Freezing

If you have frozen the transformations of an object, you will likely obtain undesirable results when you use it as an envelope. If this is the case, try freezing its geometry with the **Edit** > **Freeze Operator Stack** command first.

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

Setting Envelopes

To assign an envelope to deformers:

1. Select the objects, hierarchies, or clusters to become envelopes.
2. Choose **Deform** > **Envelope** > **Set Envelope** from the Animate toolbar.

   The Envelope dialog box opens. This dialog box is useful if any of the selected objects are already envelopes. If this is the case, see *Adding and Removing Deformers* on page 227. Otherwise, click OK.

3. Pick the objects that will act as deformers. Use the left mouse button to pick individual objects and the middle mouse button to pick branches. If you make a mistake, Ctrl-click to undo the last pick. If you change your mind, press the Esc key to cancel the entire operation.

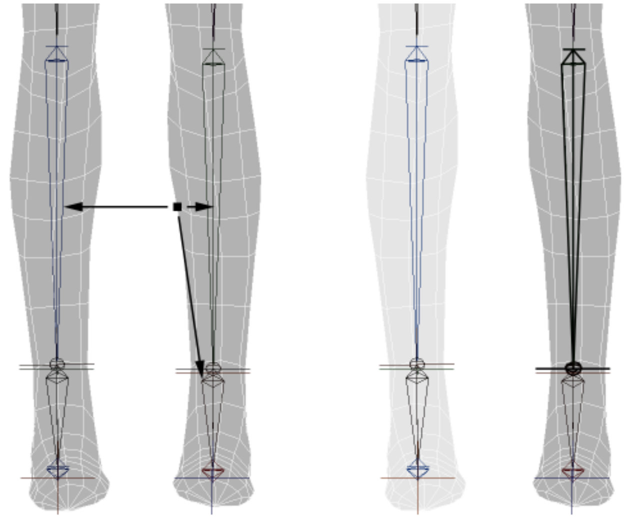
   You can sometimes get better results by picking bones individually, rather than picking entire chains.

4. When you have finished picking deformers, click the right mouse button to terminate the picking session. The Automatic Envelope Assignment property editor opens.

   Each deformer is assigned a color, and points that are weighted 50% or more toward a particular deformer are displayed in the same color. To see the deformer colors on points in a viewport, make sure that **Show** > **Clusters** is on. To change the deformer colors, see *Modifying Display Colors* on page 223.

5. Set the **Number of Skeleton Objects** to which each point on the envelope is assigned initially. For example, if you leave this parameter at the default of 2, each point is controlled by the two nearest deformers and its weight is divided between them.
6. Select the assignment method.

   - **Distance-based** assigns weights based on the distance between points and deformers.
   - **Normal-based** considers the directions of the surface normals in addition to distance. This option is useful, for example, when enveloping things like legs and fingers. This method will never assign points from adjacent appendages to a deformer. The restriction is that the deformers must be entirely inside the envelope.



Distance-based: points are assigned to nearest deformers, which may include the wrong leg

Normal-based: deformers must be inside the envelope

7. In most circumstances, leave the **Force Reassignment** option on. This allows the initial weighting to be recalculated when necessary.

To change these options at a later time, select the envelope then choose **Edit** > **Animation Properties** or **Edit** >**Modeling Properties** and click the Automatic Envelope Assignment tab.

8. Move the deformers to see how the envelope deforms.

If necessary, you can change the deformers to which points are assigned as well as modify the envelope weights using the methods described in the next few sections. If you need to change the reference pose, see *Resetting Reference Poses* on page 226. If you need to add or remove deformers from the entire envelope, see *Adding and Removing Deformers* on page 227.

Transforming Envelopes

Envelopes behave differently from other objects when you transform them:

- If you try to scale, rotate, or translate an envelope, it appears to be unaffected. This is because all of its points are assigned to the stationary deformers. If you turn **Show** > **Centers** on, you will see that the envelope is "really" being transformed.
- If you transform an envelope in Center mode, it appears to react in the opposite way. This is because transforming a center actually applies a compensation to the object's points.

In general, it is not recommended to transform an envelope because you may get strange results if you later reset the reference pose.

Note that, to scale an envelope, you can parent it to the chain root and then scale the chain root in branch mode. Alternatively, you can adjust the bone length and the envelope will follow automatically.

Muting Envelopes

You can mute envelopes in the same way as other deformations. This gives you faster performance because you can pose a skeleton without updating the envelope.

1. Select the envelope.

2. On the Selection panel, choose **Property** and click on the icon of the Envelope Operator to open its property editor. Click the Lock icon to prevent this property editor from being recycled.
3. In the Envelope Operator property editor, turn on **Mute**. This temporarily prevents deformation of the envelope.
4. Pose the deformers.
5. When the skeleton is in the desired position, unmute the envelope.

Using Bounding Volumes

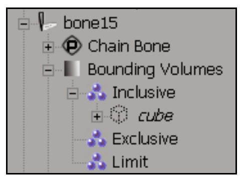
You can use bounding volumes to modify the assignment of envelope weights. Point can be included or excluded from a deformer's influence based on whether they are inside or outside a specified volume. You would typically use non-renderable objects, like an implicit sphere or cube, to define a bounding volume.

This is an optional procedure that provides more control over the assignment of deformers than the default initial assignment. If the initial envelope weights are recalculated, these bounding volumes are preserved.

Bounding volumes are always considered in relation to the reference pose; that is, it doesn't matter whether envelope points are currently inside or outside bounding volumes; what matters is whether they would be inside or outside if the envelope was in its undeformed reference pose.

Bounding Groups

Every deformer has three groups in its Bounding Volumes property. To display these groups, make sure that an explorer view is set to **Show** > **All Nodes**.

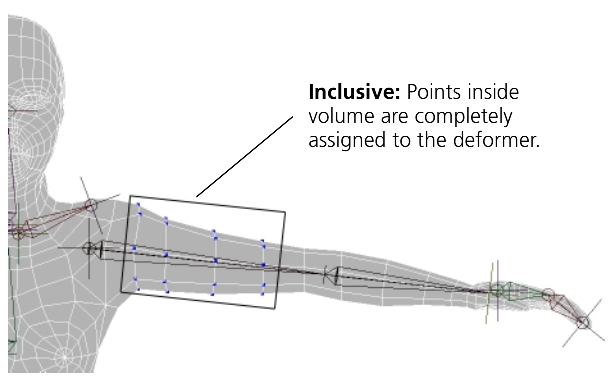


You define bounding volumes for a deformer by adding and removing objects from these groups. These groups are similar to other groups except that they cannot be copied. For more information about groups in general see *Groups and Clusters* in Chapter 5 of the *Fundamentals* guide.

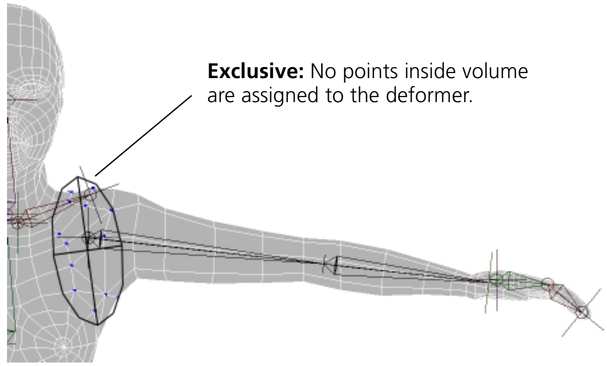
Types of Bounding Volumes

There are three types of bounding volumes:

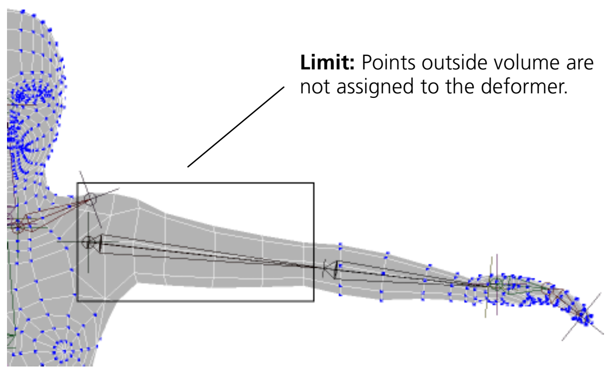
- **Inclusive—**all envelope points within the bounding volume are assigned 100% to the deformer. They are not influenced by any other deformer.



- **Exclusive—**all points within the bounding volume are prevented from being assigned to the deformer.



- **Limit—**points outside the bounding volume cannot be assigned to the deformer; they may be assigned to the deformer as well as to other deformers as applicable.



Defining Bounding Volumes

To define an object as a bounding volume for a deformer:

1. Make sure to start with the envelope in the reference pose.

   To quickly return the envelope to its reference pose, select it then click the **Property** button on the Selection panel, click the Envelope Operator icon, and turn **Mute** on. When you have finished this procedure, turn Mute off.

2. Get an object to serve as a bounding volume; for example, **Get** >**Primitive** >**Implicit** > **Cube** or **Get** >**Primitive** >**Implicit** > **Sphere**.

3. Scale, rotate, and translate the object into the desired position.

4. In an explorer view, make sure that **Show** > **All Nodes** is on and expand the deformer's Bounding Volumes node.

5. Select both the object and the desired bounding group (Inclusive, Exclusive, or Limit) then choose **Edit** > **Add to Group**.

   Alternatively, drag the object onto the bounding group in the explorer.

The bounding objects do not need to be parented to the deformers; however, this can be useful if you think you may later change the reference pose—if they are parented, you can branch select the deformer to move the bounding objects as you move the deformer.

Removing Bounding Volumes

To remove a bounding volume from a deformer, select the object and click the **Ungroup** button on the Edit panel. Alternatively, if you no longer need the object in your scene, you can just delete it.

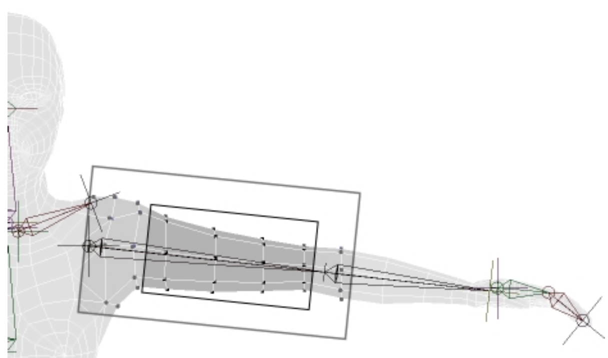
Combining Bounding Volumes

You can combine objects in these bounding groups in various ways. Two examples are given below.

Overlapping Different Bounding Volumes

To achieve a progressively diminishing influence of a deformer, use an inclusive bounding volume around it together with a bigger limit bounding volume.

- Points inside the inclusive volume are assigned 100% to the deformer.
- Points in the spillover area (outside the smaller inclusive volume but within the larger limit volume) are probably assigned to the deformer at something less than 100%, as they are farther away.
- All points outside the limit volume are prevented from being assigned to the deformer at all.

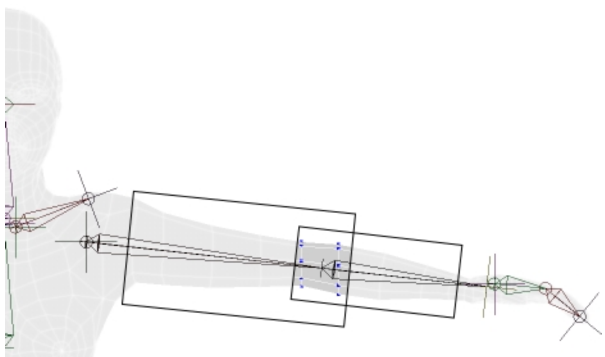


Overlapping Inclusive Bounding Volumes

Sometimes you may want to force some points to be assigned to certain deformers, but not so strongly that they move and deform in an unnatural manner.

To achieve this, create two or more inclusive bounding volumes that overlap near the joint of two bones. The envelope points that fall within the overlapping region of both inclusive bounding volumes behave like this:

- They are assigned only to the specified deformers, regardless of the **Number of Skeleton Objects** value in the Initial Envelope Assignment property.
- They are still weighted according to their relative proximity to the two deformers.
- Envelope points that fall within only one volume are assigned with the usual 100% weighting to the associated deformer.



Changing Deformer Assignments

You can change the deformers that have points assigned to them. For example, you can specify a list of deformers for specific points or completely exclude points from the deformation.

This is another procedure that provides more control over the assignment of deformers than the default initial assignment. Again, if the initial envelope are recalculated, these assignments are preserved.

Specifying Deformers for Selected Points

You can select points on the envelope and assign them to specific deformers:

1. Select points on the envelope.
2. On the Animate toolbar, choose **Deform** > **Envelope** > **Assign Locally**.
3. Pick one or more of the original deformers. Use the left mouse button to pick objects and the middle mouse button to pick branches. If you make a mistake, Ctrl+click to undo the last pick. If you change your mind, press the Esc key to cancel the entire operation.

   You can only pick an object if it is already a deformer for the envelope. To add deformers to an envelope, see *Adding and Removing Deformers* on page 227.

4. When you have finished, right-click to end the picking session.

   The selected points are weighted to the new set of deformers.

Resetting Envelope Weights

You can reset to zero the weights of individual points on an envelope. This effectively excludes these points from the envelope deformation. It also removes any manual modifications to the points' weights that you may have made previously.

1. Select one or more points on the envelope.
2. Choose **Deform** > **Envelope** > **Reset Assignment** on the Animate toolbar. The weight assignment of the selected points is set to zero and they are no longer affected by the deformers.

Reset Assignment does not work with envelopes imported from SOFTIMAGE|3D.

Modifying Envelope Weights

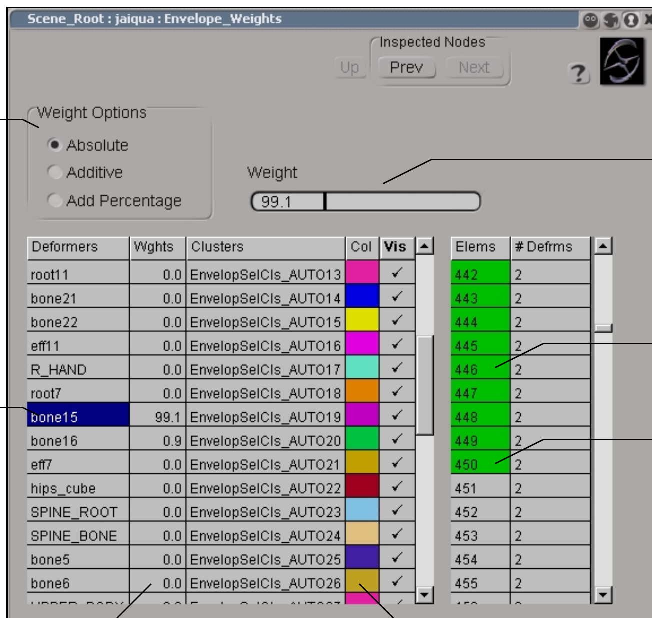
You can modify the weights of points on an envelope either manually or by painting. Both methods use the envelope weight editor.

Each point on an envelope has a total weight of 100%, which is divided between the deformers to which it is assigned. For example, if a point is weighted by 75% to deformer A and 25% to deformer B, then A pulls three times as strongly as B on the point.

Displaying the Envelope Weight Editor

To display the envelope weight editor:

1. Select an envelope.
2. Choose **Deform** > **Envelope** > **Edit Weights** on the Animate toolbar, or press Ctrl+e.



The first time you open the envelope weights editor, you will probably need to resize it to see all the controls. To keep the new size for future sessions, first close the editor and then save your layout. For more information about saving layouts, see *Customizing the Layout* in Chapter 8 of the *Fundamentals* guide.

If you will be working for a while with the envelopes weight editor, it's a good idea to lock it and prevent it being recycled by other property editors.

Selecting Deformers

To select a deformer, click on it in the **Deformers** column. The selected deformer is highlighted in orange in the 3D views. Points that are wholly or partly weighted to the selected deformer are highlighted in green in the **Elems** column.

Selecting Points

To select a point, click on it in the **Elems** column. The selected point is highlighted in white in the 3D views. The **Wghts** column shows how the selected point is weighted to all the deformers.

You can also select multiple points. Click and drag to select a range of points. Ctrl+click to select or deselect multiple points individually.

Showing Selected Points

Because the list of points is often very long, you can restrict it to the points you are working on. In a 3D view, tag the points using the **t** supra key. To show all points again, untag all points.

Modifying Display Colors

You can change the display color for deformers and their point clusters, as well as set the threshold for displaying points in a deformer's color. To display deformer colors on points in 3D views, make sure that **Show** > **Clusters** is on.

Changing Deformer Colors

To change the display color of a deformer and its point cluster in 3D views, click on a color swatch and use the color editor.

Color changes made in the cluster's property editor are not applied.

Changing Color Thresholds

By default, points that are assigned 50% or more to a deformer are displayed in the corresponding color. To change this threshold for a particular deformer:

1. Find the deformer in the **Deformers** column of the envelope weights editor, then find the name of its cluster in the corresponding row of the **Clusters** column.
2. Click the **Clusters** button on the Selection panel, then click the icon of the deformer's cluster. The cluster's property editor opens.
3. On the Envelope Selection Clusters Op page, set the **Weight Threshold** to the desired value.

Setting Weight Options

You can modify weights by painting or by editing them manually, as described in the sections that follow. Either way, the weight options determine how the weights are affected by values you set:

- **Absolute** sets the weight to exactly the value you apply.
- **Additive** adds or subtracts an amount to the current weight.
- **Add Percentage** adds or subtracts a percentage of the current weight.

Editing Weights Manually

To edit envelope weights manually:

1. In the envelope weights editor, select a point in the **Elems** column.

   You can select multiple points in the Elems column, but note that the displayed weight values reflect only the first point selected.

2. Select a deformer in the **Deformers** column.
3. Use the **Weight** slider to adjust how the selected points are weighted to the deformer according to the current **Weight Options**. Alternatively, you can type values directly in the **Wghts** column.

Painting Weights

To paint envelope weights interactively in a viewport:

1. Display the envelope weights as described in the previous section.
2. Press the **w** key to activate the Paint tool.

   The pointer changes to reflect the current brush width. To change the width, click and drag with the middle mouse button. You can also set the radius and other brush properties by choosing **Get** > **Property** > **Paint Properties** or pressing Ctrl+w.

3. Click on the name of a deformer in the envelope weights editor to select it and paint in its color.

   You can toggle the display of weight maps for individual deformers on or off by clicking in the **Vis.** column. To make it easier to see the weights you are painting, turn off the display for the other deformers.

4. In a viewport, click and drag to paint on the envelope. Use the left mouse button to add weight and the right mouse button to remove weight.

   When painting, you can increase performance by reducing the geometry-approximation settings. The Paint tool uses the triangulation of the object to follow its surface. For more information see *Geometric Approximation Parameters* in Chapter 1 of the *Modeling & Deformations* guide.

Freezing Envelope Weight Maps

You can freeze envelope weight maps. This operation collapses the weight map's operator stack, removing the individual paint-stroke operations. It also removes the Automatic Envelope Assignment property, with the result that the initial envelope weights can no longer be recalculated—it's as if the envelope was imported as is.

1. Make sure that the envelope weight map is selected. To select it, first select the envelope then click the **Property** button on the Selection panel, expand the Envelope Operator node, and click on the name of the map *Envelope_Weights*.
2. Choose **Edit** > **Freeze Operator Stack** or click the **Freeze** button on the Edit panel.

Resetting Reference Poses

After an envelope has been assigned, you can change the reference pose of the envelope and automatically reassign envelope weights. Any manual adjustments you made to the deformer assignments and envelope weights are preserved.

You do not need to return to the reference pose to reset it; you can adjust deformers in any pose. When you reset the reference pose, it is updated based on the change in the deformer's position.

1. Select the envelope.
2. On the Selection panel, choose **Property** and click on the icon of the Automatic Envelope Assignment operator to open its property editor.
3. In the Automatic Envelope Assignment property page, make sure that **Force Reassignment** is on. This allows the envelope weights to be recalculated for the new reference pose.

   If your envelope is very heavy, you can temporarily turn this off to speed up interaction while you adjust the poses.

4. On the Selection panel, choose **Property** and click on the icon of the Envelope Operator to open its property editor. Click the Lock icon to prevent this property editor from being recycled.
5. In the Envelope Operator property editor, turn on **Mute**. This temporarily prevents deformation of the envelope.
6. Adjust the positions of the deformers and envelope as desired.
7. When you are satisfied with the new poses, make sure to turn **Force Reassignment** back on in the Automatic Envelope Assignment property page if you turned it off temporarily.
8. Do both or either of the following:
   - To reset the reference pose of a deformer, select it and choose **Deform** > **Envelope** > **Set Reference Poses** from the Animate toolbar. Repeat for other deformers as necessary. Note that you can also branch- or tree-select deformers in hierarchies before choosing this command.

     and/or

   - To reset the reference pose of an envelope, select it and choose the same command.

     The envelope weights are recalculated based on the relative positions of the reference poses of the deformers and envelope.
9. Back in the Envelope Operator property editor, turn off **Mute**. The envelope deformation is reactivated. Move the deformers to see the result of the new assignments and weights.

Adding and Removing Deformers

After you have applied an envelope, you can add and remove deformers. You can also remove the entire envelope.

Adding Deformers

When you add deformers to an envelope, you have the option of recalculating the original initial weights. If the weights are recalculated, any modifications you made to the weights and deformer assignments are preserved.

If you want to recalculate weights, first select the envelope, click the **Properties** button on the Selection panel, click the Initial Envelope Assignment icon, and make sure that **Force Recalculation** is on.

1. Select the envelope.
2. Choose **Deform** > **Envelope** > **Set Envelope** from the Animate toolbar. The Envelope dialog box opens.
3. Do one of the following:
   - If you do not want to recalculate weights, leave the **Automatically Reassign Envelope When Adding Deformers** option off. The new deformers will be added to the list of deformers but no points will be automatically assigned to them.

     or

   - If you do want to recalculate weights, turn this option on. The weight assignments of all the points on the envelope will be recalculated, taking into account any modifications you have made.
4. Click OK to close the Envelope dialog box.
5. Pick the objects that will act as new deformers. Use the left mouse button to pick individual objects, and the middle mouse button to pick branches. If you make a mistake, Ctrl+click to undo the last pick. If you change your mind, press the Esc key to cancel the entire operation.
6. When you have finished picking deformers, click the right mouse button to terminate the picking session.

Removing Deformers

To remove deformers:

1. Select the envelope.
2. Choose **Deform** > **Envelope** > **Remove Deformers** from the Animate toolbar.
3. Pick the deformers to remove. Use the left mouse button to pick individual objects, and the middle mouse button to pick branches. If you make a mistake, Ctrl+click to undo the last pick. If you change your mind, press the Esc key to cancel the entire operation.
4. When you have finished picking deformers, click the right mouse button to terminate the picking session.

Removing Envelopes

To remove an envelope, select it and choose **Deform** > **Envelope** > **Remove Envelope**. This removes the envelope deformation and any modifications you have made to assignments and weights. However, you can undo the envelope removal.

Section III • Animation Mixing

Chapter 10 The Animation Mixer

The animation mixer is a tool that gives you high-level control over animation. It helps you reuse and fine-tune animation you've created with keys, expressions, and constraints. You can also use the animation mixer with actions (stored animations), shapes, and audio.

You work in the mixer by manipulating clips on tracks. Each clip is an instance of a source. On the frames covered by the clip, the items stored in the source are active and play back with the scene. You can have a sequence of clips in time and add transitions. You can also have clips that overlap in time and mix them by modifying their weights.

There are separate mixers for each model, including the scene root. In addition, compound clips each have their own mixer. Every mixer contains its own set of track and clips.

This chapter describes the animation mixer in general, as well as features that are common to all or most types of clip. The chapters that follow describe actions, shapes, and audio in more detail.

