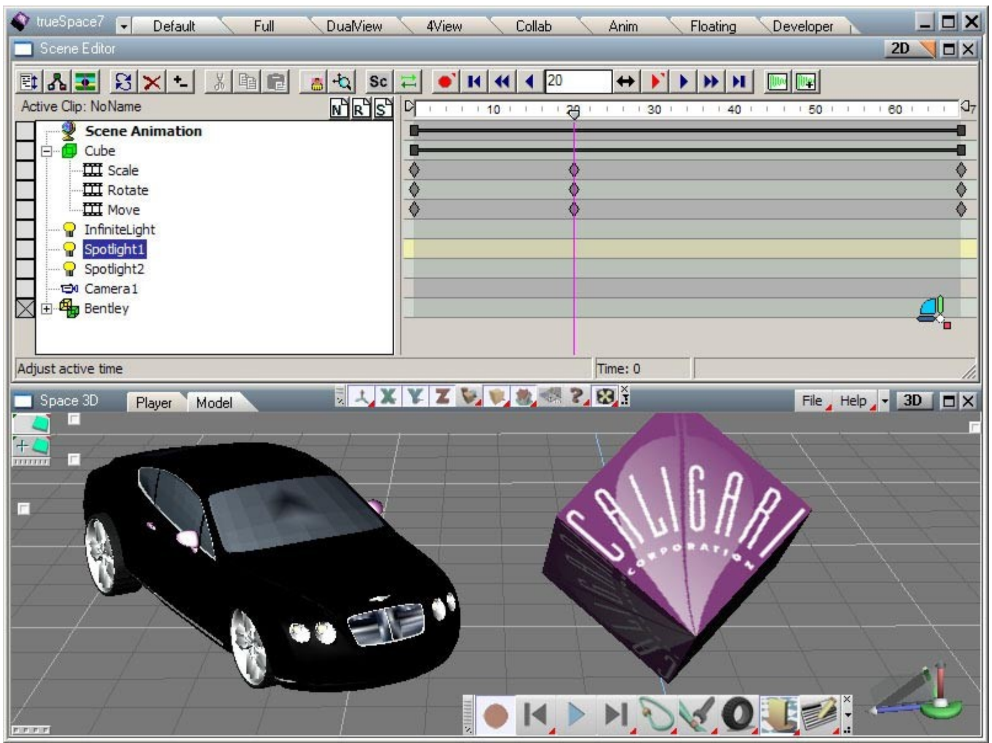
Chapter 6

Animation

6.1 Animation



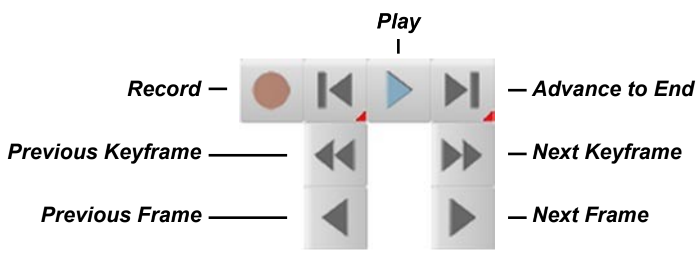
This chapter describes the animation tools in trueSpace.

6.1.1 Animation Control Group

By default, trueSpace places the Animation Control group on your toolbar when you start the program. Here, you will find a variety of VCR-like controls for playing animations, recording key frames, and moving between animation frames. Unlike traditional animation techniques (such as cel drawing and stop-motion photography), where one must draw every frame to create motion, in trueSpace you define the important points in time where actions take place (known as keyframes) and the program then draws all the in-between frames. Many types of actions can be keyframed, including movement, rotation, scaling, deformations, camera zooms, look at and ahead constraints, shader and material attributes, edge/vertex/face translation, and Inverse Kinematics/Bones manipulation. To set a keyframe for an object, simply move to that frame and manipulate the object; to keyframe texture changes, select the desired frame number, make changes to your shader settings, and use the **Paint Over** tool to apply the modified surface (also see the material editor documentation for information on trueSpace's animated material capabilities). When used in conjunction with the **Object Move** tool, creating keyfames for position also creates an animation path for the object which can then be displayed with by selecting the **Animation Path** tool.

In addition, keyframes may be recorded into clips, which can easily be expanded, contracted, moved, and blended with other clips to compose complex animations in a non-linear fashion. For example, one clip may contain keyframes of a character moving his legs as if walking. Another clip may be recorded just to move the character across the grid. A third clip could be recorded to swing the character's arms. With the "walking" and "swinging" clips shortened and set to repeat, the animations would blend together, and the character would appear to walk across the screen.

This is the full Animation Controls group, offering access to most of trueSpace's animation functionality without cluttering up your workspace.



Return to Start

Jumps to the animation's first frame, or to the first keyframe of the selected object. If a specific animation parameter (such as Move) is selected, clicking this button jumps to the first keyframe in that parameter.

Reverse to Previous Keyframe

Clicking this control jumps the Current Frame Number to the previous keyframe for a scene or object. If a specific animation parameter is selected, clicking this button jumps to the previous keyframe for that parameter, ignoring other animation tracks.

Reverse to Previous Frame

Play

Plays the animation to screen in wireframe or solid mode depending on the current screen mode.

Advance to End

Advance To Next Keyframe

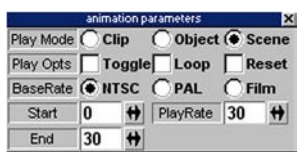
Jumps the Current Frame Number to the next keyframe for a scene or object. If a specific animation parameter is selected, clicking this button jumps to the next keyframe for that parameter, ignoring the other animation tracks.

Record Key

Sets a key frame for animated parameters for the current object at the current frame. The type of key set depends on the current tool active. There are two modes for the **Record Key** tool and are set by a right-click on the record tool. With Autorecord enabled, a key is set every time an object is manipulated at a frame other than 0. If Autorecord is disabled then the **Record Key** must be pressed each time you wish to create a keyframe.

Advance to Next Frame

Animation Parameters Panel



Right-click on the **Play** tool to open the Animation parameters panel. Settings here include the animation start and end frame, whether the current object or the entire scene is animated, looping controls, and base rate controls.

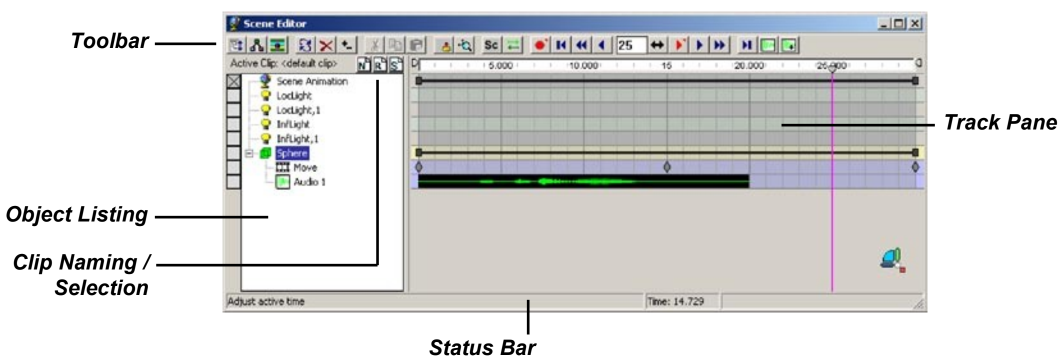
- **Play Mode:** This determines whether the current clip, the current object, or the entire scene (all objects) is animated when the **Play** button is activated. If two or more windows are open, including the main view, the object animates in all windows. Scene animates only in the active window. **Note:** Clip-view defaults to scene mode; key-view defaults to clip mode.
- **Play Opts:**
  - **Toggle:** Plays the animation forward then backward once.

  - **Loop:** Loops the animation during screen playback only. The ESC key or the double right-click interrupts play back. It can be combined with Toggle.
  - **Reset:** When enabled, trueSpace will return you to frame 0 after animation playback has completed.
- **Base Rate:** The rate in fps (frames per second) that the animation will play at. The choices are 30 fps for NTSC, 25 fps for PAL, and 24 fps for film.
- **Start/End:** These values show the current start and end frames for animation preview. By default Start is set to 0, the first frame, and **End** is set to the last frame in which animation for the currently selected object ends. You can also change these in the Project window.
- **PlayRate:** This limits the real-time playback rate to the specified number of frames per second. trueSpace will play back frames as fast as it can draw them (dependant on hardware), not exceeding this rate.

6.1.2 Overview of the Scene Editor

Scene Editor

Click on the **Scene Editor** tool to open the Scene Editor (also known as the SE for short), which combines all the trueSpace animation functions within one window. In addition to giving you complete control over all aspects of your animation, this powerful utility gives you the ability to edit your scene in much the same way you would use Windows Explorer to manage your hard drive.