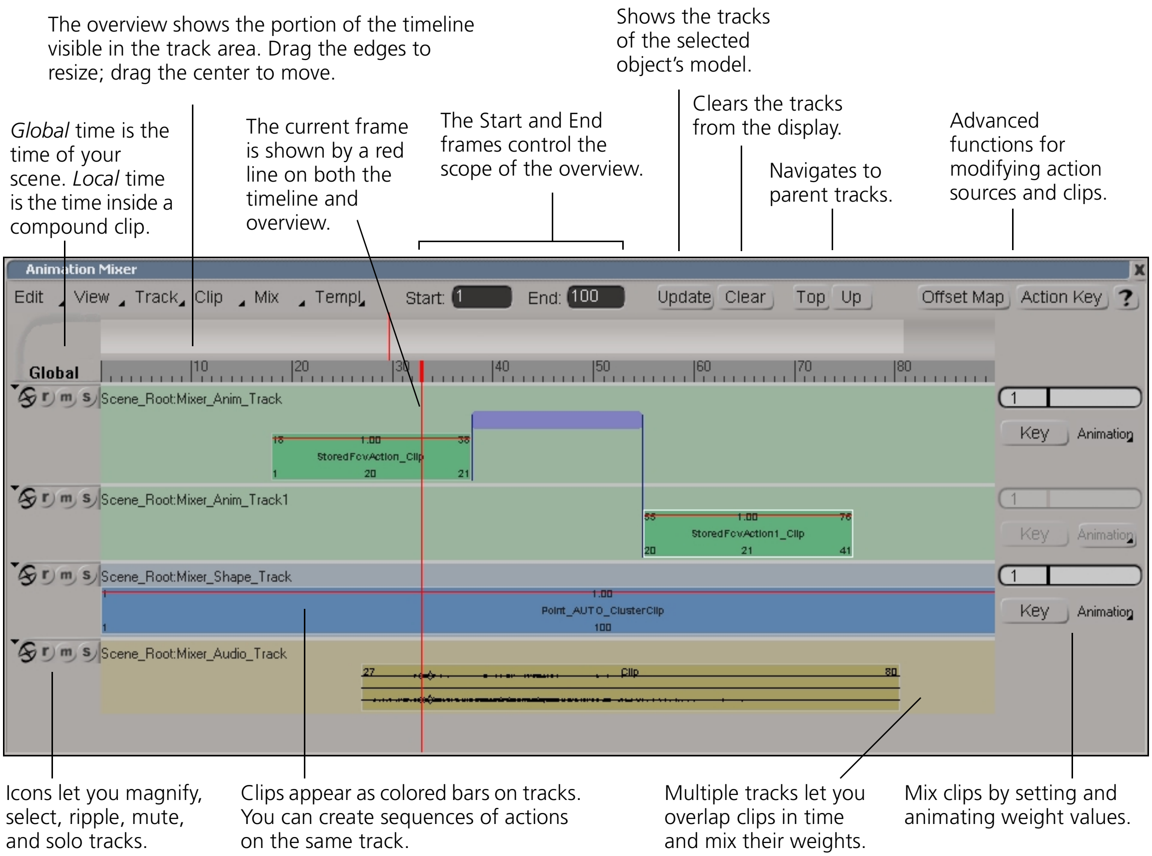
Displaying the Animation Mixer

There are two ways to display the animation mixer:

- To display it docked in a viewport, select it from the viewport's View menu.

  or

- To display it in a floating window, choose **View** > **Views** > **Animation Mixer** on the main-menu bar.



Opening Models in the Animation Mixer

To open a model in the animation mixer, select a model or any of its children and click the **Update** button or choose **View** > **Update from Selected** on the mixer's command bar—this loads the model and displays its tracks.

Alternatively, click the **Top** button in the mixer's command bar to see the mixer containers for all models side by side. To open a container, double-click on it or right-click and choose **Expand**.

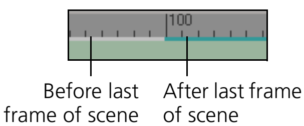
You can navigate down into only one container at a time. However, you can display multiple animation mixers at once in floating windows with different models in each.

Clearing Tracks

You can clear all open models and tracks by choosing **View** > **Clear All** from the mixer's command bar. Alternatively, click **Clear** on the command bar. This removes the tracks from the display—the animation is not deleted.

Viewing in the Mixer

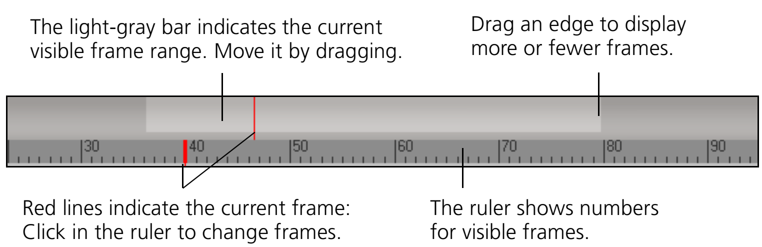
There are many ways to control the view in the track area.



Frame Ranges

Use the **Start** and **End** boxes in the mixer's command bar to set the maximum viewable frame range. These values control the display only—they do not affect the scene's start and end frames. A colored line between tracks indicates frames that are outside the scene's frame range.

You can use the overview area to focus on a specific frame range:



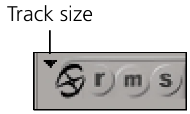
Scrolling Vertically

If there are more tracks than can be displayed at once, you can scroll vertically. Click and drag up or down on an empty area of a track (that is, not on a clip) or on the space below the track control icons to the left of tracks.

Framing

You can frame specific clips in the animation mixer. Select one or more clips, then choose **View** > **Frame Selection** from the mixer's command bar. If you select a track first, this command frames all clips on that track.

To frame all clips on all tracks, choose **View** > **Frame All**.



Track Size

To change the height at which a track is displayed, click the triangle in the track controls to the left of the track and choose a size.

View Options

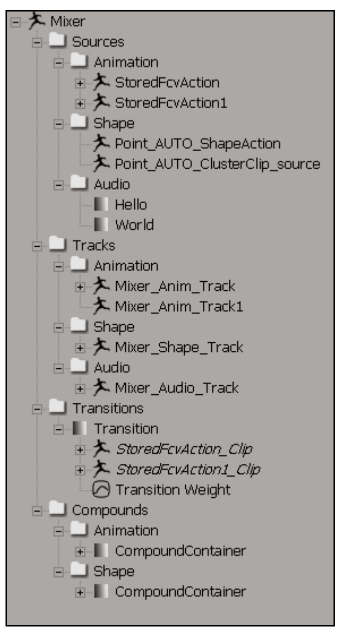
You can toggle the display of many elements on or off.

- To set common view options, use the items on the mixer's **View** menu.
- To set all view options, choose **View** > **Preferences** and use the mixer's Animation Mixer Preferences dialog box.
- To set view options that are specific to tracks or clips, right-click on one then choose **Display Options** and use the Tracks Display Options or Clips Display Options dialog box.

Exploring the Mixer

You can use an explorer view to select and manipulate elements in the animation mixer.

- If the explorer is displayed in a viewport or floating window, make sure that **Show** > **Mixers** is on.
- To display mixer elements in a pop-up explorer, select an object in the corresponding model, then choose **Explore** > **Mixers** from the Selection panel.



Tracks

Tracks are the background on which you add and sequence clips in the animation mixer. There are three types of track:

- **Animation** tracks are light green and can contain action clips. See *Chapter 11: Actions* on page 253.
- **Shape** tracks are light blue and can contain shape clips. See *Chapter 12: Shape Animation* on page 291.
- **Audio** tracks are a light sand color and can contain audio clips. See *Chapter 13: Audio* on page 319.

You can sequence one clip after another on the same track. However, clips cannot overlap on one track; to overlap clips in time, they must be on separate tracks.

Adding Tracks

To add a track, choose the appropriate command from the mixer’s **Track** menu:

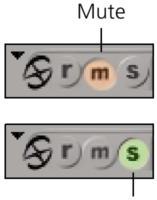
- **Add Animation Track** adds a track that can hold action clips.
- **Add Shape Track** adds a track that can hold shape clips.
- **Add Audio Track** adds a track that can hold audio clips.

Alternatively, right-click on an existing track and choose the appropriate command from the **Add Track** submenu.

Selecting Tracks

To select a track, click the icon to the left of the track. Ctrl+click on the icon to toggle-select multiple tracks. When a track is selected, the line immediately above it is highlighted in white.

Playing Tracks



As you work in the mixer, you can play the tracks in your scene as usual using the buttons in the Playback panel or by scrubbing the timeline. You can also use the Mute and Solo options to concentrate on the tracks you are working on:

- Activate a track’s Mute option to stop it from playing back with the rest of the scene.
- Activate a track’s Solo option to play back only that track. You can solo more than one track so that only the tracks you specify play back.

You can also set these options by right-clicking on a track and choosing **Track Properties** or by selecting the track and choosing **Track Properties** from the Tracks menu.

You can also restrict the playback to the frame range of a marker—see *Playing Markers* on page 252.

You can also mute individual clips. To mute a clip, right-click on it and choose **Action Clip**, then turn **Active** off.

Removing Tracks

Removing a track deletes it together with any clips that it contains. To delete a track, do either of the following:

- Right-click on the track you want to delete and choose **Delete Track**.
- Select the track, then choose **Tracks** > **Remove Track** from the animation mixer's command bar or press the Delete key.

Naming Tracks

To rename a track, you must use the explorer. Right-click on a track and choose **Rename**. For more information about displaying the mixer in the explorer, see *Exploring the Mixer* on page 236.

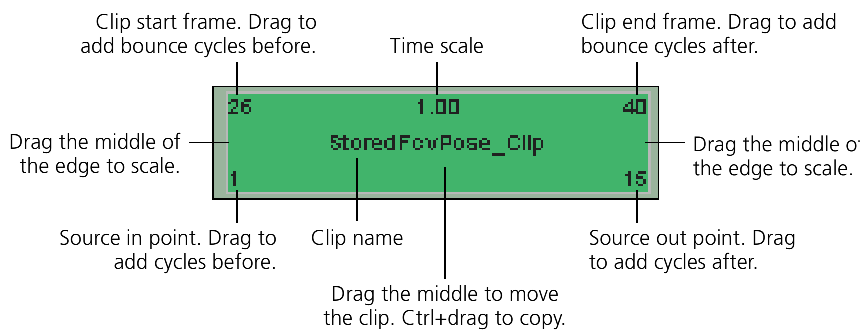
Clips

Clips are instances of sources. They are represented by medium-colored bars on tracks that you can move, scale, trim, and extrapolate. Clips define the range of frames over which the animation items in the source are active and play back.

While sources contain data such as function curves or cluster shapes, clips contain the time control and other properties. You can instantiate multiple clips from the same source and modify the clips independently of each other without affecting the data in the source.

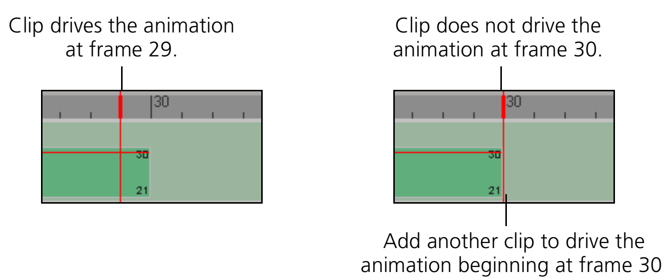
To create clips and their sources, see the appropriate chapter:

- *Chapter 11: Actions* on page 253
- *Chapter 12: Shape Animation* on page 291
- *Chapter 13: Audio* on page 319



Clip Timing

An important thing to remember about clips is that they drive the animation for the frame at which they start but not the frame at which they end:



Viewing Clip Info

You can toggle the information display and other options in any of the following ways:

- Using the **View** menu on the animation mixer command bar.
- Choosing **View** > **Preferences** from the animation mixer command bar.
- Right-clicking on a clip and choosing **Display Options**.

Selecting Clips

To select a clip, click on it. Ctrl-click to toggle-select multiple clips. Selected clips are highlighted with a white border.

Moving Clips

To move a clip, click on it and drag it to a new location. You can drag it somewhere on the same track or to a different track.

Hold the Alt key down while dragging to keep the clip on the same track.

You can also move a clip in its Time Control property editor:

1. Display the Time Control property editor by right-clicking on the clip and choosing **Time Control**.
2. On the **General** property page, set the **Start Offset** to the frame where you want the clip to start.

There is no undo function for dragging an audio clip in the mixer.

Ripple


Ripple

If the ripple option is on, other clips on the target track move automatically when they are “pushed” by the track you are moving. If it is off, you cannot move a clip past the start of the next one or the end of the previous one.

You can set this option with the **r** icon on the track controls at the left of the tracks. You can also set it by right-clicking on a track and choosing **Track Properties** or by selecting the track and choosing **Track Properties** from the Tracks menu.

Ripple is always on when adding clips.

Copying Clips

To copy a clip, hold the Ctrl key while dragging it to a new location. Like moving clips, copying clips is affected by the ripple option.

When you copy a clip, its mix weight is not copied.

The Edit > Copy command and Ctrl+c keyboard shortcut do not work with clips,

You cannot copy audio or compound clips.

Scaling Clips

Scaling a clip makes it longer or shorter, speeding or slowing its animation. You can scale a clip interactively by dragging, or use its Time Control property editor.

The **Scale** value refers to how much time has been scaled. Thus, if time has been scaled by a factor of two, the length of the clip is halved.

You cannot scale audio clips.

Scaling Interactively

To scale a clip, click on the middle of either edge and drag in either direction.

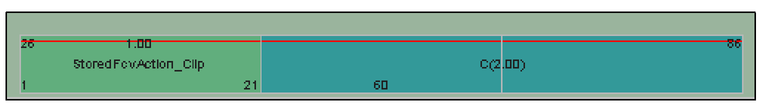
Scaling with the Time Control Property

You can also scale a clip in its Time Control property editor. This lets you enter exact values.

1. Display the Time Control property editor by right-clicking on the clip and choosing **Time Control**.
2. On the **General** property page, set the **Scale** to the desired value.

Extrapolating Clips

Extrapolation determines how a clip affects the frames before and after it. For example, you can repeat the clip (cycle), repeat it backward and forward (bounce), or hold the first or last frame. Extrapolations are displayed as teal-colored bars:



You cannot extrapolate audio clips.

Extrapolating Interactively

To extrapolate a clip, click on any corner of the clip and drag the mouse:

- Drag a top corner to bounce by whole multiples of the clip length.
- Drag a bottom corner to cycle by whole multiples of the clip length.

Extrapolating with the Time Control Property

You can also set the extrapolation of a clip in its Time Control property editor. You can hold the first or last frame as well as cycle and bounce for fractions of the clip length.

1. Display the Time Control property editor by right-clicking on the clip and choosing **Time Control**.
2. Set the desired values on the **Extrapolation** property page:
   - You can set the **Extrapolation Before** and **Extrapolation After** independently.
   - For no extrapolation, set **Type** to **No contribution**.
   - To hold the first or last frame, set **Type** to **Hold** and enter the number of **Frames to Hold**.
   - To cycle the animation, set **Type** to **Cycle** and enter the number of **Cycles**.
   - To bounce the animation, set **Type** to **Bounce** and enter the number of **Bounces**.

Trimming Clips

Trimming (cropping) a clip involves setting its in and out points so that only part of the original source contributes to the clip.

To trim a clip

1. Display the Time Control property editor by right-clicking on the clip and choosing **Time Control**.
2. In the **Source Clipping** box, set **In** to the first frame of the source action you want to use in the clip and **Out** to the last frame.

You cannot trim audio clips.

Renaming Clips

By default, clip names have the form ***sourcename*_Clip**. You can rename clips using the explorer; however, if you later rename the source the clip name will change automatically.

Muting Clips

You can mute a clip so that it does not contribute to the animation on playback but remains in the mixer so that you can reactivate it later. To mute a clip, right-click on it and choose **Action Clip**, then turn **Active** off.

Removing Clips

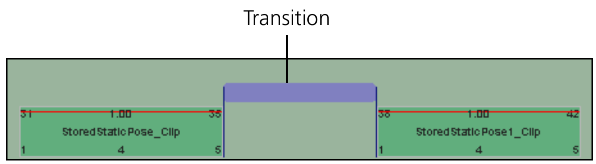
To remove a clip from a track, right-click over the clip and choose **Delete**. This deletes the clip from the track's clip list but does not affect the original animation in the model's source list.

Transitions

Transitions are an interpolation from one clip to the next. They allow the animation to flow smoothly, rather than jerk suddenly at the first frame of a clip. The interpolation is controlled by a transition weight curve: you can add keys and edit the curve to control the timing of the transition.

If the clips do not overlap in time, the interpolation is a weighted blend between the last frame of the first clip to the first frame of the second clip. If the clips do overlap, the interpolation is a weighted blend between the clip values at each frame.

Clips are represented in the mixer by purple bars connecting clips.



You cannot create transitions between audio clips.

If you create a compound clip, transitions are removed.

Adding Transitions

To create a weighted transition curve between clips

1. Choose **Mix** > **Transition Tool** from the mixer's command bar.
2. Pick the first clip.
3. Pick the second clip. The two clips do not need to be on the same track. A transition is created and displayed as a purple bar.

If you cannot see the transition on the track, make sure that **Transition Curves** is on in the **View** menu of the animation mixer.

4. The Transition tool remains active and you can continue to pick clips to create a chain of transitions.
5. When you have finished, click the right mouse button to deactivate the Transition tool.

Adding Transitions Automatically

To create transitions automatically as you add clips, turn on the **Auto Transition** option on the Mix menu.

The **Auto Transition** option is independent of the shape-key modes set on the **Deform** > **Shape** menu of the Animate toolbar.

Modifying Transition Weights

You can modify the profile curve that controls the transition by setting keys or by using the animation editor. Keep in mind that values represent the proportional weight of the second clip:

- A value of 0 represents the last frame of the first clip with no contribution from the second.
- A value of 1 represents the first frame of the second clip with no contribution from the first.
- Values outside of the range [0, 1] are "clamped": values below 0 are treated as 0, and values above 1 are treated as 1.

Keying Transition Weights

To set keys for the transition weight between two clips:

1. Display the transition's property editor by right-clicking on the purple bar that represents the transition, and choosing **Properties**.
2. Use the property editor's animation controls to set keys for **Transition Weight**.

The colors of the animation controls do not properly reflect whether there is a key at the current frame. This is because the keys are set in the local time of the transition rather than the global time of the scene.

Editing the Transition Weight Curve

To edit the transition weight curve directly in the animation editor, right-click on the purple bar that represents the transition and choose **Animation Editor**. Keep in mind that time is relative—the transition appears to be 30 frames long, but this time is stretched or squashed to the duration of the transition.

Removing Transitions

To remove a transition, right-click on the purple bar that represents the transition and choose **Delete**. Alternatively, you can use the Clear Transition tool:

1. Choose **Mix** > **Clear Transition Tool** from the animation mixer's View menu.
2. Pick the clips between which you want to remove the transitions. The transitions are removed.
3. The Clear Transition tool remains active and you can continue to pick clips.
4. When you have finished, click the right mouse button to deactivate the Clear Transition tool.

Mixing Clips

When two or more clips overlap in time and drive the same elements, you can mix them by setting weights that control their relative influence. The higher the mix weight, the more strongly a clip contributes to the combined animation. You can also set the mixer properties to control how weights are combined in each model and compound clip.

Mix weights can be set outside the range [0,1]. By default, mix weights are normalized so that, for example, if Clip A has a weight of 1.0 and Clip B has a weight of 1.5 at a given frame, then A's influence is 40% (1.0 ÷ (1.0 + 1.5)) and B's influence is 60% (1.0 ÷ (1.0 + 1.5)). You turn off normalization in the mixer's property editor, in which case it is possible to "overshoot" the data in the clip by setting weights higher than 1.0.

Setting and Keying Mix Weights

When setting the weights of clips in the animation mixer, make sure that **View** > **Weight Mixer Panel** is on. A Weight Mixer Panel is displayed at the right of each action and shape track.

For each track, the Weight Mixer Panel affects the clip at the current frame. If the frame slider is not on a clip, the mixer for the corresponding track is unavailable.

- Use the slider to set the clip's weight. To set values higher than 1, enter them in the number box.
- Click **Key** to set a key for the clip's weight at the current frame.
- Use the **Animation** menu to access other options. For example, you can open the animation editor, copy and paste animation, set an expression on the mix weight, and so on.

When setting clip weights, note the following:

- If a clip's weight is 0.0, it does not contribute to the animation even if there is no other clip driving the elements at that frame.
- In the animation editor graph, the horizontal axis refers to each clip's local time rather than the scene's global time. If clips are of different lengths or have been offset, scaled, or trimmed, the frame numbers in local time and global time are different.
- If a clip's weight is not driven by a function curve (for example, if it is controlled by an expression or link), no weight curve is displayed on the clip when **View** > **Weight Curves** is on.

Setting Weights in a Clip's Property Editor

You can also set weights in a clip's property editor:

1. Right-click on a clip and choose **Action Clip**. The Instanced Action property editor opens.
2. Set the **Weight** parameter.

The colors of the animation controls in its property editor do not properly reflect whether there is a key at the current frame. This is because the keys are set in the local time of the clip rather than the global time of the scene.

Setting Mixer Properties

The mixer properties control how the weights of clips are combined within the mixer or a compound clip. To set the mixer properties:

1. Right-click anywhere in the track area of the animation mixer and choose **Mixer Properties**, or choose it from the Mix menu.
2. Set the various options:

- **Active** controls whether the entire compound clip contributes to the animation during playback.
- **Normalize** controls how the mix is calculated:
  - When on, mixes are a weighted averages. The results are mixes that fall in-between the values of the separate clips.
  - When off, mixes are additive. The values of the separate clips are added on top of each other.
- **Quaternion Mixing** controls whether rotation values are mixed according to their quaternion values (on) or their Euler values (off). Quaternions usually result in smoother rotations. When mixing three or more rotations, Euler angles are always used.

If shape clips are in a compound clip, you can open the mixer properties as described above. However, if the shape clips are not in a compound, you must open their mixer properties using the explorer: make sure that **Show** > **Mixers** is on then expand the model's Mixer\Compounds\Shape\CompoundContainer node and click the Mixer icon.

Timewarps

Timewarps change the relationship between the local time of the clip and the time of its parent (either a compound action or the entire scene). You can make a clip speed up, slow down, and reverse itself in a non-linear way.

Creating Timewarps

To apply a timewarp to a clip:

1. Right-click on a clip and choose **Time Control**. The clip's Time Control property editor opens.
2. On the **Warp** property page, turn on **Do Warp**.
3. Edit the function curve to modulate the timewarp. The graph maps the clip's local time along the X axis to the parent's time along the Y axis. The graph is a miniature version of the animation editor—while editing the curve, you can use any of the keyboard commands defined for the animation editor. You can also right-click in the graph to display the animation editor's menus. For details about using the animation editor, see *Editing Function-Curve Animation Using the Animation Editor* on page 62.

   Remember that you need to select the curve before you can edit it.

If **View** > **Time Warps** on the mixer's command bar is active, the active timewarps are displayed as profile curves in the clips.

Removing Timewarps

To remove a clip's timewarp, open its Time Control property editor and turn **Do Warp** off. The mapping from the clip's local time to its parent's time becomes linear again. However, the warp profile curve is not affected and you can turn **Do Warp** back on to reactivate it without re-editing the curve.

Compound Clips

You can combine multiple clips into a single compound clip and then move, scale, extrapolate, and warp the entire compound at once. You can also create transitions between compound clips. Certain procedures automatically create compound clips; for example, when you save shape keys, a separate compound is created for each cluster.

In the animation mixer, compound clips appear darker than ordinary clips.

- Each compound can contain only one type of clip: either actions or shapes. You cannot add audio clips to a compound.
- You cannot create compound inside other compound clips.
- When you create a compound clip, transitions are removed.
- You cannot copy compound clips.
- You cannot save presets for compound clips.

Creating Compound Clips

1. Select multiple clips using Ctrl-click.
2. Choose **Clip** > **Create Compound Action**.

   A compound clip is created on a new track, and the selected clips are removed from their old tracks.

Navigating Compound Clips

Use these procedures to navigate through compound clips in the mixer.

To open a compound clip

Do any one of the following:

- Double-click on a compound clip.

  or

- Right-click on a compound clip and choose **Expand**.

  or

- Select a compound clip and choose **Clip** > **Open Compound Action**.

To collapse a compound clip and return to its parent

Do any one of the following:

- Click the Back icon on the mixer command bar.

  or

- Double-click on an empty area of a track.

  or

- Choose **Clip** > **Collapse Compound Action**.

Local versus Global Time

When working with compound clips, it is important to remember the difference between global and local time. The time is indicated in the upper-left corner of the mixer.

- **Global** time corresponds to the time line of the scene. Thus, frame 1 in global time is the first frame of the scene.

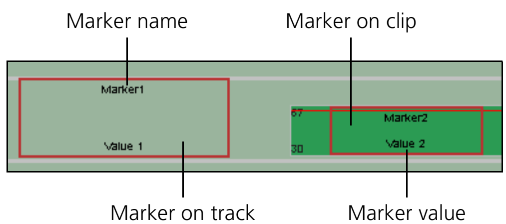
- **Local** time is the time within the clip. Frame 1 in local time is the first frame of the clip, but it is not the first frame of the scene if the clip starts at a different frame in global time. In addition, clips may be scaled, which further complicates the relationship between local and global time.

For example, suppose you add a 20-frame clip to a track and drag it to start at frame 5. The first frame of the clip (frame 1 in local time) corresponds to frame 5 in global time, and the last frame of the clip (frame 20 in local time) corresponds to frame 25 in global time.

Suppose you now scale the clip's time by two so that it is half as long. The clip is still 20 frames long in local time but only 10 frames long in global time. It still begins at frame 1 in local time and frame 5 in global time but it now ends at frame 20 in local time and frame 15 in global time.

Markers

Markers let you highlight and annotate frames on tracks and in clips; for example, to identify phonemes when doing lip synch. You can also restrict playback to the frame range of a marker. Markers are displayed as rust-colored outlines in the animation mixer.



Adding Markers

You can add a marker on a single frame or on a range of frames. Depending on where you click, you can add a marker to a track or clip. You can also add a marker across all tracks in a model or compound clip.

Adding Markers on Single Frames

To add a marker on a single frame, right-click at the desired frame and choose **Add Marker**. Make sure you right-click on a clip (medium-colored bar) to add a marker to a clip, and on a track (light-colored bar) to add a marker on a track.

You can choose to display the marker's name when you hold the mouse pointer over it for a brief period. Choose **View** > **Preferences** from the animation mixer's command bar and make sure that **Tooltips** is active on the Markers page.

If you don't see the correct results when undoing or redoing marker creation, click **Update** in the mixer command bar.

Adding Markers on Ranges of Frames

To add a marker on a range of frames:

1. Hold the Ctrl key down while dragging the mouse over a track. The selected frames are highlighted in gray.

2. Right-click and choose **Add Marker**. Make sure you right-click on a clip (medium-colored bar) to add a marker to a clip, and on a track (light-colored bar) to add a marker on a track.

To clear the highlighted frames without creating a marker, choose **View** > **Clear Region** from the mixer's command bar.

Adding Markers Across All Tracks

You can add a marker across all tracks in a model or compound clip:

1. Click the **Up** button on the mixer's command bar.
2. Ctrl+drag to highlight a frame range on the model or clip's track.
3. Right-click over the model or clip and choose **Add Marker**.

You cannot interactively move or resize a marker if it spans multiple tracks. However, you can set its start frame and duration using scripts.

Selecting Markers

To select a markers on a track, click on it. Ctrl+click to toggle-select markers on tracks; selected markers are highlighted in white.

To select a single marker on a clip, use the explorer. Markers on clips are stored in the Markers folder of the clip, which is in turn inside the Clip List folder of the track.

To select all markers on a clip, right-click on the clip and choose Select All Markers. Note that selected markers on a clip are not highlighted.

Renaming Markers

To rename markers, use the explorer view. Right-click over a marker and choose **Rename**.

Setting Markers' Value

Every marker must have a unique name. However, any number of markers can share the same value. This is useful when annotating tracks because, for instance, it means that you can have several markers with a value of "Jump."

To set a marker's value

1. Display the marker's property editor by right-clicking on the marker and choosing **Properties**.
2. Enter any annotation in the value box; for example: "fluffy's left foot down."

Moving Markers

To move a track marker, click and drag it along the track. You cannot interactively move clip markers—they move with the clip. In addition, you cannot interactively move markers that span multiple tracks.

Changing the Duration of Markers

You can change the duration of a marker on a track by dragging either of its edges:

1. Move the mouse pointer over one of the edges of a marker.
2. Once the edge turns orange, click and drag in either direction.

You cannot interactively change the duration of markers on clips or markers that span multiple tracks.

Playing Markers

To repeatedly play back the frame range of a marker, right-click on the marker and choose **Set In/Out Loop**. Use the Play button on the Playback panel to stop and start the animation, and the Loop button to control looping.

Removing Markers

To remove a marker, right-click on it and choose **Delete**.

Chapter 11 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 jumps, and copy them from one model to another. You can apply them one after the other in any sequence so 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.

Actions also provide a method for combining animation from different sources: function curves, expressions, constraints, and linked parameters. You can also combine inverse and forward kinematics on the same chain by putting them in separate actions and combining them.

Actions in SOFTIMAGE|XSI are not the same as actions in SOFTIMAGE|3D, where they are elements within **.ani** files. In SOFTIMAGE|XSI, actions are separate elements.

Actions and Models

Actions are associated with the models in your scene. Models provide a way to organize the objects in a scene. Each model has its own animation mixer and maintains its own namespace. You can have two or more models in a scene, each with their own objects called left_arm, right_arm, and so on, and reuse actions created on one model to animate the other models.

The root of a model is in its parent model's namespace. If you store actions for the root of a model, you must work in the parent model's mixer.

For more information about models in general, see *Models* in Chapter 5 of the *Fundamentals* guide.

If you split a model by creating submodels after you have created action sources, the existing actions are not updated.

If you duplicate a model in branch mode, or if you export and import a model, all clips it contains are connected as appropriate. However, if you duplicate something that is driven by an action in the mixer without duplicating the entire model, the duplicated object is not driven by the mixer.

Action Sources and Clips

Actions have two parts: sources and clips. First you animate a model and store the animation as an action source, then you instantiate that source as clips on tracks in the animation mixer. Each source can be instantiated by multiple clips. Each clip is an instance of a source and has its own time control and other local properties. For more information about clips, tracks, and the animation mixer in general, see *Chapter 10: The Animation Mixer* on page 231.

Action Sources

Action sources contain one or more animation items. Each animation item contains a parameter name and its animation data (function curve, expression, constraint, and so on).

The parameter name is used as a hint to connect the data when you instantiate a clip. You can remap the parameter names in an action source using a connection-mapping template as described in *Connection-Mapping Templates* on page 268. If a parameter name cannot be resolved when a clip is instantiated, you can specify a parameter to which to connect the data as described in *Resolving Unconnectable Parameters* on page 265.

When you store an action source that contains an expression, the model name of the target parameter is not stored. The target must be an item within the model. This lets you transport an action with an expression between models.

You can modify the animation data stored in a source as described in *Modifying Action Sources* on page 283. You can also specify offsets to add to the animation data when a clip is instantiated as described in *Value-Mapping Templates* on page 280.

Action Clips

Action clips also contain one or more animation items. For clips, each animation item contains the name of the parameter that is driven and its data.

For function curves, the data is a reference to the source—all clips share the function curves of their source. For expressions and constraints, each clip maintains its own local data—you can edit this data independently for each clip as described in *Editing Clip Data* on page 276. You can also adjust the offset of the source animation data as described in *Modifying Action Values* on page 275.

Once a clip has been instantiated, you cannot reconnect the animation to a different parameter. Instead, you must delete the clip and instantiate a new one using a different connection-mapping template.

Actions and Animation

Actions let you combine animation from different sources. A parameter can still be driven by another animation source, such as a function curve or expression, in addition to one or more action clips. When a parameter is driven by a clip at a given frame, the animation values determined by the mixer take precedence over the others. Between clips, the parameter is driven by the “background” animation. Even on frames where a parameter is driven by an action, you can still adjust values and set keys to define the background animation. You can also use a fill action to set the background animation; see *Fill Actions* on page 286.

Storing Action Sources

Storing an action for a model adds the action to the model's list of animation sources. Later, you can create one or more clips for the animation—each clip is an instance of the action.

You can create and store an action from a model's existing animation or current values, or you can store an action that has been saved to disk as a SOFTIMAGE|3D **.ani** file. In both cases, the action appears in the model's Mixer/Sources/Animation branch in the explorer—to see it make sure that **Show** > **Mixers** is on.

If you create action sources with different frame rates, there may be rounding errors when you instantiate clips based on the sources.

Choosing What to Store

When creating action sources, you can store either the transformations (scaling, rotation, translation) or marked parameters.

Although you can create an action based on an object's transformations without explicitly marking them, it might still be a good idea to mark them if you want to keep your data light; for example, you can mark just the translation and rotation to leave scaling values out of the action.

If you are working with inverse kinematics (IK), select the effectors and the chain root, and mark the local translation values (kine.local.pos). If you are working with forward kinematics (FK), select the bones and mark the local rotation values (kine.local.rot).

Remember that constraints act in global space, not local space. If you are using constraints and marking parameters, you must mark the global transformations (kine.global.*). If you are storing transformations instead of marked parameters, the constraints in global space take precedence over other animation in local space.

If there is an up-vector operator on a constraint, the up-vector operator will not be included in with the constraint in the action. Function curves are the only animation sources driving constraint parameters that can be stored in actions.

You can store only parameters that are below a model or the Scene_Root node in the scene hierarchy. For example, you cannot store partition or image clip properties because these parameters are under the project and not part of a model.

Storing New Actions

To create a new action and store it in a model's list of animation sources:

1. If desired, animate the model. You can use any combination of function curves and expressions for animating the transformations and other parameters.

   If the model is not animated, you can still store a "static action" that contains a snapshot of the current parameter values but no animation.

2. Select the objects. If you are not storing transformations, you should also mark the parameters whose animation or values are to be included in the action. See the previous section, *Choosing What to Store*.

   If an object is selected in branch mode, animation will be stored for its children as well. This is unlike saving keys and plotting actions, which work only on the root of the selection.

3. From the **Actions** > **Store** menu of the Animate toolbar, choose the appropriate command:

   - **Transformations - Current** creates an action that is a snapshot of the local transformations at the current frame. The action does not include any animation.

   If you are familiar with SOFTIMAGE|3D, selecting an object in branch mode and choosing **Transformations - Current** is the equivalent of saving a structure key.

   - **Marked Parameters - Current** creates an action that is a snapshot of the values of the marked parameters at the current frame. Again, the action does not include any animation.
   - **Transformations - Fcurves** creates an action for the local and global transformation parameters that are animated with keys (function curves).
   - **Marked Parameters - Fcurves** creates an action for all marked parameters that are animated with keys.
   - **Transformations - All Sources** creates an action for all animated local and global transform parameters, no matter how they are animated: keys, expressions, linked parameters, and so on.
   - **Marked Parameters - All Sources** creates an action for all marked parameters, no matter how they are animated.

   The Store Action dialog box opens.

4. Set the options as desired:

   - **Action Name** lets you apply a name to the action, such as "Jump." You can also rename the action source later in the explorer.
   - **Number of Items** refers to the number of parameters to be included in the action. This value is provided for information only.
   - **Default In** and **Default Out** specify the frame range of the animation to store.

- **Remove Animation** determines whether or not to remove the animation from the corresponding parameters. You can quickly build up a library of actions; for example, by creating a walk cycle, storing it and removing the animation, then creating a jump, and so on. You can later reapply the animation as described in *Applying Actions* on page 260.

When **Remove Animation** is on, the entire animation is removed from the parameters, not just the keys between **Default In** and **Default Out**.

When you click OK, a mixer branch is created under the model in the explorer if there isn't one already, and the action is added to the Mixer/Sources/Animation branch.

Plotting Animation to Actions

When you plot an action, the animation is evaluated frame by frame and function curves are created. These function curves are stored in an action source. This lets you "bake" animation from a combination of sources (including action clips) into a single action. Plotting does not remove the existing animation.

1. Select the objects to plot. If you are not storing transformations, you should also mark the parameters whose animation or values are to be plotted. See *Choosing What to Store* on page 257.

   If an object is selected in branch mode, animation will not be plotted for its children. This is unlike storing actions, which automatically expands the branch selection; rather, it is similar to saving keys work only on the root of the selection.

   To plot the animation of an objects children as well, branch-select then object and then choose **Selection** > **Select Child Nodes**.

2. Choose one of the following commands from the **Tools** > **Plot** menu of the Animate toolbar:

   - **All Transformations** plots the local and global transformation parameters. Only parameters with animation are plotted.

   or

   - **Marked Parameters** plots the marked parameters.

3. Set the options:

   - **Start Frame** and **End Frame** specify the range of frames to plot.
   - **Step Value** specifies the frame step increment. If this value is 1, a key is saved at every frame. If this value is 2, a key is saved at every other frame, and so on.
   - **Action Name** specifies the name for the new action source.

4. Click OK. A new action source is created in the model's Mixer\Sources\Animation folder and is available for instancing clips.

Applying Actions

Applying an action restores the animation contained in an action back to the original objects. It does not create a clip, and it does not use any connection- or value-mapping templates. It also uses the timing of the original animation, so it does not matter what the current frame is when you apply an action. This is useful for restoring animation if you accidentally removed it when creating an action.

1. Select the action to apply. Actions are stored in the Mixer/Sources/Animation path of the parent model. Make sure that the explorer is set to show all nodes.

2. Choose **Actions** > **Apply Action**. The animation is restored to the original objects.

Apply Action does not work with actions created from constraints, nor with compound action clips.

Renaming Actions

You can rename an action source by right-clicking on it in the explorer and choosing **Rename**. If you have already created clips that reference the action source, they are automatically renamed to reflect the new source name.

Deleting Stored Actions

To delete a stored action source:

1. In the explorer, make sure that **Show** > **Mixer** is on.

2. Select the action in the model's Mixer/Sources/Animation branch and press the Delete key.

   The action is removed and is no longer available for that model. In addition, any clips that reference the action are also removed.

Adding Action Clips

When you create an action clip, the stored animation drives the corresponding parameters over the range of frames covered by the clip.

1. Make sure that the model is open in the animation mixer.

   If you have not already done so, open the animation mixer in a viewport or floating window. Select the model or one of its children and click **Update** on the mixer's command bar. If this is the first time you have opened the model's track container, two animation tracks are created by default.

2. If you want to add the clip to a new track, first create a new track by right-clicking on any existing track and choosing **Add Track** > **Animation**. Clips cannot overlap on a track, so you can add clips to the same track to play them in sequence, but to mix (overlap) clips you must add them to separate tracks.

3. Do one of the following:

   - Right-click over the location on the track where you want the clip to start, and choose one of the model's stored actions from the **Load Source** menu. You can also use the **File** item to load a preset action created for another scene or import a SOFTIMAGE|3D animation file—see *Using Preset Actions* on page 264 and *Importing Animation Files* on page 288 for more information.

     or

   - Drag a stored action from the model's Mixer\Sources\Animation folder in the explorer onto a location on the track. You can also use this technique to copy an action from another model—see *Copying Actions Between Models* on page 263 for more information.

   A clip is created, representing an instance of the action. You can modify clips independently of each other without affecting the animation stored in the original action.

Visual Feedback

When a parameter is driven by an action in the mixer, its icon in the explorer is a plug. More information is available in the parameter's property page:

- If the parameter is driven by an action, its animation icon is a plug.

- If the parameter is driven by more than one clip, its icon is a plug with the letter *x*.
- If the parameter is driven by an action at the current frame, the icon is red. Otherwise, it is green.

Action Clips in the Mixer

Once you have added action clips to the animation mixer, you can use any of the mixer's features to move, copy, extrapolate, scale, trim, and blend them. For example:

- You can create sequences of actions by creating clips one after another using transitions to help smooth them.
- You can composite actions by adding clips for different parameters on the same frames of different tracks. For example, one clip could drive the legs of a character while a different clip drives the arm movement.
- You can mix actions that drive the same parameters by adjusting their clip weights and setting keys or other animation.

For more information about using the mixer in general, see *Chapter 10: The Animation Mixer* on page 231.

Sharing Actions between Models

Actions are reusable. You can create an action for one model and use it again to animate another model.

This is possible because each model maintains its own namespace. This means that two or more models in the same scene can contain objects with the same name. For example, a scene can have several characters each with their own left_arm, right_arm, and so on.

When you can drop an action created with one model onto a second model, the names are resolved automatically and the second model is properly animated.

If the objects in two models do use the same naming convention—for example, if one contains a child called l_arm and the other arm_left—you can still reuse actions by defining connection-mapping templates. Connection-mapping templates map the object and parameter names stored in an action to the objects and parameters available in a model. For more information, see page 268.

Whether or not you use connection templates, if you attempt to create an action clip and the source contains object and parameter names that cannot be matched to anything in the current model, the **Action Connection Resolution** dialog box opens. This is described in *Resolving Unconnectable Parameters* on page 265.

Copying Actions Between Models

Every model including the Scene_Root has its own mixer node. To use an action source that was created on a different model in the same scene, you must first copy it.

1. In the animation mixer, open the model to which you want to copy the action. To do this, select the model or one of its children then click the **Update** button in the mixer.
2. Open an explorer view and make sure that **Show** > **Mixers** is on.
3. Expand the Mixer\Sources\Animation node of the model containing the action, then drag the desired action from the explorer onto a green animation track in the mixer.

A copy of the action source is created in the model's Mixer\Sources\Animation folder, and a clip is instantiated on the track. If there is no match for some of the objects and parameter names in the source, the Action Connection Resolution dialog box opens—see *Resolving Unconnectable Parameters* on page 265.

Once the action has been copied, you can instantiate additional clips by right-clicking in an animation track and selecting the source name from the **Load Source** menu.

Using Preset Actions

You can save and load preset actions to share animation between models in different scenes.

You can only save presets for action sources. You cannot save presets for clips, compounds, or shape sources.

To save an action source as a preset

1. Display the action source's property editor by doing one of the following:
   - Right-click on a clip that references the source in the mixer, then choose **Source**.

     or

   - In the explorer view, make sure that **Show** > **Mixers** is on, then click the source's icon in the model's Mixer\Sources\Animation folder.

     or

   - Select the model or one of its children, then choose **Explore** > **Mixers** on the Selection panel and click the source's icon.

Save Preset

2. In the source's property editor, click the Save Preset icon. A browser opens.
3. Use the browser to specify a file name and directory, then click OK.

To load a preset action source

1. In the animation mixer, open the model into which you want to load the action. To do this, select the model or one of its children then click the **Update** button in the mixer.
2. Do one of the following:
   - On an animation track, right-click where you want a clip of the action and choose **Load Source** > **From File**. Use the browser that opens to find and select a saved action preset then click OK.

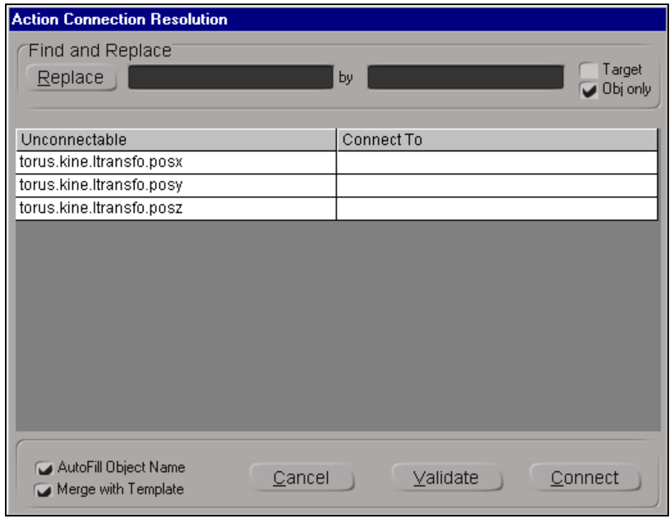
     or

   - Open a browser in a viewport or floating window, and drag an action from the browser onto an animation track in the mixer. On Windows NT, you can also drag actions from folder windows.

The action preset is added to the model's sources, and a clip is instantiated on the track. If there is no match for some of the object and parameter names in the source, the Action Connection Resolution dialog box opens—see *Resolving Unconnectable Parameters* on page 265.

Resolving Unconnectable Parameters

The **Action Connection Resolution** dialog box opens when you try to create an action clip and there is no match for some of the objects and parameter names in the source. For example, if an action source contains animation for an object called "torus" and there is no such object in the current model, the following would appear when you try to instantiate an action clip based on that source:



Use this dialog box to resolve the unconnectable parameters by specifying the existing parameters they should connect to. Once you have specified parameters, you can validate the replacements before connecting the animation.

Specifying Parameters

For each parameter in the **Unconnectable** column, type a new parameter in the **Connect To** column. You can type the entire scripting name of a parameter, or you can use the autofill features. To leave a source parameter unconnected so that it does not animate anything, leave the corresponding column blank.

Autofilling parameter names

When entering a parameter in the Connect To box, you can type just the object name and press the Tab key to automatically fill in the rest of the parameter name.

In the previous example, typing

```
cone
```

in the first row, then pressing the Tab key, results in

```
cone.kine.local.posx
```

Autofilling object names

When the **AutoFill Object Name** option is on and you replace one instance of an object name, all instances are replaced.

Continuing the previous example, when you type

```
cone
```

and press Tab, all rows are changed:

```
cone.kine.local.posx
cone.kine.local.posy
cone.kine.local.posz
```

Replacing Substrings

If you want to replace parts (substrings) of a name rather than an entire name, you must use the **Find and Replace** options.

Type a search string in the first text box, type a replacement string in the second text box, then click **Replace**. The first occurrence of the search string in each unconnectable parameter name is replaced. If **Obj Only** is on, the scope is limited to the object name before the first period, and the rest of the parameter name is not considered.

Iterative substring replacements

Sometimes you need to perform more than one find-and-replace operation on the parameter names. For example, you may need to map prefix_Object1_suffix to Object1, prefix_Object2_suffix to Object2, and so on.

If you replaced "prefix_" with an empty string, the objects in the **Connect To** column would be:

```
Object1_suffix. ...
Object2_suffix. ...
...
```

However if you then replace "_suffix" with an empty string, the objects in the **Connect To** column become:

```
prefix_Object1. ...
prefix_Object2. ...
...
```

This happens because the replacements are performed on the **Unconnectable** column. To stop it from happening, turn the **Target** option on. This allows the replacements to be performed on the **Connect To** column, allowing you to use successive find-and-replace operations yielding the desired result:

```
Object1. ...
Object2. ...
...
```

Validating

Before you click **Connect**, you can test your replacements by clicking **Validate**. Any parameter name in the **Connect To** column that is still not recognized is shown in red.

Connecting

Once you have specified new parameters for the unconnectable ones, click **Connect** to create the action clip. If any parameter names in the **Connect To** column are still unrecognizable, they are shown in red. However, you can leave rows blank—the corresponding parameters are unconnected and don't drive anything.

If **Merge with Template** is on, the **Unconnectable** and **Connect To** columns are saved as new rules in the active connection-mapping template. If no template is active, a new one is created. This means that the next time you create a clip with the same source parameters, these values are remembered and automatically used again. For more about connection-mapping templates in general, see the next section.

Connection-Mapping Templates

Connection-mapping templates are sets of rules used to convert object and parameter names in action sources to those that are available in a model. For example, an action might have been created to animate a model with an object called LEFTARM. To apply this action to another model with an object called left_arm, you can use a connection-mapping template.

Each model can have multiple connection-mapping templates, and each of these templates can be active or not. Connections are made when you instantiate an action clip. The rules from the first active template in the list are used. The order of templates is the order of their creation, as listed in the explorer or the **Templ** > **Cnx Template** menu in the animation mixer. You cannot change the connections once a clip is instantiated—you must instead delete the clip, change the template, then reinstantiate the clip.

If the object and parameter names in an action source are still unconnectable using the rules in the first active connection-mapping template, the Action Connection Resolution dialog box opens. For information about using this dialog box to connect parameters on the fly, see *Resolving Unconnectable Parameters* on page 265.

Creating Connection-Mapping Templates

You can create an empty connection-mapping template then add rules manually, or you can create a template based on objects in your scene.

After you create or edit a connection-mapping template, new clips use the first active template to establish their connections. Existing clips are not affected.

To create an empty connection-mapping template

There are two ways:

- Select an object in the scene, then on the Animate toolbar choose **Actions** > **Templates** > **Create Empty Cnx Template**. Don't be surprised if nothing appears to happen—an empty connection-mapping template was indeed created for the selected object's parent model.
- In the animation mixer, choose **Templ** > **Create Empty Cnx Template.** A template is created for the current model and the Connection Map property editor opens.

Once you have created an empty connection-mapping template, you can add and modify rules as described in *Editing Connection-Mapping Templates* on page 269.

Creating a Connection-Mapping Template Based on Objects

You can create a connection-mapping template that automatically maps the object names in a source hierarchy to a target hierarchy. This procedure works properly only when the two hierarchies have the same number of children and the same subtree structure.