When you open up the Scene Editor, you'll notice it's divided into five parts:

- The **Toolbar** (across the top of the SE), which contains a row of icons for editing, managing, and previewing your scene's animation;
- The **Object Listing** (the left pane below the toolbar), a hierarchical display of the objects in your scene;
- The **Clip Naming/Selection** functions which can be used to create, rename, and change active clips;
- The **Track Pane** (to the right of the **Object Listing**), which contains a timeline of all animation in your scene; and

- The **Status Bar** (across the bottom of the window), which displays information on the current frame number, including running time and total number of frames of animation in your scene. When moving your cursor over the Toolbar, the Help field to the far left displays the name of the tool your cursor points to.

The Scene Editor behaves in every way like a standard Microsoft Windows window. To resize the Scene Editor, click and drag on any of its four corners. To minimize, maximize, or close the Scene Editor, click on the appropriate buttons in the title bar. To resize the Object Listing or Track Panes, click and drag on the divider which separates these panes. By right-clicking on any object in the Object Listing, you can rename, copy, delete, or render the selected object. (If you have used Windows Explorer, you'll probably catch on to these conventions quickly. See "The Object Listing" for more information.)

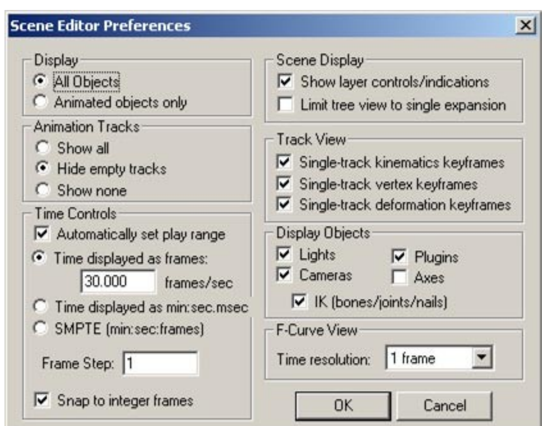
Note that when you minimize the Scene Editor, it is nestled into the main title bar of trueSpace. You can restore the Scene Editor by either by clicking on the Restore button or the Scene Editor tool.

The Toolbar

On the toolbar you will find all the tools necessary for editing keyframe locations (such as **Copy**, **Paste**, and **Delete**). You will also find controls to manage the contents of the Track Pane.

Scene Editor Preferences

Clicking on the Scene Editor Preferences button opens up the preferences dialog box. (**Note:** The Scene Editor preferences dialog can also be opened by right-clicking the **Scene Editor button** from the animation toolbar.)



- **Display:** This field allows you to select which objects show up in the Scene Editor. Choose between All Objects, or Animated Objects Only.
  - **All Objects:** When selected, the Scene Editor displays every object in your scene.
  - **Animated Objects Only:** This option restricts the information in the Scene Editor to only those

objects which are animated (contain keyframes).

Note: For relatively simple scenes, it is a good idea to select All Objects to give you easy access to all the objects in your scene. However, with more complex scenes containing large amounts of animation, consider selecting Animated Objects Only so you don't have to hunt through the Scene Editor for a specific object.

- **Animation Tracks:** The options in this field let you select which animation properties are displayed in the Track Pane. Choose between Show all, Hide empty tracks, or Show none. (For more detailed information on the information displayed, see the section titled "The Track Pane" later on.)
- **Show All:** Displays the complete range of animatable properties for all the objects in your scene, even those properties not currently keyframed.
- **Hide empty tracks:** Restricts the Scene Editor display to only those properties which are animated. (For example, if an object is set to rotate in your animation, the Scene Editor will only display rotational keyframes, and ignore other properties such as movement and scaling.)
- **Show None:** When this option is selected, only the objects (and sub-objects) in your scene are displayed in the Object Listing. All actions associated with that object are combined into one action bar in the Track Pane. (For example, even if you set both movement and rotation for an object, you will see only one action bar to the right.)

- **Time Controls**
  - **Automatically set play range:** When enabled, the play range automatically adjusts to animation changes. For example, if an operation in the SE increases the length of the animation beyond the initial play range, the play range adjusts to include all the animation.
  - **Time displayed as frames:** When enabled, the track view markings and status bar display reflect the current position as frames.
  - **Time displayed as min:sec.msec:** When enabled, the track view markings and status bar display reflect the current position in minutes, seconds, and milliseconds.
  - **SMPTE (min:sec:frames):** When enabled, the track view markings and status bar display reflect the current position in minutes, seconds, and frames.
  - **Frame Step:** The number of frames the time will be changed when the Advance to Next Frame or Reverse to Previous Frame buttons are clicked on the SE
  - **Snap to integer frames:** When enabled, frames, active time, etc., will snap to integer frames while being dragged within the SE.