To create an object-based connection-mapping template

1. Select the target; that is, select, branch-select, or tree-select the object or hierarchy you want to be animated.
2. From the Animate toolbar, choose **Actions** > **Templates** > **Create Connection Template**.
3. Pick the source object or hierarchy in object mode (left mouse button), branch mode (middle mouse button), or tree mode (right mouse button).

   A template is created by matching objects in the hierarchies according to their positions in their subtrees. You can fine-tune the template as described in the next section, *Editing Connection-Mapping Templates*.

Editing Connection-Mapping Templates

Once you have created a connection-mapping template, you can use the Connection Map property editor to add, edit, and delete rules. You can also use this editor to activate and deactivate templates, as well as save and load preset templates.

To open the Connection Map property editor for a model

1. Make sure the correct model is open in the mixer: if not, select the model or one of children and click **Update** on the mixer command bar.
2. Choose an item from the **Templ** > **Cnx Template** menu. The Connection Map property editor for the chosen connection-mapping template opens.

To see the rules better, you can resize the editor as well as the columns. To resize a column, move the pointer over the dividing line between column headings. Once the pointer changes to a double-headed arrow, click and drag to set a new width.

If there is no **Templ** > **Cnx Template** menu, then the current model has no connection templates. Choose **Templ** > **Create Empty Cnx Template** to create a blank one.

Selecting Rules

You can select a rule in the Connection Map property editor by clicking its label in the **Template** column (for example, **Rule 1**) or by placing the cursor in either text box in the corresponding row.

Adding Rules

To add a row for a new rule, click **Insert Rule** in the Connection Map property editor. The new rule is added immediately above the selected rule—this is important to remember because the rules are evaluated in order from top to bottom.

Editing Rules

You can edit rules in the Connection Map property editor by typing object and parameter names in the **Map From** and **Map To** columns. The rules work by substituting object and parameter names in the **Map From** column with those in the **Map To** column.

The rules use the same parameter names that are used in scripting—if you are not sure of a parameter's name, modify the parameter in a property editor and check the name that gets logged in the **SetValue** command in the Command Box at the lower left of the main window.

To leave the animation of a particular object or parameter unmapped, map it to an empty string or the <**unmapped**> token. To substitute part of an object or parameter name you must use special characters as described in *Substring Substitution* on page 272.

It is important to remember the following about how rules are applied:

- For each parameter contained in an action, the rules are evaluated in order (top to bottom) to see whether they can be applied.
- Once a rule is applied, the remaining rules are not evaluated and the next parameter in the action is processed. The exception is substring substitution—if a substring substitution rule is applied, the remaining rules are still evaluated (until, of course, a non-substring rule is applied).

Example 1: Mapping objects

Map from	Map to
torus	cone

The above rule maps any parameter on the object named **torus** to the corresponding parameter on the object named **cone**.

Example 2: Mapping parameters

Map from	Map to
torus.kine.local.posx	cone.kine.local.posy

The above rule maps the local X position of **torus** to the local Y position of **cone**.

Example 3: Leaving objects disconnected

Map from	Map to
torus	<unmapped>

The above rule maps all parameters on the object named **torus** to nothing. This rule could also have been entered in the following way:

Map from	Map to
torus	

Example 4: Leaving parameters disconnected

Map from	Map to
torus.kine.local.posx	<unmapped>

The above rule maps the local X position of **torus** to nothing. This rule could also have been entered in the following way:

Map from	Map to
torus.kine.local.posx	

Example 5: Using wIldcards

Map from	Map to
null	null*

The above rule maps all parameters on the object named **null** to all objects whose names begin with "null"; for example, null1, null2, and so on.

Validating Rules

Once you have entered rules in the Connection Map property editor, you can check their validity by clicking **Validate**. If an object or parameter name does not exist in the current model, it is shown in red.

Rules with wildcards, substring substitution, or the <**unmapped**> token are always shown in red.

Deleting Rules

To delete a rule, select it in the Connection Map property editor and click **Delete Rule**.

Substring Substitution

Substring substitution lets you create connection-mapping rules that change part of an object name. This allows the template to be reused in different contexts—you don't need to explicitly specify a particular object name. In addition, you can "clamp" the substring search so that it only replaces substrings at the beginning or end of the object name. These rules are most useful if you follow a strict naming convention for your objects and models. Substring substitution does not work for parameter names.

Once a substring substitution rule is applied to a parameter in an action, the remaining rules are still evaluated. This lets you perform multiple substring substitutions on the same parameter.

Substituting Substrings

To specify a substring to replace, enclose the characters within square brackets [ ] in the **Map From** column of the Connection Map property editor.

Example

Map from	Map to
[Object1]	Object2

This rule maps, for example, "prefix_Object1_suffix" to "prefix_Object2_suffix".

Substituting Substrings at the Beginning of an Object Name

You can restrict a substring substitution to the beginning of an object name. This is useful for changing a SOFTIMAGE|3D object's prefix without accidentally affecting the same substring if it occurs elsewhere. To do this, use a caret ^ as the first character within the square brackets.

Example

Map from	Map to
[^prefix_]	

This rule strips out "prefix_" from the beginning of object names, replacing it with nothing (an empty string).

Substituting Substrings at the End of an Object Name

Similarly, you can restrict a substring substitution to the end of an object name. This is useful for changing a SOFTIMAGE|3D object's suffix without accidentally affecting the same substring if it occurs elsewhere. To do this, use a dollar sign ($) as the last character within the square brackets.

Example 1

Map from	Map to
[$]	_suffix

This rule adds the string "_suffix" to the end of all object names.

Example 2

Map from	Map to
[suffixA$]	suffixB

This rule changes "suffixA" to "suffixB" in all object names.

Substituting Substrings in Parameter Names

Ordinarily, substring substitution works only on object names. To substitute substrings in parameter names as well as object names, use an at symbol @ as the first character within the square brackets.

Example

Map from	Map to
[@local]	global

This rule changes "local" to "global" in all object and parameter names.

Combining Substring Substitution and Other Rules

Substring substitution rules are applied before the other rules. This means that a rule can be applied to the result of a substring substitution.

Example

Map from	Map to
null.kine.global.posx	<unmapped>
[@local]	global

The substring substitution rule is applied first, mapping **null.kine.local.posx** to **null.kine.global.posx**. This result is then mapped to nothing, so that animation originally defined for **null.kine.local.posx** is left unconnected.

Activating and Deactivating Connection-Mapping Templates

Each model can have multiple connection-mapping templates, and each of these templates can be active or not. When you instantiate an action, the first active template is used to establish the connections. The order of templates is the order of their creation, as listed in the explorer or the **Templ > Cnx Template** menu of the animation mixer.

To activate or deactivate a template, toggle the **Active** option in its Connection Map property editor.

Deleting Connection-Mapping Templates

To delete a template, you must use an explorer:

1. Select a connection-mapping template. They are located in the Mixer/Templates/Connection_Mapping path of the model.
2. Press the Delete key, or right-click on the template and choose **Delete**.

Modifying Action Values

The animation values in actions often need to be modified. For example, you may need to offset an action in local space so that the animation occurs in a different location from where it was originally defined. Value maps let you do this. Each clip has its own local value map that can be modified independently of the others.

To automatically modify animation values in actions when clips are created, use a value-mapping template as described in *Value-Mapping Templates* on page 280.

When you copy a clip, its value map is not copied. In addition, you cannot save a preset. However, you can copy and paste value maps as described on page 279.

You can also use scripting commands to read and write value-map items. Use ***clipname*.actionclip** as the value of the **MappingTemplate** parameter with the **GetNumMappingRules**(), **GetMappingRule**(), and **SetMappingRule**() commands. See the HTML scripting reference (!) for more information.

Offsetting Action Values

You can use the Offset Map button to quickly create offset values for the animation in a clip.

1. Set the current frame to a frame that is controlled by the clip.
2. If any other clips drive the same parameters at that frame, mute their tracks.
3. Adjust the desired parameters; for example, move the object to a new position.
4. Make sure that the correct objects are selected and the correct parameters are marked.

Select all applicable objects explicitly. Branch and tree selections are not automatically expanded.

5. In the animation mixer, click the **Offset Map** button.

   Offsets are added to the clip's value map so that it reaches the new values at the current frame. You can see these offsets as described in *Editing Clip Data* on page 276. For example:

InstAction	Active	Src	Item	Expr
Item 1	✓	F01	torus.kine.local.posx	this+3.184062
Item 2	✓	F02	torus.kine.local.posy	this+4.495249
Item 3	✓	F03	torus.kine.local.posz	this+-0.449525

Original values are offset by different amounts.

Aligning Clip Sequences

You can also use the Offset Map button to align clips in sequence so that the second action starts where the first action ends.

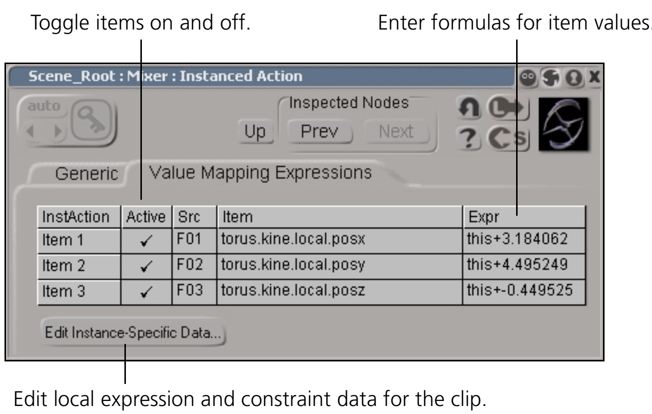
1. Make sure that the correct objects are selected and the correct parameters are marked.
2. Set the current frame to the last frame of the first clip. See *Clip Timing* on page 239.
3. Using the right mouse button, move to the first frame of the second clip.
4. In the animation mixer, click the **Offset Map** button. Again, this works only if no other clip drives the same parameters at that frame.

Editing Clip Data

In addition to using the Offset Map button to automatically create offsets for clip values, you can manually edit any of the clips' local data. You can modify offsets, create complex expressions involving cycle IDs and other tokens, and edit constraint and expression data.

To display a clip's value-map data

1. Right-click on a clip in the animation mixer and choose **Action Clip**.
2. Click the Value Mapping Expressions tab.



Modifying Clip Values

To modify the value of an animation item in a clip, enter a formula in the **Expr** column. You can use any of several tokens for predefined variables as well as any supported function or operator.

Variable Tokens

When writing formulas, there are several special tokens you can use to represent variables.

Token	Meaning
`this`	The original parameter value. This lets you create offsets of the original value rather than a new value entirely.
`cid`	The cycle ID number. If you have cycled or bounced an action clip in the animation mixer, the original clip has a cid of 0, the first cycle has a cid of 1, and so on. This lets you continuously increment or decrement a value based on the cycle; for example, so that a car moves forward with every wheel rotation.
`time`	The current local time of the clip's source, in seconds. If you cycle or bounce the clip, the values in the first cycle are repeated in the second, and so on.
`frame`	The current local frame of the clip's source. Again, if you cycle or bounce the clip, the values are repeated. Note that this is always an integer, even if the clip has been scaled and offset.
`gtime`	The current global time of the scene.
`gframe`	The current global frame of the scene.
`weight`	The current mix weight value of the clip.

Examples

Expression	Result
5	The corresponding parameter value is always 5, no matter what animation is contained in the action.
this + 5	5 is added to the parameter value of the action at the current frame.
(cid * 10) + this	The parameter value of the action is used for the duration of the original clip, then 10 is added for the first cycle, 20 is added for the second cycle, and so on.

Supported Operators and Functions

The expressions in value-mapping templates do not use exactly the same syntax as expressions in the expression editor. In value-mapping templates, you can use any operator or function supported by VBScript. The following tables list the most useful operators and functions.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Exponentiation
\	Integer division
Mod	Remainder
And	Logical conjunction
Imp	Logical implication
Not	Logical negation
Or	Logical disjunction
Xor	Logical exclusion

Function	Description
Abs()	Absolute value
Cos()	Cosine
Fix(), Int()	Integer portion of a number
Log()	Natural logarithm
Sgn()	The sign of a number: –1, 0, or 1
Sin()	Sine
Sqr()	Square root
Tan()	Tangent

For more information, see the Microsoft VBScript language documentation at **http://msdn.microsoft.com/scripting/vbscript/doc/vbstoc.htm**

Copying and Pasting Value Maps

You can copy and paste formulas in the **Expr** column of a clip's local value map. This is especially useful because the value map is not copied when you copy clips.

- Use Ctrl+x to cut.
- Use Ctrl+c to copy.
- Use Ctrl+v to paste.
- To copy or paste the entire column at once, first select it by clicking on the **Expr** heading.

Modifying Constraint and Expression Data

When an action contains animation data from a constraint or expression, each clip maintains its own local copy of the data. To edit it, select an item in the **InstAction** column then click the **Edit Instance-Specific Data** button. For more information, see *Chapter 4: Animating with Constraints* on page 107 or *Chapter 5: Animating with Expressions* on page 133.

Activating and Deactivating Animation Items

To toggle animation items on and off, click in the corresponding row of the Active column of a clip's local value map. When an animation item is off, the clip no longer drives the corresponding parameter.

Value-Mapping Templates

Value-mapping templates provide a way to automatically modify the animation values in an action when clips are instantiated. For example, you can add an offset so that two characters don't try to walk in the same place at the same time. Value-mapping templates are similar in many ways to connection-mapping templates, but there are also important differences.

Each model can have multiple value-mapping templates, and each of these templates can be active or not. The order of templates is the order of their creation, as listed in the explorer or the **Templ** > **Value Template** menu of the animation mixer. When you instantiate an action, the rules in the first active value-mapping template are copied locally onto the clip. You can modify this local value map as described in *Editing Clip Data* on page 276, but the changes affect only that particular clip—the next time you instantiate a clip, the original rules in the first active value-mapping template will be used again (unless you have changed them in the meantime).

Creating Value-Mapping Templates

You can create an empty value-mapping template then add rules manually, or you can automatically create a template that contains the offsets of the current transformation values between two objects.

To create an empty value-mapping template

Do one of the following:

- Select an object in the scene, then on the Animate toolbar choose **Actions** > **Templates** > **Create Empty Value Template**. Don't be surprised if nothing appears to happen—an empty value-mapping template was indeed created for the selected object's parent model.

  or

- In the animation mixer, choose **Templ** > **Create Empty Value Template**. A template is created for the current model and the Value Map property editor opens.

Once you have created an empty value-mapping template, you can add and modify rules as described in *Editing Value-Mapping Templates* on page 281.

Creating a Value-Mapping Template Based on Objects

You can automatically create a value-mapping template that contains the offsets of the current transformation values between two objects or hierarchies. This procedure works properly only when the two hierarchies have the same number of children and the same subtree structure.

To create an object-based value-mapping template

1. Select the target, that is, the object or hierarchy whose animation will be offset from the original action.
2. From the Animate toolbar, choose **Actions** > **Templates** > **Create Value Template**.
3. Pick the source object or hierarchy in object mode (left mouse button), branch mode (middle mouse button), or tree mode (right mouse button).

   A template is created by matching objects in the hierarchies according to their positions in their subtrees with their corresponding transformation offsets.

Once you have created a template in this way, you can fine-tune it as described in the next section, *Editing Value-Mapping Templates*.

Editing Value-Mapping Templates

Once you have created a value-mapping template, you can use the Value Map property editor to add, edit, and delete rules. You can also use this editor to activate and deactivate templates, as well as save and load preset templates.

To open the Value Map property editor for a model

1. Make sure the correct model is open in the mixer; if not, select the model or one of descendants and click **Update** on the mixer command bar.
2. Choose an item from the **Templ** > **Value Template** menu. The Value Map property editor for the chosen value-mapping template opens.

To see the rules better, you can resize the editor as well as the columns. To resize a column, move the pointer over the dividing line between column headings. Once the pointer changes to a double-headed arrow, click and drag to set a new width.

If there is no **Templ** > **Value Template** menu, then the current model has no value templates. Choose **Templ** > **Create Empty Value Template** to create a blank one.

Selecting Rules

You can select a rule in the Value Map property editor by clicking its label in the **Template** column (for example, **Rule 1**) or by placing the cursor in either text box in the corresponding row.

Adding Rules

To add a row for a new rule, click **Insert Rule** in the Value Map property editor. The new rule is added immediately above the selected rule—this is important to remember because the rules are evaluated in order from top to bottom.

Editing Rules

You can edit rules in the Value Map property editor by typing parameter names in the **Parameter** column and mathematical expressions in the **Expression** column. The rules use the same parameter names that are used in scripting—if you are not sure of a parameter's name, modify the parameter in a property editor and check the name that gets logged in the **SetValue** command in the Command Box at the lower left of the main window.

The rules work by applying the formula in the Expression column to the value of the corresponding parameter given by the action at a given frame. The value-mapping rules are applied after the connection-mapping rules, so if you have a connection-mapping template that maps **torus** to **cone**, your value-mapping rules should affect the parameters of **cone** not **torus**.

See the following sections for more details:

- *Variable Tokens* on page 277
- *Supported Operators and Functions* on page 278

Validating Rules

Once you have entered rules in the Value Map property editor, you can check their validity by clicking **Validate**. Invalid rules are shown in red so that you can correct them. A rule can be invalid if a parameter is misspelled or if the expression is not valid.

Deleting Rules

To delete a rule, select it in the Value Map property editor and click **Delete Rule**.

Activating and Deactivating Templates

Each model can have multiple value-mapping templates, and each of these templates can be active or not. When you instantiate an action, the value-mapping rules of the first active template are copied locally onto the clip. The order of templates is the order of their creation, as listed in the explorer or the **Templ** > **Value Template** menu of the animation mixer.

To activate or deactivate a template, toggle the **Active** option in its Value Map property editor.

Deleting Value-Mapping Templates

To delete a template, you must use an explorer:

1. Select a value-mapping template. They are located in the Mixer/Templates/Value_Mapping path of the model.
2. Press the Delete key, or right-click on the template and choose **Delete**.

Modifying Action Sources

After you have created an action source, you can modify the animation data stored in it. You can modify stored function curves interactively by saving action keys, or you can modify any animation data manually using the action source's property editor.

Saving Action Keys

You can interactively add keys to function-curve animation stored in an action source by saving action keys. All clips that reference the action source are affected.

1. Set the current frame to a frame that is controlled by a clip that instantiates the action source.

2. If any other clips drive the same parameters at that frame, mute their tracks.

3. Adjust the desired parameters; for example, move the object to a new position.

   The colors of the animation controls may be incorrect in the Animation panel and in property editors; they do not properly reflect whether there is a key at the current frame. This is because the keys are set in the local time of the action rather than the global time of the scene.

4. Make sure that the correct objects are selected and the correct parameters are marked.

   Select all applicable objects explicitly. Branch and tree selections are not automatically expanded.

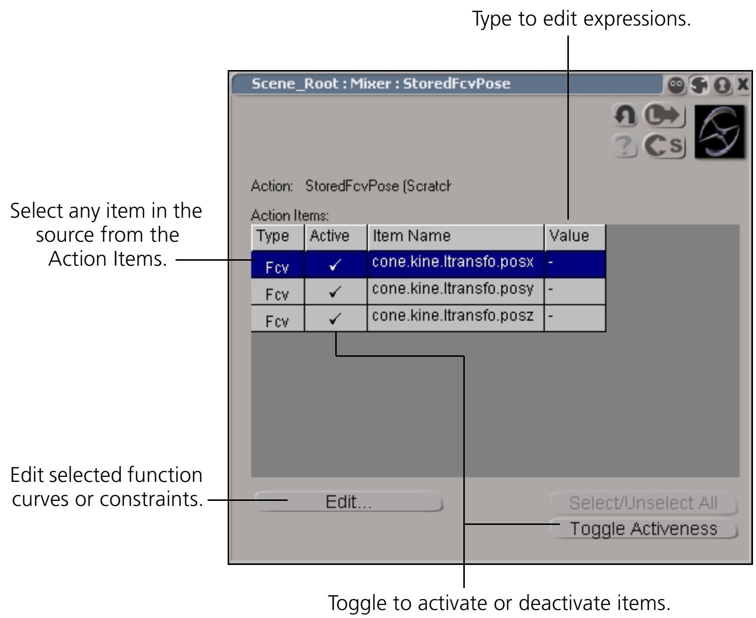
5. In the animation mixer, click the **Action Key** button. New keys are added to the action source's function curves at the corresponding local time, taking into account any scaling or warping of the clip.

   When you save an action key, offsets (created with Offset Map or manually) are not taken into account. If there is an offset, values may jump when you click **Action Key**.

   If the action is a static pose created with **Store** > **Transformation - Current** or **Marked Parameters - Current**, click the **Offset Map** button instead of Action Key to change the source values.

Viewing Action Source Properties

To view an action's source properties, right-click on a clip and choose **Source**. The action source appears in a property editor. You may need to resize the editor.



Each row in the Action Items list represents a parameter or degree of freedom controlled by the source:

- **Type** is the type of animation:
  - **Fcv:** function curve (keys)
  - **Exp:** expression
  - **Cns:** constraint
  - **CMP:** items in a compound action clip
  - **Key:** shape key
  - **SHP:** items in a compound shape clip
- **Active** shows whether the item contributes to the animation.
- **Name** shows the parameter that is driven by an animation item.
- **Value** shows the result of the item.
  - For function curves, this is blank.
  - For expressions, this is the expression itself.
  - For constraints, this is the type of constraint and the name of the constraining object.

Selecting Action Items

To select an item, click on it in the Action Items list. Ctrl+click to toggle the selection status of one or more items.

Activating and Deactivating Source Items

You can deactivate an item in the Action Items list so that it no longer contributes to the animation. Select the item and click the **Toggle Activeness** button or click in the **Active** column of the Action Items list to clear the corresponding check mark. Repeat this procedure to reactivate an item.

Activating and deactivating items in a source applies to subsequent clips that you instantiate from that source. However, clips that already exist are unaffected—you must delete the clips and create them again.

Modifying Source Function Curves

To modify the function curves in a source action, select its row and click **Edit Source Data**. The source's function curves open in the animation editor. Changes to these curves are reflected in all existing and future clips based on that source.

You can also mark parameters and change them interactively (for example, by translating an object in a geometry view), then press the **Action Key** button in the animation mixer. This works only if there is exactly one action controlling the object at the current frame.

Modifying Source Expressions

To modify expressions in a source action, type directly in the **Value** box. The changes affect subsequent clips that you instance from that source. However, clips that already exist are unaffected—you must delete the clips and create them again.

Modifying Source Constraints

To modify a constraint in a source action, select its row and click **Edit Source Data**. The Constraints property editor opens. Any changes affect subsequent clips that you instance from that source. However, clips that already exist are unaffected—you must delete the clips and create them again.

Adding Animation Items

With scripting, you can use the **AddPoseActionItems**() command to add 9 dummy animation items to an action source. Typically, these extra channels are to allow easy connection to a parent "offset" object for control over matching the end pose of the root object of an action with the start pose of the next action in a non-destructive fashion. See the HTML scripting command reference (!) for more information.

Fill Actions

Fill actions control the values of parameters that are not being driven by any other action. They provide a way to control the background animation.

Normally, nothing drives parameters in the gap between action clips. Consequently, the parameters are not updated when you jump to a frame in the gap. As a result, a parameter can have different values at the same frame, depending on its value at the last frame that was current. A similar situation can happen even if there is no gap but different clips drive different parameters or some parameters have been deactivated.

Fill actions solve these problems. They are similar to other action clips, but they are never used in any animation mixing. Their values are used only when no other action is driving a given parameter.

You can have multiple fill actions. The value used for a parameter depends on the fill actions' relative priorities (as well as whether a track has been muted).

Creating Fill Actions

To create a fill action:

1. Instantiate a clip of the source you want to use for the fill; for example, a reference pose. For more details about creating clips in general see *Clips* on page 239.
2. Scale the clip to the desired length.
3. Right-click over the clip and choose **Toggle Fill**.

   Fill actions are displayed in a light green with a black outline.

   To change a clip from a fill action back to an ordinary action, choose this command again.

Setting Fill Priority

For fill actions, the weight-mixer panel is used to set the priority:

1. Make sure that the weight-mixer panel is displayed (**View** > **Weight Mixer Panel**).
2. Enter an integer value in the number box.

When there are two or more fill actions for the same parameter, the one with the highest priority is used. If two or more fill actions have the same priority, the one that was instantiated last is used. You can temporarily prevent a fill action from being used by muting its track. You can also activate and deactivate individual parameters of a clip as described in *Activating and Deactivating Animation Items* on page 279.

Fill Properties

You can also set an action's fill properties on the clip's property editor:

1. Right-click on the clip and choose **Action Clip**.
2. Click on the Generic tab and then on the Instanced Action tab.
3. Adjust the values in the **Fill Controls** box.

Combining Function Curves and Expressions

You can also use an action to combine a function curve and expression on the same parameter. At each frame, the clip reads the value from the function curve and then alters it according to the expression.

1. Create a self-referential expression. For example:

Target	Expression
`cone.kine.pathcns.perc`	`cone.kine.pathcns.perc * 4`

2. Store an action source for the parameter and remove the animation.
3. Animate the parameter by setting keys.
4. Instantiate a clip based on the source you created.

Importing and Exporting Actions

You can import and export actions as SOFTIMAGE|3D animation files. This lets you exchange animation data with SOFTIMAGE|3D scenes.

Actions in SOFTIMAGE|XSI are not the same as actions in SOFTIMAGE|3D. In SOFTIMAGE|3D, actions are elements within **.ani** files. In SOFTIMAGE|XSI, actions are separate elements.

Importing Animation Files

You can import an **.ani** file using either the Animate toolbar or the animation mixer. When you import an .ani file, an action source is created from which you can instantiate clips. If you use the animation mixer to import, you can also instantiate a clip at the same time as creating a source.

You can import the following types of files:

- ***.ani:** SOFTIMAGE|3D animation files.
- ***.key:** SOFTIMAGE|3D structure keys (static poses).
- ***.exp:** expressions in plain text format. These consist of a SOFTIMAGE|XSI object and parameter name, followed by an equality symbol and a valid expression. For example:

```
sphere.kine.local.posy = 5 + cos( 10 * torus.kine.local.posy )
```

  These are not the same as SOFTIMAGE|3D binary *.exp files.

Importing Animation Files Using the Animate Toolbar

When you import an **.ani** file using the Animate toolbar, an action source is created.

1. Select an element in the model to which you want to import animation.
2. From the Animate toolbar, choose **Tools** >**Sources** > **Import Action from SI3D Anim File**. A browser opens.
3. Navigate to the ANIMATION subdirectory of a SOFTIMAGE|3D database, then select an **.ani** file and click OK.

An action source is created in the Mixer\Sources\Animation folder of the corresponding model. By default, the action source name is based on the animation file name, but you can rename it by right-clicking on it in an explorer view and choosing **Rename**.

Once you have imported a animation file as an action source, you can instance action clips based on that source by right-clicking on a track in the animation mixer then selecting the source name from the **Load Source** menu. For more information about creating clips in general, see *Adding Action Clips* on page 261.

Importing Animation Files with the Animation Mixer

You can import **.ani** files from the animation mixer. This procedure creates an action source in the Mixer\Sources\Animation folder of the current model and also instantiates a clip from that source.

1. In the animation mixer, right-click on an animation track at the frame where you want the instanced action clip to begin.
2. From the pop-up menu, choose **Load Source** > **From File**. A browser opens.
3. Navigate to the ANIMATION or STRUCT_KEYS subdirectory of a SOFTIMAGE|3D database, then select an animation file and click OK.

An action source is created in the Mixer\Sources\Animation folder of the current model, and a clip is instantiated on the track. The default names of the source and clip are based on the **.ani** file name.

Once the action is imported, you can instantiate additional clips by right-clicking in an animation track and selecting the source name from the **Load Source** submenu.

You can also create an action source and instantiate a clip by opening a browser view and dragging an .ani file onto an animation track in the mixer. On Windows NT, you can also drag an **.ani** file from a folder window or Windows Explorer.

Exporting Actions Files

You can export an action source to a SOFTIMAGE|3D animation file. Actions with function curves are exported as *.ani files, and actions with static values are exported as *.key files. After exporting, you can then load the animation file into SOFTIMAGE|3D.

You can export only actions based function curves or static values. You cannot export actions with expressions, constraints, compound clips, or cluster keys (shapes).

In addition, you cannot export actions for parameters that have no equivalent in SOFTIMAGE|3D. For example, you cannot export an action for **grid.twist.angle**.

1. Select an element in the model from which you want to export an action, then choose **Explore** > **Mixers** from the Selection panel and click the name of the action source.

   Alternatively, open an explorer view, make sure that **Show** > **Mixers** is on, expand the Mixer\Sources\Animation node of the appropriate model, and click the name of the action source.

2. From the Animate toolbar, choose **Tools** > **Sources** > **Export Action to SI3D Anim File**. A browser opens.

3. Navigate to the ANIMATION or STRUCT_KEYS directory of a SOFTIMAGE|3D database, enter a file name and click OK.

You can only export files to the correct directory of a SOFTIMAGE|3D database. If you select another location, no file is exported.

Chapter 12 Shape Animation

Shape animation (sometimes called *morphing*) lets you animate the geometrical structure (topology) of an object using clusters of points. You create clusters and then animate them in different positions to create key shapes.

For each variation in the shape you make, you can save a key shape for it to create a key-shape library. To animate these shapes, you select key shapes in a certain order and SOFTIMAGE|XSI interpolates the shapes between them.

In SOFTIMAGE|XSI, shapes correspond to cluster keys in SOFTIMAGE|3D. SOFTIMAGE|XSI has no equivalent to what are called shape keys in SOFTIMAGE|3D.

Overview of Shape Animation Techniques

There are two basic ways to create shape animation:

- Create an object and deform it, saving or storing shape keys along the way. You can reuse existing shape keys as the basis for new ones.
- Create a series of similar objects deformed in different ways and select them as shape keys. This sets up a relation between the objects and the shape keys, allowing you to fine-tune the keys by modifying the original models. The objects you are deforming must have the same number of points.

You can then apply your shape keys as clips on tracks in the animation mixer. The animation mixer lets you apply the shape keys in sequence, create transitions, and mix shapes using weights. For more information about the animation mixer, see *Chapter 10: The Animation Mixer* on page 231.

Shape Sources and Clips

Shape keys have two parts: sources and clips. A **source** is simply a shape. By storing several shapes for an object, you build up a library of sources.

A **clip** is an instance of a source at a particular position along a track in the animation mixer—although you don't need to open the animation mixer to create shape animation. You can create multiple clips of the same source, thereby returning to the same shape several times in the same animation.

Different things happen depending on whether you store, apply, save, or select shape keys:

- When you **store** a shape key, you simply create a source.
- When you **apply** a shape key, you create a clip (instance) of the selected shape source at the current frame.
- When you **save** or **select** a shape key, you create a source as well as a corresponding clip at the current frame.

Clusters and Compound Clips

Shape animation is stored per cluster. You can shape animation on multiple clusters on the same object. In the animation mixer, a separate container (compound clip) is created for each animated cluster. This lets you easily manipulate the entire animation of a cluster at once; for example, you can scale it or offset it using the compound clip.

If clusters overlap—that is, if certain components belong to more than one cluster—you can control the relative weight of each cluster's animation on the shared components. See *Mixing Shape Animation on Multiple Clusters* on page 315.

If you select a different cluster or set of points, the shape keys are stored for the other cluster and instantiated in a different compound shape action. Use the + icon or the **Cluster** menu on the Selection panel to help select the correct cluster.

Shapes and Models

Before you start to animate shapes, it's a good idea to create a model. Select the objects and choose **Create** > **Model** on the Model toolbar. This parents the objects under a model node and creates a mixer node for the model. This way, the shape keys are stored within the model rather than within the entire scene. You can then easily reuse the model with its shape keys in another scene. For more information about models, see *Models* in Chapter 5 of the *Fundamentals* guide.

If you duplicate a model in branch mode, or if you export and import a model, all shape keys it contains are connected as appropriate. However, if you duplicate something with shape animation without duplicating the entire model, the duplicated object has no shape animation.

Shape References

The first time you save, select, or store a shape key for a cluster, a Cluster Shape Combiner node is added to the operator stack. This defines the reference shape for that cluster. All subsequent shape keys are stored as the difference of point positions from this reference.

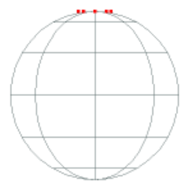
If you modify an operator that occurs earlier in the stack, the reference is also modified. For example, if you apply a Twist deformation and then save a shape key, the twisted shape is used as the reference. If you later animate the twist angle, the reference shape is also animated.

If you move points or deform an object after you apply shape animation, the operators are applied on top of the shape animation. However, they are removed when you instantiate another shape clip (that is, save, select, or apply a shape key, or load a shape clip in the mixer). Otherwise, they would be applied twice: once as part of a shape key and again as an operator in the history. If you do not want these deformers removed, mute them before instantiating any shape clips.

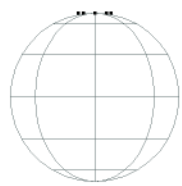
A Brief Example of Shape Animation

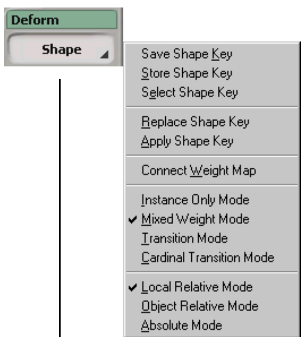
To become familiar with the concepts behind shape animation, follow this quick example.

1. Get a primitive surface sphere (**Get** > **Primitive** > **Surface** > **Sphere**) and tag the top points (**t** supra key).



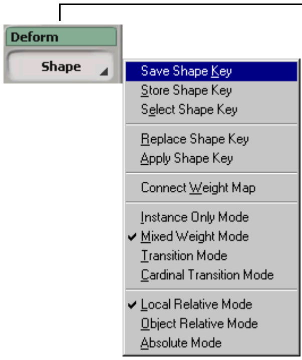
2. Create a cluster by clicking the **Cluster** button on the Edit panel.



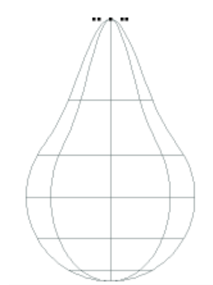


3. On the Animate toolbar, choose **Deform** > **Shape** and make sure that **Mixed Weight Mode** is on. The various shape key modes are described in *Understanding Shape-Key Modes* on page 298.

4. Save a shape key for the cluster in its current shape by choosing **Deform** > **Shape** > **Save Shape Key.**



5. Create a pear shape by translating the cluster in Y using the **v** supra key.

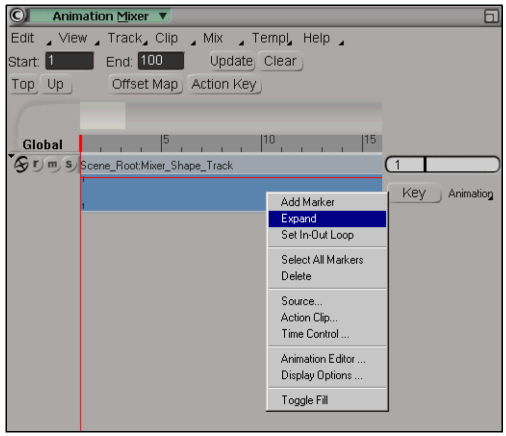


6. Go to frame 10 and choose **Deform** > **Shape** > **Save Shape Key** again, or middle-click on **Deform** > **Shape** to repeat the last command from that menu.

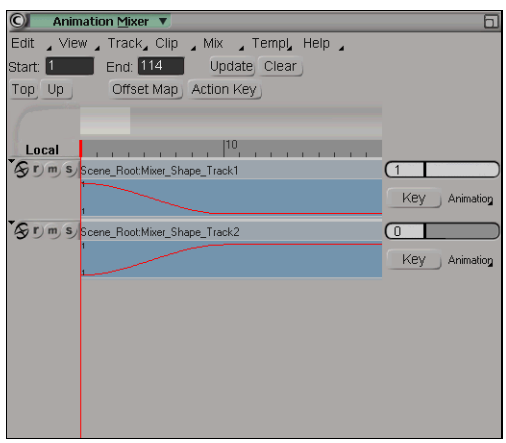
7. Play back the scene to see the shape animation. The sphere changes to a pear shape over ten frames.

8. Now take a peek under the hood to see what's happening. In viewport C, change the view to the animation mixer then click the **Update** button on the mixer's command bar.

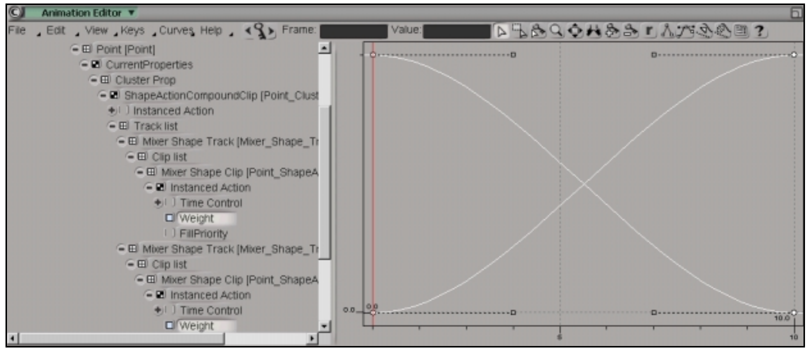
9. Right-click on the dark-blue compound shape clip and choose Expand. Alternatively, you can double-click on the clip.



10. You should see two shape clips on separate tracks—one for each shape key you saved. A curve represents the weight of each shape key. You can use the Weight Mixer Panel at the right to adjust the weights and set keys.



11. You still have access to the animation editor to edit the weight function curves. In viewport C, change the view to the animation editor.



Now you're acquainted with the basics of shape animation. The rest of this chapter expands on these concepts and describes the many options and techniques available.

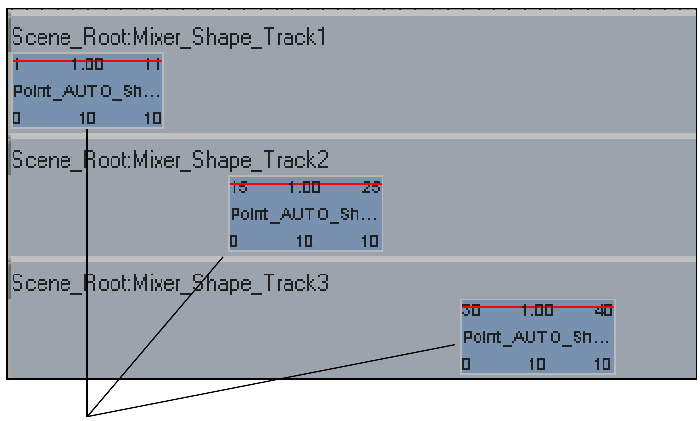
Understanding Shape-Key Modes

Shape-key modes let you control shape animation in different ways. There are four shape-key modes that affect what happens when you save, apply, or select shape keys: Instance Only, Mixed Weight, Transition, and Cardinal Transition.

Each of these modes can be chosen from the **Deform** > **Shape** menu on the Animate toolbar. A check mark indicates which mode is currently active. Once you save, apply, or select a shape, you cannot change the shape key mode for that cluster.

Instance Only Mode

Instance Only Mode creates shape clips that are 10 frames long. They are instantiated on separate tracks within a compound shape action. Because the clips are 10 frames long, they will mix with each other if you set keys that are less than 10 frames apart—you can adjust this in the mixer.

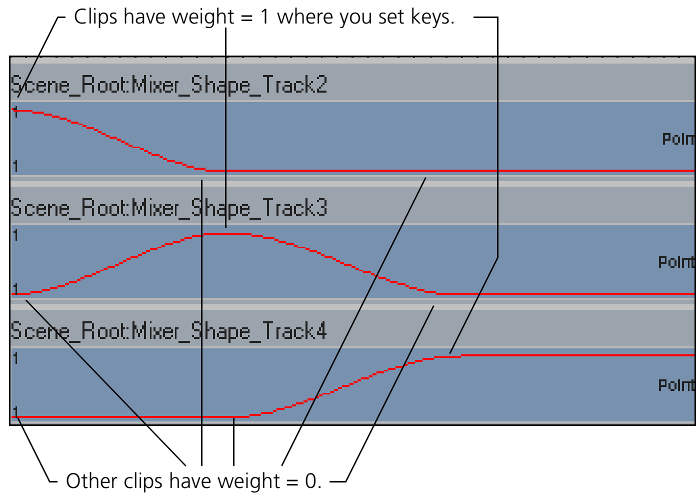


10-frame clips. Resize them and add transitions or mix weights.

In the mixer, you can scale the clips to any size then manually add transitions or mix weights. You can scale a clip down to one frame interactively by dragging its edges. To scale it to zero frames, right-click on it then choose **Time Control** and set the **Source Clipping In** and **Out** values to 0. To see the name of a zero-frame clip, make sure that **Tooltips** is active on the Clips page of the Animation Mixer Preferences dialog box. For more information about the animation mixer view preferences, see *View Options* on page 236.

Mixed Weight Mode

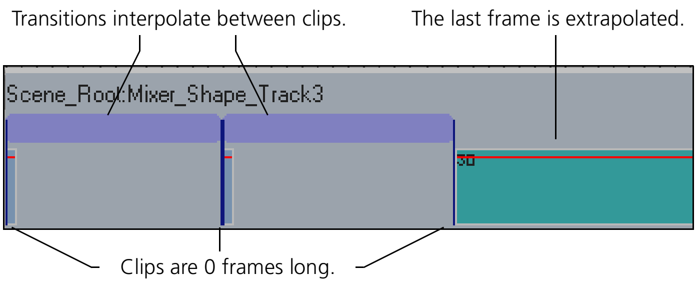
Mixed Weight Mode creates shape clips that are the same length as your scene. They are instantiated on separate tracks with default weight curves. This is useful if you want to control shapes by mixing the weights of different shapes.

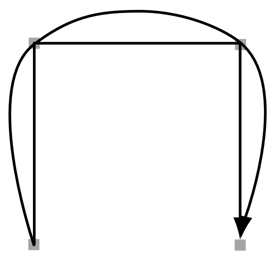


Transition Mode & Cardinal Transition Mode

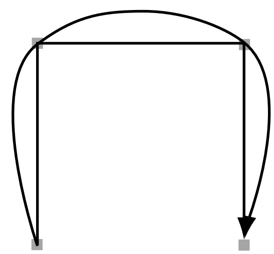
Transition Mode and Cardinal Transition Mode are similar. They both create shape clips that are zero frames long. The clips are instantiated on the same track, and transition curves are automatically generated between them.

- **Transition Mode** creates transitions with a linear interpolation in space, moving directly from one shape to the next.
- **Cardinal Transition Mode** creates transition with a cardinal interpolation in space. This results in a smooth, arcing transition that still passes through each shape key at the defined key frames. With only two shape keys, the transition is still linear; you need at least three keys for a cardinal curve, and the curve is fully defined when there are at least four keys. The interpolation on any segment is influenced by the four nearest keys.





Transition: Points follow a linear path between key positions.



Cardinal Transition: Points follow a curved path between key positions.

Understanding Shape Reference Modes

Shapes are stored as the differences of point positions from the reference. The reference is the shape of the cluster the first time you save, store, or select a shape key. By setting the shape reference mode, you have further control over how the shape behaves when the object is deformed further.

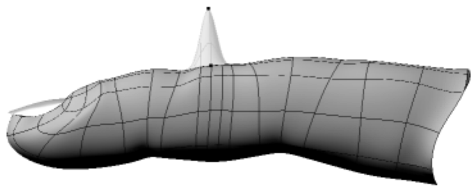
There are three different shape reference modes available from the **Deform** > **Shape** menu on the Animate toolbar:

- **Local Relative Mode—**Shapes are stored relative to the each point's local reference frame. When you deform the surface, the shape maintains its position and orientation relative to its immediate surroundings. For example, if you use shapes to animate a mole on a finger and then bend the finger, the mole moves with the finger as you would expect. This mode is the default, but it also requires the most computation.

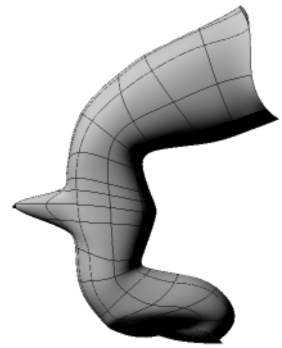
In some cases it may be necessary to freeze the object's transformations before using local relative mode.

- **Object Relative Mode—**Shapes are stored relative to the object as a whole. If you deform the object, the shape moves with it but maintains its original orientation. If there is no deformation animation before you save or store shape keys, the effect is the same as local relative mode.
- **Absolute Mode—**Shapes are absolute and not relative. They are stored in terms of the local coordinate system of the object. If the object is deformed, the cluster with the shape is unaffected; the effect is similar to a "lockdown." Note that if you use shape animation on an envelope, this mode will overwrite the envelope deformation.

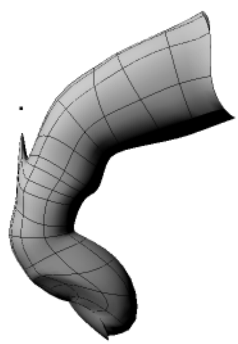
Once you have started saving or storing shape keys on a particular cluster, you cannot change the shape reference mode for that cluster.



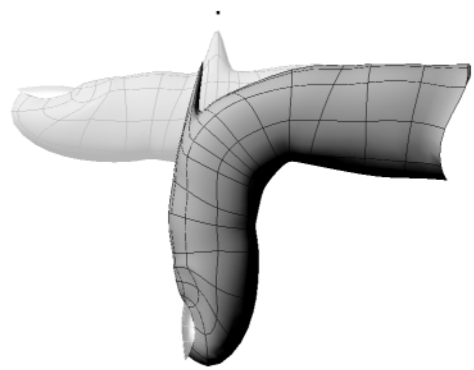
Shape animation on a single cluster



Local Relative Mode:
Shape deforms with object.



Object Relative Mode: Shape deforms with object but keeps original direction.

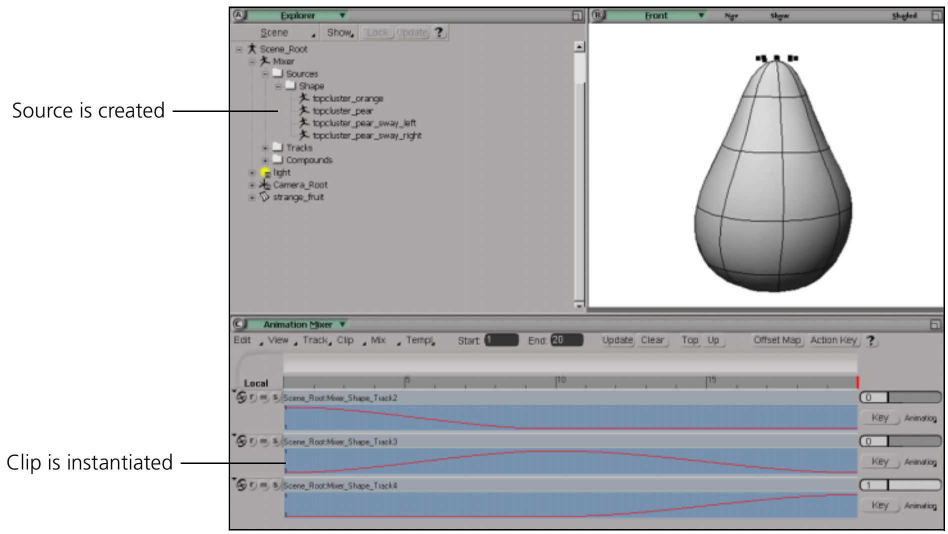


Absolute Mode: Shape stays locked in place as object deforms.

Saving Shape Keys

When you save shape keys, two operations are performed:

- A shape key is created for the current shape and added to the model's list of shape sources.
- An instance of the shape key is added as a clip on a track in the animation mixer. The track is contained in a compound clip corresponding to the cluster.



This is the equivalent of storing a shape key and then immediately applying it at the current frame.

If you do not want your object to ease in and ease out of each shape, you can set your preference before saving any shape keys. From the main-menu bar, choose **File** > **User Preferences**. Click the **Animation** tab and make sure that **Default Interpolation for Inserted Keys** is set to **Linear**.

To save shape keys

1. Select a shape-key mode from the **Deform** > **Shape** menu on the Animate toolbar. The shape key-modes determine how the shape clips are instantiated when they are automatically added to the animation mixer. For more information, see *Understanding Shape-Key Modes* on page 298.
2. Select a shape reference mode from the **Deform** > **Shape** menu on the Animate toolbar. The shape reference mode determines how the shape information is stored. For more information, see *Understanding Shape Reference Modes* on page 301.

3. Select a cluster of points. If you select multiple points instead of a cluster, the shape will be saved for the smallest cluster that contains all those points; if no such cluster exists, a new cluster will be created automatically. If you select an object instead of either a cluster or multiple points, a cluster will be created for all the points on the object. Note that you cannot save shape keys if multiple clusters are selected.

4. Change the current frame, then deform and transform the cluster into a new shape.

5. Choose **Deform** > **Shape** > **Save Shape Key** from the Animate toolbar.

If you are on a frame with an existing shape key, you are prompted to confirm whether you want to replace it:

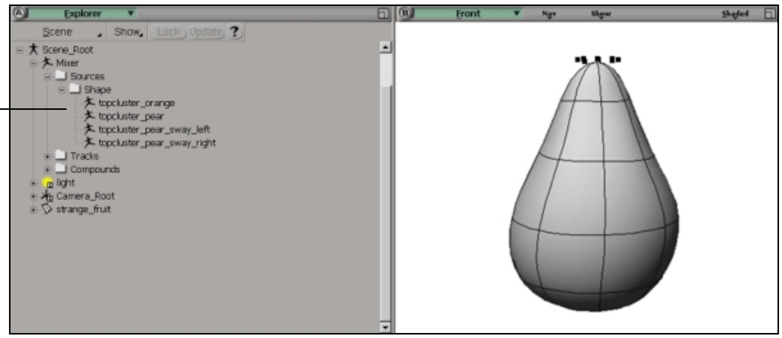
- If you click Yes, the new clip replaces the old clip.
- If you click No, the new clip is added and mixed with the old clip.
- If you click Cancel, no new clip is created.

6. Repeat steps 4 and 5 as necessary.

Storing Shape Keys

When you store shape keys, a shape key is created for the current shape and added to the model's list of shape sources. However, the shape key is not automatically instantiated in the animation mixer. Instead, you can later apply it manually.

Source is created but no clip is instantiated.



To store a shape key

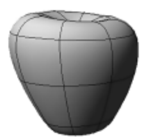
1. Select a shape reference mode from the **Deform** > **Shape** menu on the Animate toolbar. The shape reference mode determines how the shape information is stored. For more information, see *Understanding Shape Reference Modes* on page 301.
2. Select a cluster of points. If you select multiple points instead of a cluster, the shape will be saved for the smallest cluster that contains all those points; if no such cluster exists, a new cluster will be created automatically. If you select an object instead of either a cluster or multiple points, a cluster will be created for all the points on the object. Note that you cannot store shape keys if multiple clusters are selected.
3. Deform and transform the cluster into a new shape.
4. From the Animate toolbar choose **Deform** > **Shape** > **Store Shape Key**. A source is added to the model's Mixer/Sources/Shapes branch.
5. Repeat steps 3 and 4 as necessary to create a library of stored shapes.

Once you have created a library of stored shapes, you can apply them as described on page 308 or using the animation mixer as described on page 310.

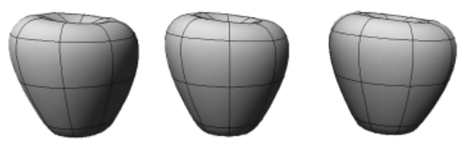
Selecting Shape Keys

To create shape animation by selecting shape keys, you first create a series of similar objects deformed in different ways. The objects must have the same type of geometry (polygon mesh or surface) and the same topology (number and arrangement of points). The easiest way to do this is to duplicate the model you want to deform. You then select the objects as shape keys for the model—this sets up a relation between the objects and the shape keys, allowing you to fine-tune the keys by modifying the original deformed models.

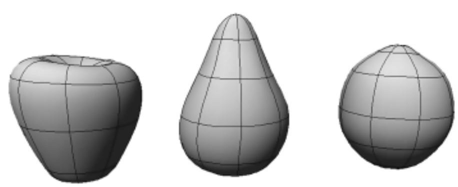
1. Create the model you want to deform.

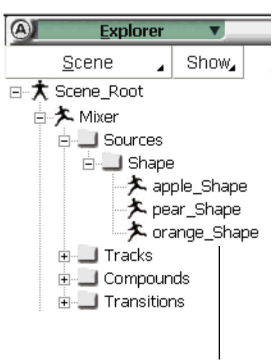


2. Duplicate this model as many times as the number of the key shapes you want to use. Move them out of the way of the camera.



3. Deform the copies using any method. You can even animate the deformations, for example, using Cluster Center.



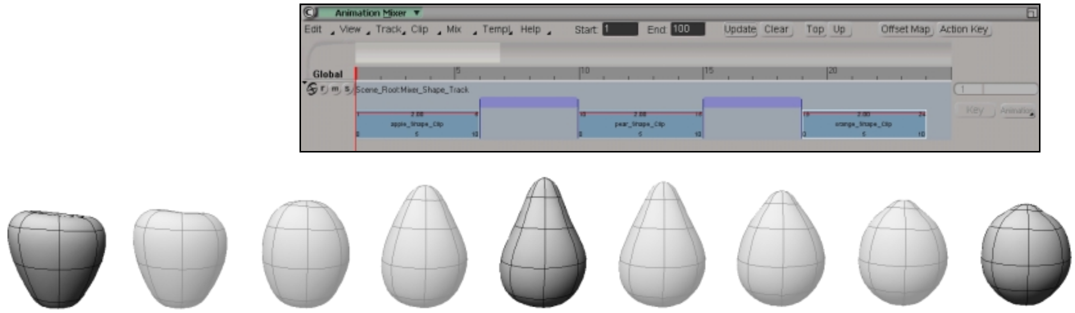


Sources added to mixer

4. Select the model you want to deform.
5. Choose **Shape** > **Select Shape Key** from the Animate toolbar, then pick each of the deformed models. Use the left mouse button to pick objects and the middle mouse button to pick branches.

   When you have finished picking shapes, click the right mouse button.

For each shape you pick, a source is added to the model's Mixer/Sources/Shapes branch. You can load the model into the animation mixer, where you can instantiate the source shapes as clips before sequencing and mixing them.



Instantiate the sources you selected as clips in the mixer.

Fine-Tuning Selected Shape Keys

When you select shape keys, a relationship is established between the shape keys and the deformed objects you picked. If you modify the deformed objects, the changes are automatically reflected in all applied instances of the corresponding shape key. This lets you fine-tune the shape animation by adjusting the original shapes.

Breaking the Link to Selected Shape Keys

You can break the relationship between the shape-animated object and the deformed objects that you selected as shape keys. Once you have done this, changes to the original deformed objects no longer affect the shape animation.

1. In the explorer, make sure that **Show** > **Clusters**, **Show** > **Primitives** and **Show** > **Operators** are on.
2. Expand the object with shape animation, then expand its operator stack, its Clusters container, and the cluster associated with the shape animation (by default, this cluster is called **Point_AUTO**).
3. Expand the cluster key associated with one of the deformed objects you selected as a shape key. By default, the cluster key name begins with the same name as the original deformed object and ends with **_Shapekey**.
4. Delete the **Copy Shape** operator under the cluster key by selecting its name and pressing the Delete key.
5. Repeat this procedure for each of the original deformed objects whose relationship you want to break.

Applying Shape Keys

Once you have created shape keys, you can apply them without opening the animation mixer. You can apply shape keys that you have stored or selected, as well as those that you have saved if you want to return to a previous shape.

1. Select a shape-key mode. The shape-key modes determine how the shape clips are instantiated when they are automatically added to the animation mixer—see *Understanding Shape-Key Modes* on page 298.
2. Set the current frame.
3. In the explorer view, select the shape key in the model's Mixer/Sources/Shape branch.
4. From the Animate toolbar, choose **Deform** > **Shape** > **Apply Shape Key**.

If you are on a frame with an existing shape key, you are prompted to confirm whether you want to replace it.

- If you click Yes, the new clip replaces the old clip.
- If you click No, the new clip is added and mixed with the old clip.
- If you click Cancel, no new clip is created.

5. Repeat steps 2 to 4 as necessary.

You can later modify the animation in the animation mixer.

Replacing Shape Keys

You can replace a shape-key source with a new shape. All new clips using that source will be based on the new shape, and any existing clips that use the source are also updated.

1. Deform and transform the cluster into a new shape.
2. In the explorer, make sure that Properties are displayed.
3. Expand Mixer/Sources/Shapes and select the shape you want to replace with the new one.
4. From the Animate toolbar, choose **Deform** > **Shape** > **Replace Shape Key**.

Plotting Shapes

Plotting shape animation plays back the scene, storing shape keys along the way. You can plot shapes to “bake” together shape animation on different clusters and apply it to the whole object.

Shapes are always plotted in absolute mode. For more information about shape reference modes, see page 301.

1. Select a shape-key mode. The shape-key modes determine how the shape clips are instantiated when they are automatically added to the animation mixer—see *Understanding Shape-Key Modes* on page 298.

2. Select what you want to plot:

   - Select a cluster to plot its shape.

   - Select (tag) points to create a new cluster and plot its shape.

   - Select an object to create a cluster containing all its points and plot its shape.

3. Choose **Tools** > **Plot** > **Shape** from the Animate toolbar. The Plot Shape dialog box opens.

4. Set the options:

   - **Start Frame** and **End Frame** specify the range of frames to plot.

   - **Step Value** specifies the frame step increment. If this value is 1, a shape key is saved at every frame. If this value is 2, a shape key is saved at every other frame, and so on.

   - **Apply Keys** specifies whether to apply the shape keys in addition to storing them. If this option is on, the animation plays back twice, applying keys on the second pass.

   - **Shape Name** specifies a prefix for the shape source names.

5. Click OK. The animation plays back, storing shape keys.

Mixing and Sequencing Shapes in the Animation Mixer

Once you have created shape keys, you can use the animation mixer to sequence and mix them.

In the animation mixer, shape keys are applied as clips on shape tracks in exactly the same way that actions are applied as clips on animation tracks. You can add shape tracks, create clips from shape sources, move clips, copy clips, transition between clips, and add annotations. For more information about working in the animation mixer, see *Chapter 10: The Animation Mixer* on page 231.

While it may seem odd to modify the time control or apply time warps to shape clips (which are essentially static), these features become useful when you create compound shape tracks. You can create a compound shape track that includes transitions between shape keys, then modify the time control or add a time warp to the entire compound.

Adding Shape Clips

To manually add a shape clip in the animation mixer:

1. Make sure that the correct model is open in the animation mixer.

   If you have not already done so, open the animation mixer in a viewport or floating window. Select the model and click **Update** on the mixer's command bar.

2. If you want to add the clip to a new track, first create a new track by right-clicking on any existing track and choosing **Add Track** > **Shape**. Clips cannot overlap on a track, so you can add clips to the same track to play them in sequence, but to mix (overlap) clips you must add them to separate tracks.

3. Do one of the following:

   - Right-click over the location on the track where you want the clip to start and choose one of the model's stored shape from the **Load Source** menu.

   - Drag a shape from the model's Mixer\Sources\Shapes folder in the explorer onto a location on the track; you cannot drag shape sources from another model.

   A clip is created, representing an instance of the shape source.

Blending Clips in the Mixer

Once you have added shape clips to the animation mixer, you can use any of the mixer's features to move, copy, extrapolate, scale, trim, and blend them. For example:

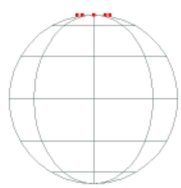
- You can create sequence of shapes by creating clips one after another using transitions to help smooth them.
- You can composite shapes by adding clips for different clusters on the same frames of different tracks. One clip could drive the brow of a character while a different clip drives the mouth.
- You can mix shapes that drive the same points or cluster by adjusting their clip weights and setting keys or other animation.

For more information about using the mixer in general, see *Chapter 10: The Animation Mixer* on page 231.

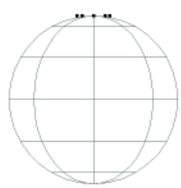
Shapes in the Mixer: A Quick Walk-Through

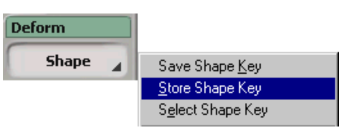
This example provides a quick introduction to using the animation mixer with shapes.

1. Get a primitive surface sphere (**Get** > **Primitive** > **Surface** > **Sphere**) and tag the top points (**t** supra key).

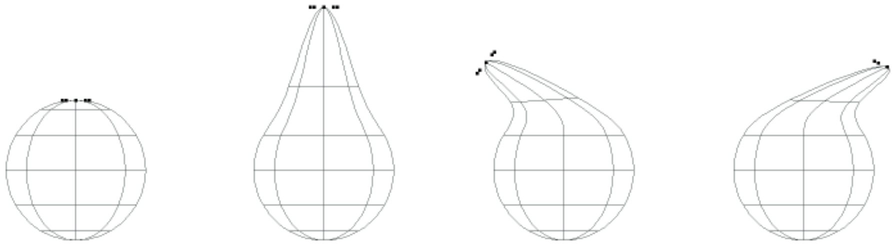


2. Create a cluster by clicking the **Cluster** button on the Edit panel.

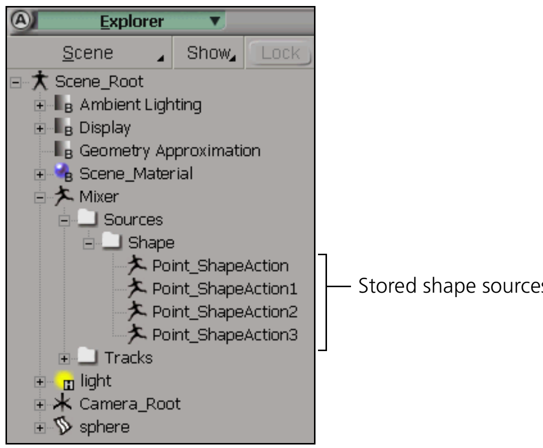




3. With the cluster in its original shape, choose **Deform** > **Shape** > **Store Shape Key**. At the same frame, translate and rotate the cluster into three more shapes, and store another shape key for each.

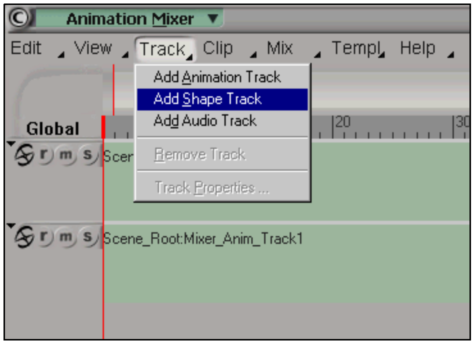


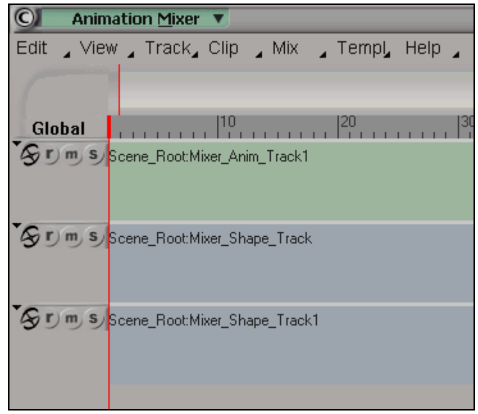
4. Scrub the timeline back and forth. Unlike when saving shape keys, there is no animation when storing shape keys. Now open a scene explorer, make sure that Show > Mixers is on, and expand the Mixer\Sources\Shapes branch—there should be four shape sources.



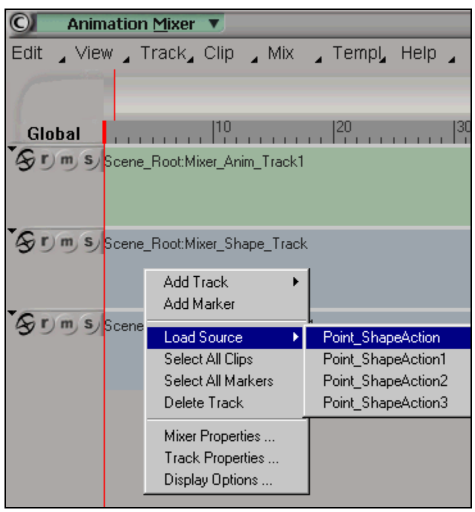
5. Change viewport C to the animation mixer view and click **Update** in the mixer's command bar.

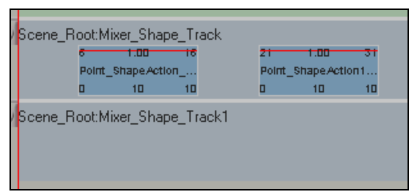
6. Choose **Track** > **Add Shape Track** to add a shape track to the mixer, then choose it again to add a second track.



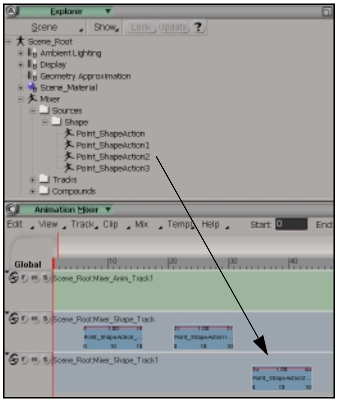


7. Right-click on a blue shape track and choose **Load Source** > **Point_ShapeAction**. A ten-frame clip of that shape is created on the track. Right-click again to the left of the first clip and choose **Load Source** > **Point_ShapeAction1**.

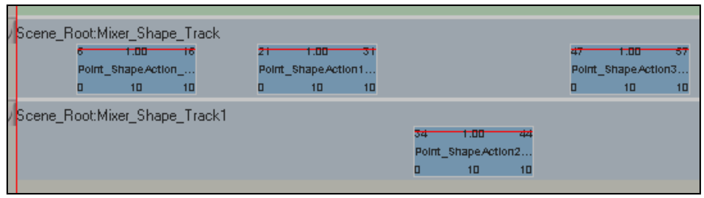




8. Another way to add clips is to drag sources from the explorer. Click on Point_ShapeAction2 in the Mixer\Sources\Shape branch, then drag it onto a track. Repeat for Point_ShapeAction3.



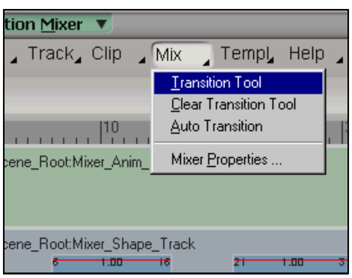
Drag clips onto a track.

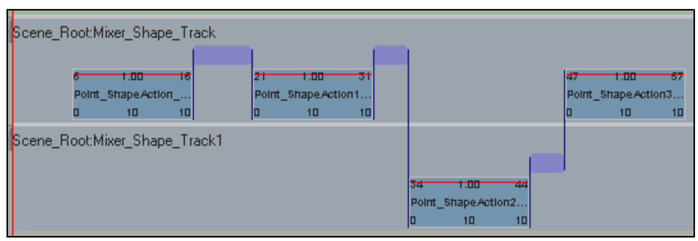


The result could look like this.

9. Play the animation back. Notice how the shapes snap at the beginning of each clip.

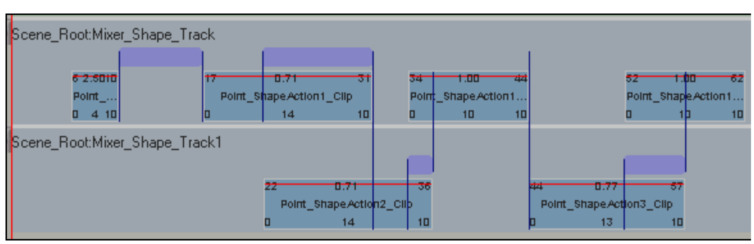
10. The next step is to blend the shapes with transitions. Choose **Mix > Transition Tool**, then left-click on each clip starting from left to right. Transitions are represented by purple bars linking clips. Play back the animation again.





Transitions between clips

11. You can now proceed to load more clips, move and scale the clips, and modify the animation.



Complex shape animation in the mixer

Mixing Shape Animation on Multiple Clusters

Once a cluster has shape animation, you can control its weight—that is, how strongly the different shapes "pull" the components in the cluster. You can modulate the weight so that the components are pulled only half as strongly by the target shape, or even produce exaggerated shapes that overshoot the target.

In addition, if clusters overlap—that is, the same components belong to two or more clusters—you can control how strongly deformations and transformations applied to the different clusters affect the shared components.

To set clusters' weights for shape animation

1. Create an object with shape animation. You can save or apply shape keys or load shape sources in the animation mixer. Once there is at least one instantiated clip, a Cluster Shape Combiner property is created.

2. With the object selected, click **Property** on the Selection panel. Expand the object's operator stack and display the Cluster Shape Combiner property editor.

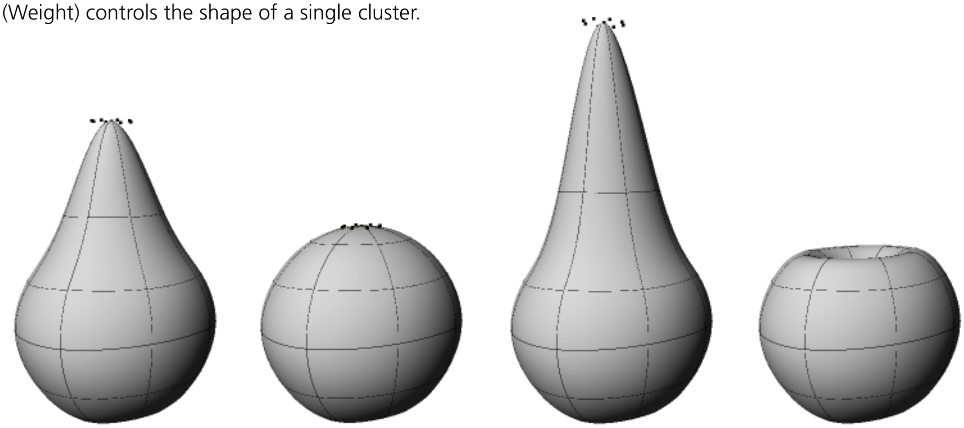
3. Adjust values as desired. The Cluster Shape Combiner property editor contains Weight and Overlap Weight parameters for each animated cluster.

   - **Normalize Cluster Weights** controls how the Weight values are combined. When off, shapes are combined additively. When this value is on, the weights are normalized, resulting in a shape that is in between the sources of the clips that affect the object at the current frame. When this value is off, clusters are mixed in additive mode; it is possible to exaggerate shapes by setting Weight values higher than 1. This option has no effect on Overlap Weights.

   - **Wt** controls how strongly the corresponding cluster is pulled by its shape sources.

   - **OvrlpWt** controls how strongly the corresponding cluster affects components shared with other animated clusters. These values are always normalized.

Wt (Weight) controls the shape of a single cluster.



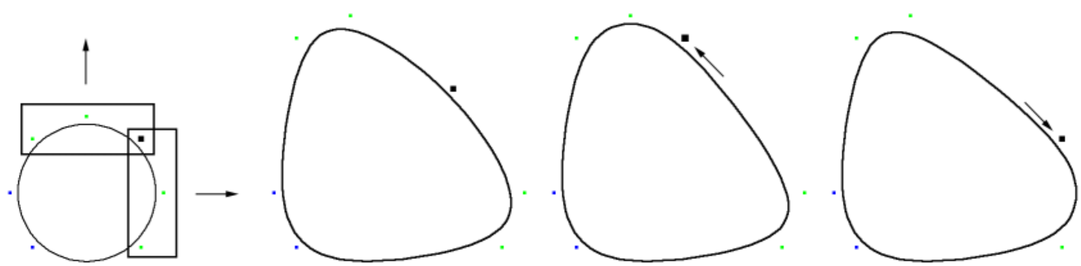
Weight = 1
Actual shape key

Weight = 0
Reference

Weight = 2
Overshoots shape key

Weight = –0.5
Inverts cluster shape

OvrlpWt (Overlap Weight) controls how strongly clusters pull on shared components.



Point is shared by Top and Right clusters.

Top_OvrlpWt = Right_OvrlpWt

Top_OvrlpWt > Right_OvrlpWt

Top_OvrlpWt < Right_OvrlpWt

Painting Shapes

You can paint shapes using the weight maps and the mixer. This lets you use the brush tool to apply a shape on top of another shape or the reference.

1. Store shape keys for an object as described on page 305.
2. Open the animation mixer and create a shape track. Load a shape key as a clip.
3. With the object selected, click the **Cluster** button on the Selection panel and choose the cluster corresponding to the shape key.
4. Add a weight map to the cluster by choosing **Get** > **Property** >**Weight Map**. The weight map property editor opens. For more information about weight maps and painting in general, see *Weight Maps* in Chapter 6 of the *Modeling & Deformations* guide.
5. Open an explorer view. Expand the object’s operator stack, then expand its Clusters folder and expand the cluster corresponding to the shape key. Select the shape key.
6. Choose **Deform** > **Shape** > **Connect Weight Map** from the Animate toolbar.
7. Pick the weight map in the explorer view. It should be listed under the cluster corresponding to the shape key.
8. Select the weight map and press the **w** key to activate the paint tool. You can now paint weights in the geometry views:
   - If the shape clip overlaps other clips on other tracks, it will be mixed against them according to the weights you paint.
   - If there are no overlapping clips, the shape clip will be mixed against the cluster’s shape reference.

Deleting Shapes

You can delete both shape clips and shape keys:

- Deleting a shape clip removes that instance from the animation mixer. This is useful if you saved or applied a shape key at the wrong frame.
- Deleting a shape source removes it as an available shape. It also removes all clips based on that source.

Deleting Shape Clips

To delete a shape clip, you must use the animation mixer:

1. Make sure that the correct model is open in the animation mixer.

   If you have not already done so, open the animation mixer in a viewport or floating window. Select the model and click **Update** on the mixer's command bar.

2. If you used the commands on the Animate toolbar to save, select, or apply a shape key, the clip is in a compound corresponding to the cluster. Expand the compound by double-clicking on it.
3. Right-click over a clip, and choose **Delete**. The shape clip is removed.

Deleting Shape Sources

To delete a shape source, you must use the explorer:

1. In the explorer, make sure that **Show** > **Mixers** is on.
2. Expand the model's Mixer\Sources\Shapes folder.
3. Right-click on a shape source and choose **Delete**. The shape source and all its clips are removed.

Chapter 13 Audio

You can add audio files to your scenes. This lets you adjust the timing of your animations by using the sound as a reference. For example, you can use an audio file as reference for lip synching.

Sound files are added as clips in the Animation Mixer. For more information about using the Mixer in general, see *Chapter 11: Actions* on page 253.

You cannot copy, scale, trim, extrapolate, warp, or mix audio clips.

Supported Audio File Formats

The following audio file formats are supported:

- ***.aiff:** Audio Interchange File Format
- ***.aiffc:** Audio Interchange File Format - Compressed
- ***.mov:** Apple QuickTime
- ***.qt:** Apple QuickTime
- ***.wav:** Windows waveform

You can only use sound files that have the same sampling rate. If you have already added a sound file sampled at one rate, you cannot add files at different rates.

Adding Sound Files

To add a sound file to a scene:

1. Select the Scene_Root or another model.
2. In the Animation Mixer, choose **View** > **Update From Selected** from the command bar.

   You can only add audio tracks to the main model level. You cannot add audio tracks inside compound actions.
3. Add an audio track by choosing **Tracks** > **Add Audio Track** on the command bar.
4. Right-click on the audio track and choose **Load Source**. A browser opens.
5. Use the browser to select a sound file and click OK.

Once you have added a sound file, you can move it along the track, add markers, mute, and solo it. For more information, see *Chapter 10: The Animation Mixer* on page 231.

- There is no undo function for dragging an audio clip in the mixer.
- To change the audio source, you should delete the clip and instantiate a new one.

Playing Sound Files

To play audio when you play back the scene or scrub the timeline, you must first enable audio in your preferences. You should also set the frame rate for playback.

Enabling Audio

To enable audio in your preferences:

1. Choose **File** > **User Preferences** from the main-menu bar. The User Preferences dialog box opens.
2. On the General tab, turn on **Play Audio When Scrubbing** in the Scrubbing box.

Setting the Frame Rate

The frame rate determines how smoothly the audio plays back. You can set it from the **Playback** menu or use the button next to the current frame box on the Playback panel.

- **All** plays all frames in sequence. The sound playback is choppy as each frame is calculated before being displayed.
- **RT** (**Real Time**) gives priority to playing the sound smoothly. If necessary, playback speed is maintained by skipping frames.

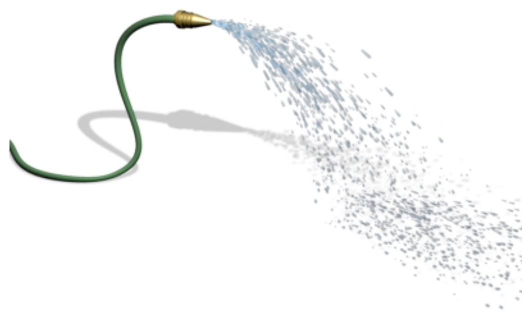
Muting Audio

You can mute audio with the Play/Mute Audio button on the Playback panel. You can also mute a specific audio track using the Mute icon in teh animation mixer.

Section IV • Simulation

Chapter 14 Particles

SOFTIMAGE|XSI's animation controls include a pixel-based particle-generation system that allows you to simulate types of natural phenomena containing many particles. This particle feature makes it easy to animate particle emissions such as sparks rising from a campfire, fireworks, or the tail of a comet trailing off into space.



The illustration above shows how a stream of particles, linked to a model of a garden hose, has been created to look like spraying water.

Particle Workflow

The sections below present a suggested workflow of the tasks involved in creating a particle simulation.

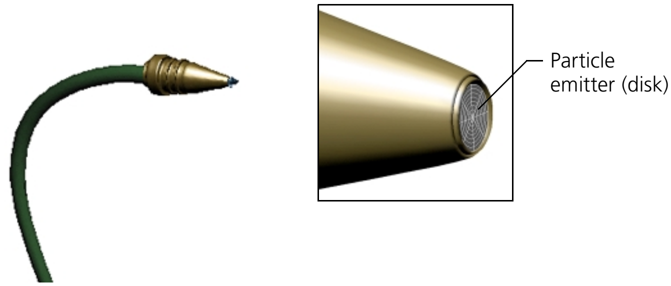
Step 1: Define the particle simulation

In this first step, you create an element in the scene known as a *particle cloud*. This is an operator whose property editor lets you define the general operating parameters of the particle simulation, such as its duration in frames and frequency in which the particles' position is calculated. For more information, see *Setting Particle Operating Parameters* on page 333.

Step 2: Choose the emitter object and define its emission characteristics

Once you have determined the global characteristics of the particle simulation, you choose the elements in your scene that will act as the *emitter object*. This is the object from which the particles will emerge. As you do so, you will define how the particle emission will take place. Here, you define such things as the rate (density), speed, and spread angle of the emitted particles.

In this example, we have chosen the disk at the mouth of the hose nozzle to act as our emitter.



For more information on defining emitter objects, refer to *Particle Emitters* on page 334.

Step 3: Define the particle type

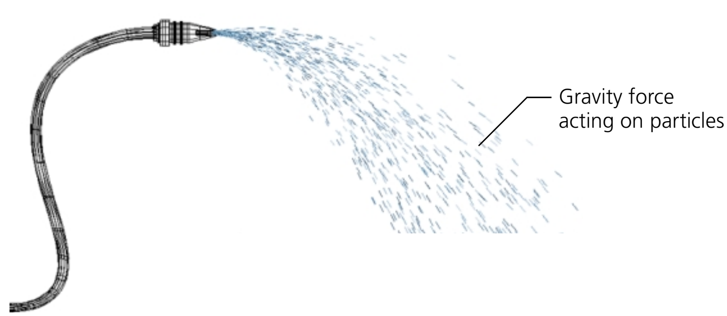
Now, you will define the particles themselves as they are ejected from the emitter object. In the Emission property editor's **ParType** drop-down list, select from a list of particle emission types (called particle types). Initially, you have one default particle type to work with, called Particle_Type. Click the **Edit** button to open the particle type's property page.

From this editor you can modify the selected particle type's physical characteristics, such as their mass, size, and lifespan before they disappear (refer to *Creating and Editing Particle Types* on page 337). You can create an unlimited number of particle types and apply them to any emitter object.

Step 4: Add natural forces to particle types

In many cases, you will want to define how particles react to environmental phenomena such as gravity, vortex effects, wind, and electrostatic force.

In the example below, gravity has been added to particles to simulate the effect of water falling.

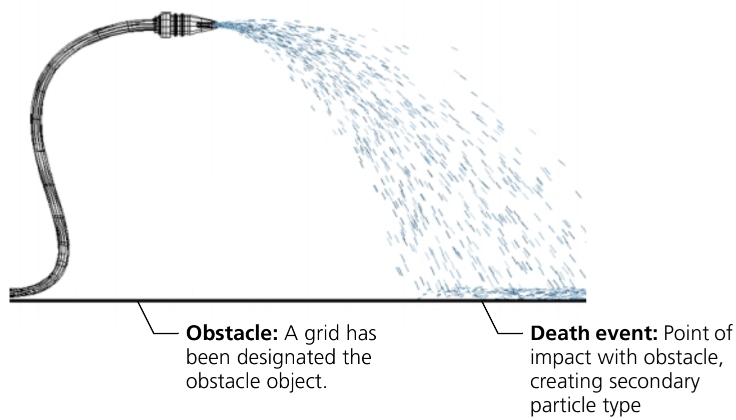


For information on how to add forces to particles, refer to *Applying Natural Forces to Particles* on page 343.

Step 5: Add death events to particle types

To add realism to your particle simulation, you also may want to have your particles transform into other types of particles at a certain time, such as at the moment of impact with an obstacle. This moment of transformation is known as the *death event*.

The example below shows how a death event has been specified to occur at the point of impact with an obstacle, thus generating a secondary particle type simulating water droplets.



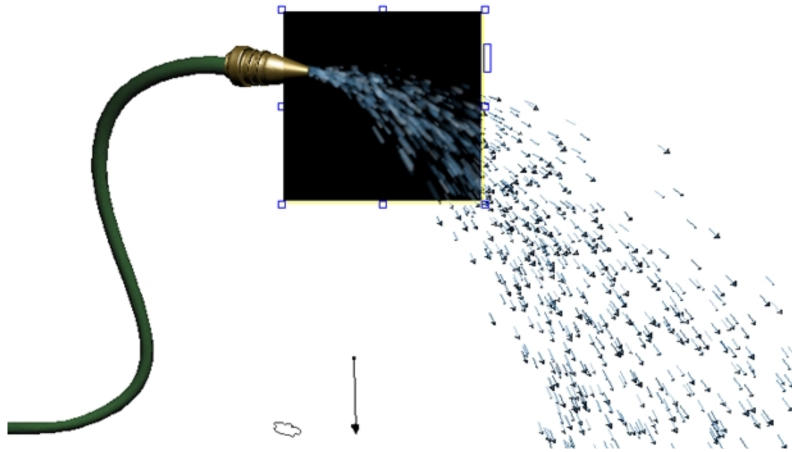
For information on defining death events, refer to *Defining the Particle Death Event* on page 341.

Obstacles are discussed later, in *Creating Obstacles* on page 349.

Step 6: Apply rendering effects to emitted particles

You can add rendered properties to your particles, such as color and shadow, as well as apply one of three types of particle shaders. You can then view them as you would any other animation by drawing a render region and caching the images for playback.

For information on applying rendered properties to particles, refer to *Rendering Particles* on page 347.



Setting Particle Operating Parameters

To create particle animation, you have to start out by creating a particle cloud in the scene. By default, each time you create a particle cloud a property editor displays, prompting you to define the global characteristics of the particle animation.

To create a particle cloud

1. In the animate toolbar, choose **Simulate** > **System** > **Create Particle Cloud.**

   A cloud icon (shown to the left) appears in the viewports and the SparksOp property editor appears.

2. Define the particle cloud's general operating parameters

   Here, you can define the particle simulation's overall characteristics, including its length, the frequency at which the position of emitted particle is calculated, and its file name (which can subsequently be saved as a preset).

   The table below summarizes the options you can set in the particle cloud's property page:

Option	Description
Execution state	Sets the visibility of the particle animation as follows: **Interactive:** Particle animation updates each time you modify a particle parameter. **Not Interactive:** There is no update of particle animation as long as you remain in the current frame. Update only occurs during playback or when you move to a different frame. **Disconnected:** The particle simulation is not calculated. You can set parameters and move back and forth on the timeline without having to wait for the particle simulation to update.
Start frame	The first frame at which the particle animation will begin.
Duration	The duration, in frames, of the particle animation.
Particle percentage	The amount of particles, in percent, to be emitted.
Output sequence	The file name of the particle animation.
Clean cached files	When selected, all the **.ptp** files that were recorded and cached for playback are erased each time you exit SOFTIMAGE|XSI. **.ptp** files record the position of your particles at each frame in the animated scene.
Number of iterations	The number of times a particle's position is calculated per interval.
Number of interframes	The number of times a particle's position is calculated per frame.

Particle Emitters

Once you created and modified a particle cloud to define the general behavior of your particle simulation (see previous section), you must choose an object in your scene that will emit the particles. You then set the dynamic properties of particle emission by assigning the emitter object attributes like particle spread angle, speed, and density.

To choose and define an emitter object

1. Select a particle cloud.
2. In the Animate toolbar choose **Simulate** > **System** > **Set Emission**.
3. Pick one or more objects in the scene that will act as the source, or *emitter*, of the particles, then right-click to end the picking session.

If you create a particle cloud with an object already selected, that object automatically becomes the emitter.

Most objects can be emitters, while clusters cannot.

4. In the Emission property editor that displays, define the properties of the particle emission.

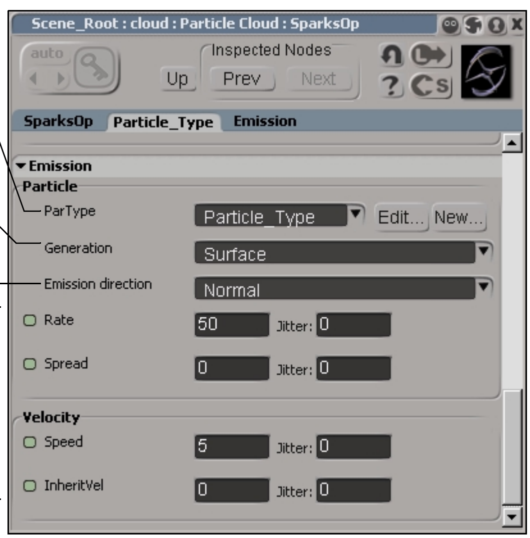
   Emission controls are summarized in the illustration below. For a detailed description of each option, refer to Online Help.

ParType controls: Selects and edits particle attributes. See *Creating and Editing Particle Types* on page 297.

Generation: Sets the point of origin of emitted particles

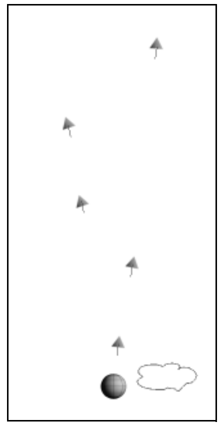
Emission Direction: Sets the direction of emitted particles.

Controls the density, speed, spread angle, and velocity of emitted particles.

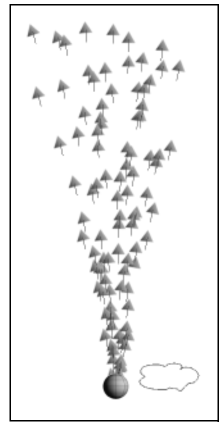


The diagrams below illustrate the effects of rate, speed, and spread angle on emitted particles.

Particle rate

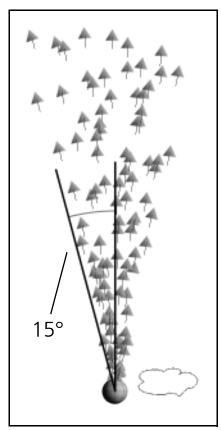


Particle rate at 5 units per second



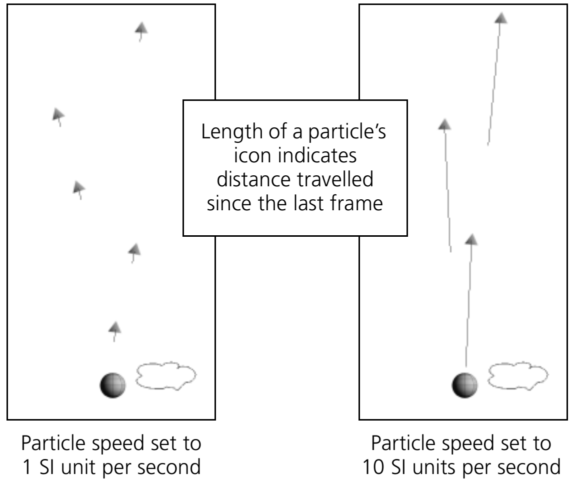
Particle rate at 100 units per second

Particle spread



Particle spread set to 15 degrees.
The net result is a total speed of 30 degrees

Particle speed



Creating and Editing Particle Types

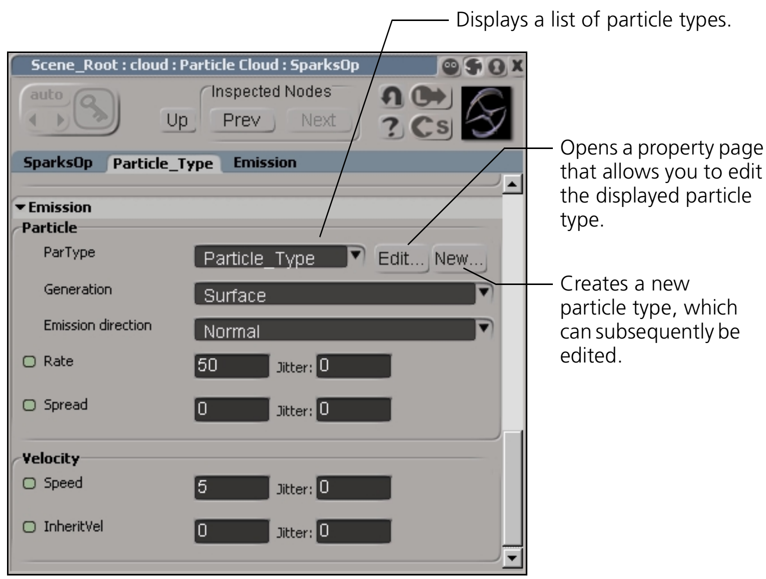
As you define the emission properties of particles in the Emission property editor (see previous section), you also have the ability of editing the characteristics of particles once they have been expelled from the emitter object. The characteristics of these particles are collectively referred to as a *particle type*. You can create an unlimited number of particle types and apply them to any emitter object.

Particle types can possess a wide range of attributes that include:

- **General characteristics** such as mass, size, and lifespan of the particles (see *Setting General Particle Characteristics* on page 338)
- **Death event information** (see *Defining the Particle Death Event* on page 341)
- **Natural force properties** (see *Applying Natural Forces to Particles* on page 343)
- **Rendering properties** that include color, shadow and shader attributes (*Applying Render Properties to Particle Types* on page 347)

The default particle type used in particle emission is called **Particle_Type** and appears at the top of the Emission property editor in the **ParType** text box. You can modify the default particle type and save it under a new name.

The emitter parameters you set in the Emission property editor apply to ALL particle types.



Creating a New Particle Type

1. Open the emitter object's Emission property editor.

   To quickly access this editor, expand the emitter object's node in the explorer and click the Emission icon.

2. Click the **New** button to create the new particle type.
3. You can now click the **Edit** button to assign properties to the new particle type.

Setting General Particle Characteristics

The fundamental characteristics of each particle type are set in their respective property editors, which can be accessed from the emitter object's Emission property editor or directly from the particle type's own property node in the explorer.

Editing a Particle Type

1. Open the emitter object's Emission property editor.
2. Choose the particle type you wish to edit from the **ParType** combo box.
3. Click the **Edit** button to display the selected particle type's property editor.

You can bypass steps 1 to 3 by clicking on a particle type's node in the explorer.

4. Click the **General** tab and modify the particle type properties as summarized in the following diagram. For more information on individual controls, refer to Online Help.

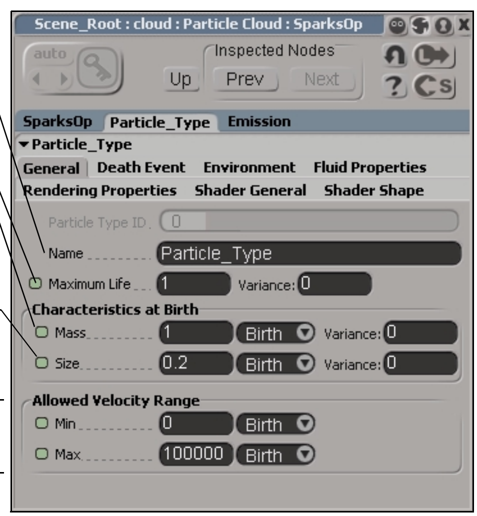
Name: Displays and lets you modify the particle type name.

Maximum Life: Controls the maximum amount of time, in seconds, that the particle exists once it is emitted.

Mass: Specifies the mass of particles. Particles with greater mass will require more force to displace compared to particles with lesser mass.

Size: Controls the size of the particle, in SOFTIMAGE units. The farther away a particle appears in the viewing area, the smaller it appears when it is rendered.

Allowed Velocity Range: Sets the range between the maximum and minimum particle speed. This is a handy way to avoid extremely high or low speeds when particles are affected by magnetic, electric or other natural forces.



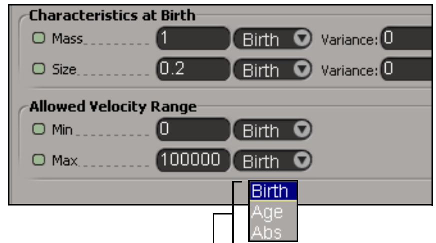
Variance controls: Specifies the degree of variance in the lifetime, mass, and size of emitted particles.

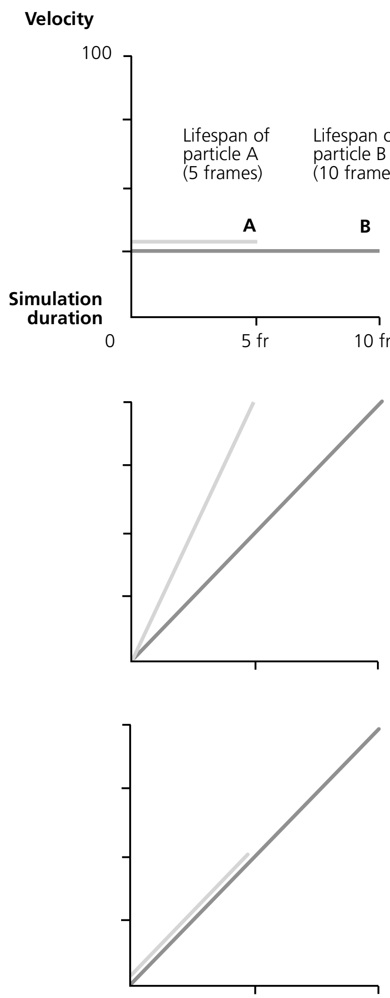
Animating Particle Parameters

You can add realism to your particle simulation by animating certain particle values such as speed, color, and the influence of environmental forces. Particle values can change as a function of their own lifetime or as a function of the lifetime of the particle simulation itself. This lets you create effects such as making particles change color as they age or having all particles suddenly accelerate at a given point of time.

Particle parameters that are animatable have a drop-down menu that lets you specify how they can be animated.

The example below shows the different ways in which particle velocity can be animated:





Birth: Parameter value remains constant throughout the lifetime of the particle (no parameter animation).

Age: Parameter value changes over the lifetime of the particle. Use a function curve in the animation editor to set this parameter change.

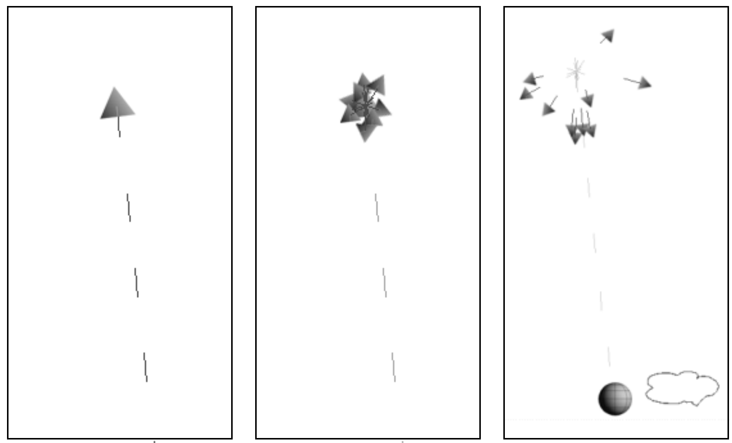
Age%: Parameter value defined in the first 100 frames of a function curve stretched or compressed to match the lifetime of the particle.

Abs(olute): Parameter values of all particles change in relation to the length of the simulation. These changes are also set in the animation editor.

Defining the Particle Death Event

At the end of its lifetime, in what is known as the *death event*, a particle can either disappear or decay into one or more secondary particles. In the latter case, the original particle becomes a source point and emits one or more particles from its current location. Often, a death event is associated with particles as they strike an obstacle, like sparks from a welder's torch hitting the ground and disintegrating into a burst of other, smaller sparks before extinguishing.

Particle with a lifetime of 30 frames



Frame 29: Particle nears end of its lifetime.

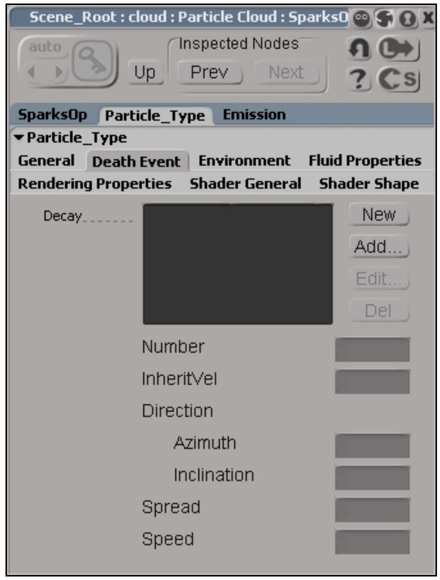
Frame 30: Particle death event occurs.

Frame 35: Particle has decayed into a new particle type.

Death events are specified for each particle type within their respective property editors.

To set particle type's death event

1. Open the particle type's property editor, either from the emitter object's Emission property editor or from the particle type's own property node.
2. Click the **Death Event** tab to display the particle type's Death Event controls.



3. Click the **New** button to create a new particle type that will replace the expired particles, or click the **Add** button to choose from a list of existing particle types.

4. Your selection is displayed in a particle list window. Clicking on the particle type's name in the Decay window makes it the particle type to appear after the death event.

   - Click the **Del** button to remove unwanted particle types from the window.

   - Click the **Edit** button to modify the selected particle type.

5. Specify the behavior of the death event in the lower portion of the property page:

   Number determines how many particles are generated from the dying particle.

   InheritVel determines at what percentage the particles have the original particle's velocity to their own. A value of 100, for example, will make the secondary particles the same speed as the original particles.

   Direction determines the degree of the inclination and azimuth vector along which secondary particles are generated.

   Spread controls the spread angle of secondary particles.

   Speed determines the speed of secondary particles in units per second.

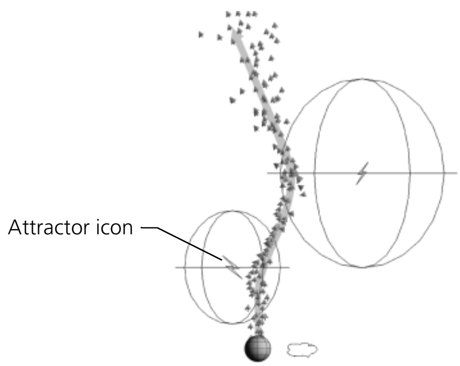
Applying Natural Forces to Particles

You can add further realism to your particles by applying simulations of natural forces to their motion. Each particle type you create can have one or more sets of natural forces applied to it.

The kinds of forces you can apply to particles fall under four categories:

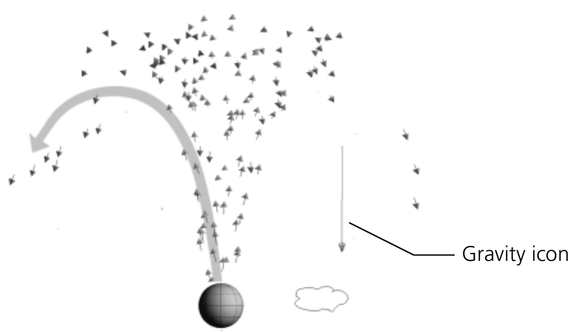
Attractor

Attracts/repells particles, much like a magnetic attracts /repells iron filings. Positive values repell while negative values attract.



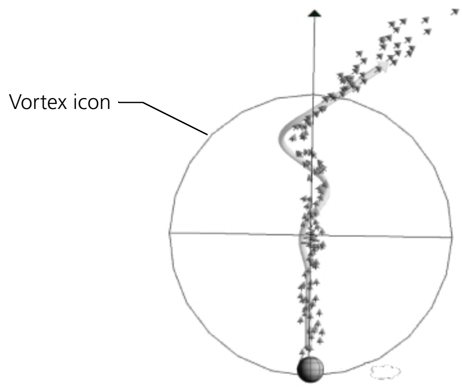
Gravity

Simulates the effect of a gravitational force on particles.



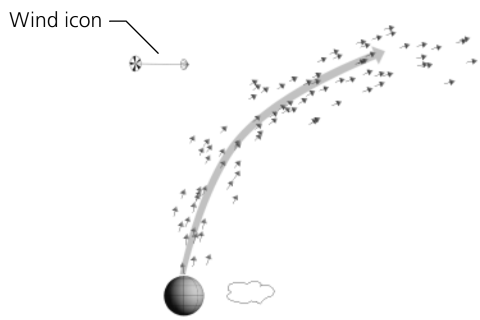
Vortex

This force tends to generate a spiraling effect on emitted particles.



Wind

Controls the effect of wind on particles.



Each force is represented by its own icon and can, through its property editors and interface commands, be displayed, selected, transformed, rotated and scaled like any other object in the scene.

Creating a Natural Force Effect

Each of SOFTIMAGE|XSI's four types of forces are created by choosing **Simulate** > **Environment** > (**force**) from the Animate toolbar.

Once a force object has been created and defined, it is applied to the particle source (emitter object). It can then be separately adjusted in the property page to act upon each of the emitter object's particle types differently.

To apply attractor, gravity, vortex, or wind force effects to a particle type

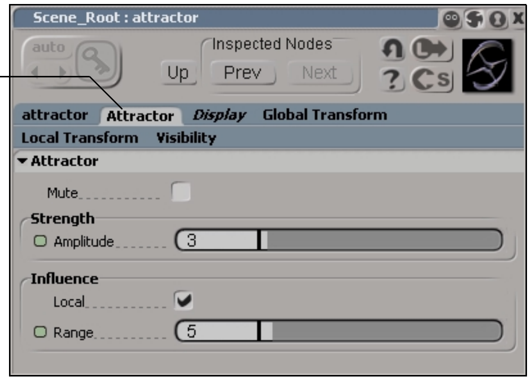
1. In the Animate toolbar, choose **Simulate** > **Environment** > **Create** (**force**).

   The force object appears in the viewports and its property editor displays.

Choose the Attractor tab and modify the Strength and Influence parameters.

Strength (amplitude) specifies the degree to which the natural force acts upon the particles.

Influence (range) specifies the area in which the natural force acts upon the particles.



   As you modify the force object's attributes, its icon updates accordingly to provide you with a visual cue as to its strength and/or range.

2. Select the particle cloud to which you want to apply the force.
3. In the Animate toolbar, choose **Simulation** > **Environment** > **Apply Force**.
4. Left-click to pick the force object or objects then right-click to terminate your selection.
5. Select the emitter object and click the **Property** button in the Selection panel.
6. In the explorer, choose the particle type node to which you want to apply the force.
7. Click the **Environment** tab and modify the properties on this page that correspond to the force acting on the particle type.

If, for example, you have chosen a vortex force to act on your particles, you must have a value of greater than zero in the Environment's **Vortex** control for any vortex effect to occur.

Forces: Specifies the degree to which the natural force acts upon the particle type.

Rendering Particles

The particle system does not create a stream of individual geometrical objects; instead, it creates a uniform series of small shapes that represent points in 3D space. The rendering process applies a spot of color at the location of the particle, much like a brush stroke. This process permits many particles to be in a scene without making the scene too large.

Applying Render Properties to Particle Types

You can assign rendering properties such as color and color variance to your emitted particle types. The Rendering page in the Particle Type property editor lets you assign these properties to particles in your scene.

To set render properties to particles

1. From a particle emitter's property editor, choose a particle type from the drop-down list and click the **Edit** button.

   The particle-type property editor displays.

2. Choose the **Rendering Properties** tab to display the particle type's Rendering property page.

3. Specify the rendering properties of the particle type by inputting your selections using the controls described below:

   RGB Values lets you apply color to the particles.

   Variance (**HLS**) lets you specify the variations to the particles' HLS color channel values.

   Alpha Variance lets you specify the variations to the particles' Alpha channel value.

Particle Shaders

To create a rendered image of your particles, you apply a shader to them, just as you would to render any other object in your scene.

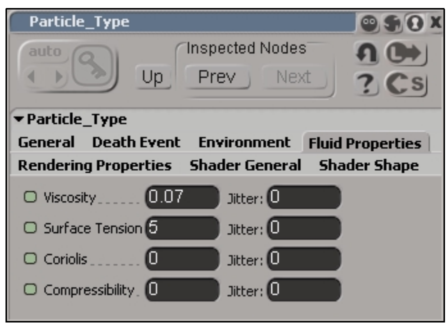
You apply rendered effects to emitted particles by choosing one of two predefined shaders then customizing the shader properties to achieve the effect you want.

- The **Sparks** shader makes your particles look like... well...sparks. You can define the shape, color, and illumination of your sparks as well as apply effects like motion blur.
- The **Blob** shader turns the particles into metaball-like objects that flow like the semi-liquid wax balls you see in lava lamps. Besides setting the shape, color, and illumination, you can specify whether or not the particles flow into one another.
- The **Hyperflow** shader renders particles as a liquid that exhibits hydrodynamic qualities. You can use this shader to create effects like frothing milk or flowing water. (See *Applying Fluid Properties to Particle Simulation (Hyperflow)* on page 352.)

Shaders are always defined within the particle cloud. The shader properties you set here are then passed on to the emitter object.

To apply shaders and rendered effects to particles

1. Select the particle cloud against which the shader properties will be applied.
2. Choose **Simulate** > **Shaders** > (**shader**) to apply the shader to the particle cloud.
3. Draw a render region over the emitter object to view to results of the shader on the particles it emits.
4. Choose **Simulate** > **Shaders** > **Edit** to edit the general properties of the shader. Refer to on-line help for full details on shader parameters.
5. In addition to the general shader characteristics you applied in the previous step, you can also specify shader properties and rendered effects to individual particle types. To do so, open the particle type property editor and click on the property page tab that corresponds to the shader.



Choose a property page whose controls correspond to the type of shader that has been applied to the particle type.

Choose **Rendering Properties** to apply color to the particles in your particle type.

Choose **Shader General** to specify decay behavior as well as set shadows and motion blur.

Note that some property pages apply only to certain types of shaders. **Fluid Properties**, for example, can only be used on Hyperflow shaders, whereas **Shader Shape** can only be used with Sparks shaders.

Refer to on-line help for more information on each property page.

Creating Obstacles

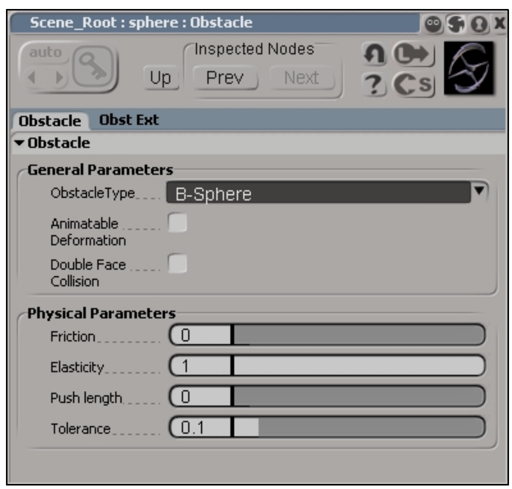
You can define objects in your scene to act as obstacles that block the path of emitted particles. Depending on the effect you wish to create, you can specify the particles to bounce off, stick to, or disappear upon striking the obstacle. You can also make the particles decay upon impact, creating a death event that generates a new set of particles, like raindrops splitting into smaller drops as they strike a windshield.

Any number of objects in a scene can be set up as obstacles to emitted particles, including NURBS and non-NURBS type objects.

To set an object as an obstacle

1. Select the particle cloud object and from the animate toolbar choose **Simulate** > **Environment** > **Set Obstacle**.
2. Pick an object in the scene that will act as an obstacle in the path of emitted particles. You can pick more than one object. Right click to end the picking session.
3. Use the controls in the Obstacle property editor to control obstacle behavior.

   For more information on obstacle parameters, refer to on-line help.



To set particle behavior upon impact

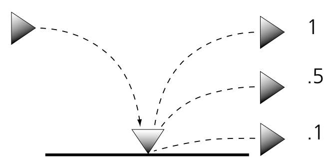
1. Open the obstacle object's Obstacle property editor and choose the Obst Ext tab.

   In the **Particles** pane, a list of all particle types defined for emitter objects in the scene is displayed.

2. Choose the particle type whose behavior you want to affect. At this point you can edit existing particle types if needed.

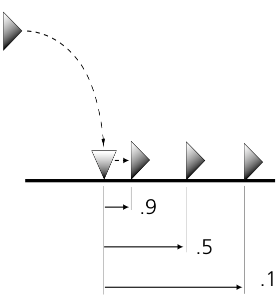
3. In the **Mode** drop-down list, choose the behavior of the particle type as its particles strikes the obstacle:

   Bounce: causes the particles to bounce off the object. The dynamics of particle bounce is set in the Obstacle property editor. Particles can still decay into other particles at their death event, if specified.



Setting bounce elasticity:

If the obstacle has been set to Bounce mode, values specified in the Elasticity controls of the Obstacle property editor will have the effect shown to the left.



Setting friction values:

To simulate particle friction, the obstacle must be set to Bounce mode, and a value of 0 (no bounce) entered in the Elasticity controls.

Values set in the Obstacle property editor' Friction controls will have the effect shown to the left. (The lower the surface friction value, the farther the particles will slide.)

   Stick: causes particles to stick to the object and remain there for the duration of their lifetimes.

   Disappear: makes particles disappear upon impact with the object.

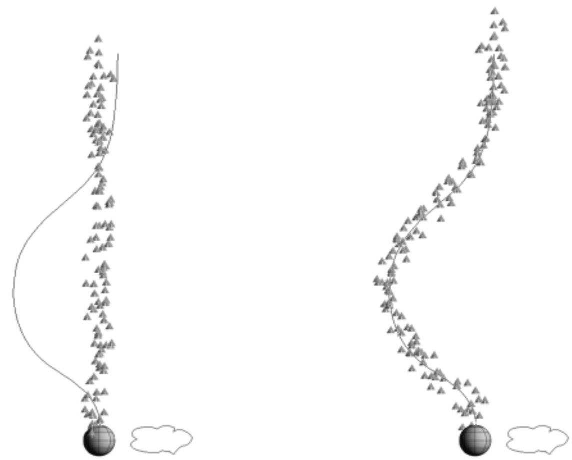
   Decay: causes the particles to bounce off the object. Particles can still decay into other particles at their death event, if specified.

   Emit: upon impact, particles decay into a new set of particles specified in the obstacle object's Emission controls. If some particles do not strike the object, and a death event has been specified for their own particle type, they will decay independently of the obstacle's emission settings.

4. If you chose **Emit** as the particle mode in the previous step, use the **Emission** controls to select and edit the particle type created at the particles' point of impact.

Applying Deformations to Particles

You can apply deform operations (e.g. push, taper, ...) to particle systems.



Particle system before deform **Particle system after deform**

For information on how to apply deformations to particle systems, refer to *Chapter 6: Introduction to Deformations* in the *Modeling & Deformations* guide.

Applying Fluid Properties to Particle Simulation (Hyperflow)

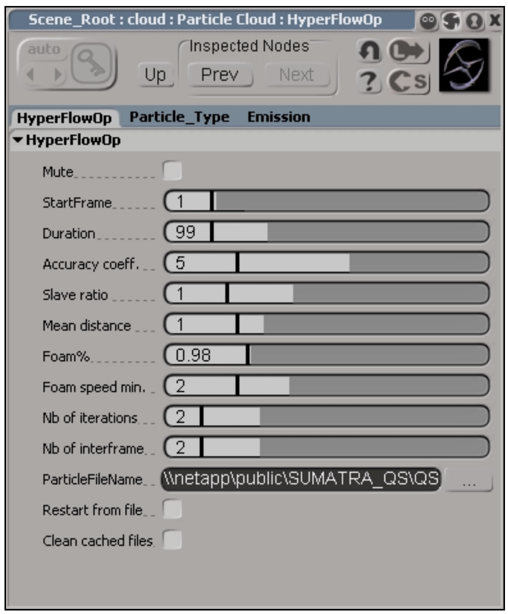
SOFTIMAGE|XSI hyperflow simulation tools lets you apply hydro-dynamic properties to particles. This lets you create liquid-based effects such as water flowing out of a pipe or sand filling up the lower portion of an hourglass.

Hyperflow simulation requires the creating of its own particle cloud and can only be applied to implicit squares, discs and cubes.

To create a hyperflow simulation

1. In the animate toolbar, choose **Simulate** > **System** > **Create Fluid** > **From** (**implicit object**).

   The Hyperflow property editor displays.

2. In this editor specify the dynamic attributes of the hyperflow simulation.

   Refer to on-line help for details on each available option.

3. Choose the **Emission** tab and set the object's emission properties (for more information, refer to *Particle Emitters* on page 334).

4. In the Emission property editor's **Par Type** controls, choose the particle type whose particles you wish to use in the hyperflow simulation and click the **Edit** button. Alternatively, you can create a new particle type by clicking the **New** button.

5. In the particle type's property editor, modify the particle type's parameters.

   Choose the **Fluid Properties** tab to display controls that let you set the particle type's fluid attributes, as follows:

   Viscosity: Sets the level of resistance to flow in a fluid. A high parameter value, such as 1.5, produces behavior similar to that of honey. Lower values, like 0.07 are useful when simulating low-resistance fluids like water.

   Surface Tension: Sets the degree of liquid tension on the particles' surface. Particles that simulate liquid in the shape of a cube, for example, will tend to transform into a spherical shape as they fall to the ground. A high level of surface tension will tend to cause the particles to hold their original cube shape longer.

   Coriolis: Sets the angular speed of a Coriolis force. A Coriolis force is the rotational motion of fluid, such as that of water emptying down a drain. The higher the value in this parameter, the stronger the Coriolis force acts on the liquid simulation.

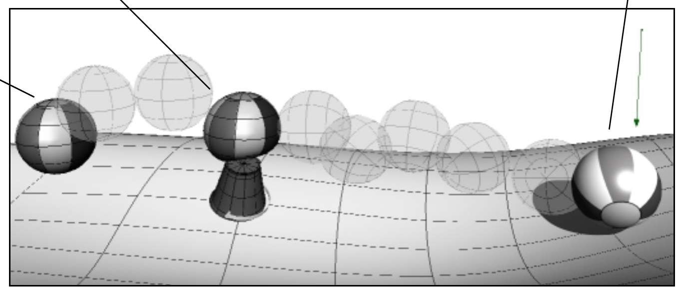
   Compressibility: Sets the degree of compressibility to the simulated fluid. Setting a low value to reduce the amount of bounce behavior in the fluid.

6. Set any other properties you wish to assign the particle type (see *Creating and Editing Particle Types* on page 297).

7. Play the particle simulation and adjust parameters as required.

Chapter 15 Soft Body

SOFTIMAGE|XSI's soft-body feature is a custom effect that simulates the dynamic behavior of non-rigid objects when they are subjected to forces and obstacles. You can, for example, use soft body to affect the morphology of a beach ball being blown across the sand. As the ball is carried along by wind (the force), it can be made to deform each time it bounces on the ground (the obstacle).



Soft body computes the movements and deformations of the object by means of a spring-based lattice whose resolution is defined by the user. The lattice is automatically created inside the volume of the model hierarchy. The higher the resolution, the better the lattice mimics the shape of the object.

You can set the object to bounce back after a collision and either retain its original form or maintain a deformed appearance after the event.

If your object is animated, you can decide whether to preserve its animation or recalculate it according to any external forces you choose to apply. In the former case, soft body only acts on the deformation of the body and not on its movements.

You can have your object interact with stationary obstacles, or animated ones.

If the object you chose for soft-body deformation is part of a hierarchy, all its descendants take part in the effect.

Animation that includes soft body deformations is calculated only once and cached so it can be quickly replayed. If any parameter is changed, the computation is recalculated and cached again.

Creating a Soft Body Deformation: Sample Workflow

The procedure below provides a step-by-step summary on how to create a soft-body deformation on an object that strikes an obstacle as it is driven by an environmental force.

To create a basic deformation using a force and obstacle

1. Select the object to receive the soft-body properties.
2. In the Animate toolbar, choose **Simulate** > **System** > **Apply Soft Body**.
3. In the Soft Body property editor's Soft Body property page, define the object's deformation behavior as follows:

   Sampling—Controls the density of the lattice used in the calculation of the deform. Higher values cause the lattice to more closely resemble the volume of the object to be deformed and thereby create a more realistic result; however, higher sampling rates take longer to process.

   Stiffness—Sets the rigidity of the Soft body lattice applied to the object. High values cause the object to better resist deformations. Note that excessively high settings may result in instability. For this control to have an effect, the **No Stretch** option must be toggled on.

   No Stretch—Determines whether or not the deformed object's length is to be stretched as a result of its collision with an obstacle.

   Body Mass—Defines the weight of the soft-body object. This value has a bearing on how the object interacts with environmental forces and collisions with obstacles.

   Body Friction—Determines how much of a deformed object's energy is absorbed by the obstacle with which it collides.

   Threshold—Defines the plasticity of the object to be deformed. This value sets the point beyond which the effects of the collision can become permanent.

   Damping—Controls the degree to which the deformed object recovers its original shape following a collision, as defined by its threshold parameter (see above). A value of 1 causes the object's deformation to become permanent and 0 simulates a more resilient material that permits the object to return to its original, pre-deformed shape.

4. Choose **Simulate** > **Environment** > **Set Obstacle**.
5. Pick all the obstacles the object will encounter in the simulation. These obstacles can be animated or stationary.
6. Right-click in an empty area to end the picking session and display the Obstacle property editor.

7. Set the obstacle object's properties. Refer to Online Help for more information on the settings available. Pay particular attention to the following:
   - Be sure to correctly set the **Obstacle Type** parameter. If this parameter is set to Actual Shape, for example, the deformed object will mold itself to the actual shape of the obstacle upon collision. This may be appropriate in some cases, but it is computationally expensive. If the soft-body object is striking a flat obstacle, for example, a bounding-box setting would be a better, more economical choice.
   - Set the **Friction** parameter to a high value if you want the soft-body object to stick to the obstacle. A low value causes the object to slide off the obstacle with little resistance.
8. Play back the animation to view the results.

Combining Soft-Body Deformations with Animation

You have the option of taking animated objects and applying a soft-body deformation to them. In this case, the simulation only takes into account the deformation of the body as it strikes an obstacle and not its movements prior to the collision.

However, you can also take an animated soft-body object, extend the animation by applying external forces like wind and gravity, then have the force and deform characteristics calculated as the object hits an obstacle.

To create a basic deformation with an animated object

1. Select the animated object to receive the soft-body properties.
2. In the Animate toolbar, choose **Simulate** > **System** > **Apply Soft Body**.
3. In the soft-body property editor's Simulation property page under **Start Frame**, define the first frame of the deformation simulation. This will be the frame at which point the deformation simulation overrides any animation on the object.
4. Click the Dynamics tab and toggle **Inherit Initial Inertia** on if you want the soft-body object to assume the speed and velocity of the animation at the beginning of the simulation.
5. Choose **Simulate** > **Environment** > **Set Obstacle**.
6. Pick all the obstacles the object will hit during the animation. These obstacles can be animated themselves, or stationary.
7. Middle-click to end the picking session and display the Obstacle property editor.
8. Set the obstacle object's properties as needed.
9. Select the soft-body object and choose **Simulate** > **Environment** > **Create** (force).
10. Play back the animation to view the results.

Index

A

D

E