- **Scene Display**
  - **Show layer controls/indications:** Show or hide the layer color swatches from the left side of the Object Listing.
  - **Limit tree view to single expansion:** When checked, the tree view will allow only one branch of the tree to be expanded; the branch that is expanded will the one that contains the selected object.

- **Track View**
  - **Single-track kinematics keyframes:** When checked, kinematics keyframe types will not

be displayed as multi-segment, but rather as a single consolidated track.

- **Single-track vertex keyframes:** When checked, vertex keyframe types will not be displayed as multi-segment, but rather as a single consolidated track.
- **Single-track deformation keyframes:** When checked, deformation keyframe types will not be displayed as multi-segment, but rather as a single consolidated track.

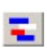
- **Display Objects:** The check boxes in this field let you determine whether lights, cameras, plugins, axes, or IK objects will be visible in the Scene Editor. Click in the appropriate boxes to toggle visibility on and off. If your scene contains animated light sources, cameras, or plugin filters, then by all means, check those boxes. This will give you maximum control over these components.

- **F-Curve View**
  - **Time resolution (1, 1/2, 1/4, 1/8 frame):** Determines how smooth the curves are drawn by setting how often parameter values are evaluated for f-curves. Smaller values will result in smoother curves in the F-Curve view.

Tree mode menu

Key View (Tree View): This is the original hierarchical view of the objects in the scene. Objects are grouped by parent-child association, and siblings are sorted by layer.

Objects by Layer: This is a single-level view of the objects sorted by layer. No hierarchy relationships are shown, and only objects with actual geometry are shown. If you use the layer paint control to move objects to a different layer, the objects will be re-sorted by layer as soon as you release the mouse button. Dragging and dropping one layer node onto another will merge layers.



Clip View: This view causes the right pane of the SE to switch to clip view mode, displaying all the clips in the scene, including their positions and lengths. Clips may be manipulated in this view.

A note on the distinction between Key View and Clip View: "Key view" refers to the hierarchy and layer views. Both of these views allow direct manipulation of keyframes, and are therefore considered key views. In Clip View, objects only appear in the view if they have animation.

Animation mode menu

 **Track Animation View:** Displays the track view of object animation.

 **F-Curve Animation View:** Displays the f-curves for the currently selected object.

 **No Animation View:** Hides the area normally occupied by the track or f-curve views. If the track or f-curve views are selected again, the SE will be expanded to its previous size.

Refresh

Refresh updates the Object Listing and Track Pane with the most current information in your scene. This button is especially useful after making a number of changes to your scene or animation. When clicked, it collapses the Object Listing down to the Scene level. Use the Expand/Collapse control to view all objects in the Object Listing after using Refresh.

Delete

The Delete button serves three purposes. If you have selected an object in the Object Listing, clicking the Delete key erases the object itself. If, on the other hand, you have selected an animation parameter from the Object Listing, the Delete key wipes out all keyframes in the currently selected animation track. Finally, if you have selected keyframes from within the Track Pane, clicking the Delete button will delete these keyframes and leave unselected ones intact.

For example, to eliminate all movement in an animated object, click on the object's Move parameter in the Object Listing, then click the Delete button. The track is cleared, allowing you to start from scratch.

Expand / Collapse Tree Control

Click this button to toggle the Object Listing and Track Pane between Scene view (all objects and key - frame parameters scrunched into Scene level) or fully expanded (all objects and animation parameters made visible).

Cut

The Cut button removes the currently selected keyframe(s) or clip(s) from the Scene Editor and places them into the Scene Editor clipboard. Cut keyframes or clips may be Pasted into the same animation track later. See "Using the Track Pane" for more information on selecting keyframes.

Copy

Clicking this button copies the currently selected keyframe(s) or clip(s) and places them into the Scene Editor's clipboard. The selected keyframes or clips remain in place. Copied keyframes and clips may then be Pasted into the same animation track later. Note that if you select an object instead of a keyframe, the Copy command makes a copy of the entire object (in essence duplicating the function of the Copy tool).

Paste

Clicking the Paste button pastes keyframes or clips from the Scene Editor clipboard into the currently selected animation track. Note that you can Paste keyframes of the same type from one object to another - for example, you can paste a Move keyframe from one object into the Move track of a second. The pasted keyframes appear at the point of the Current Frame marker found at the top of the Track Pane.

Note: The following keyframe types cannot be copied to the Scene Editor clipboard: IK, vertex, NURBS, and deformation. These can, however, be copied using clips.

Layer Paint/Selection

When this button is down, layer painting is enabled. Clicking and holding this button displays the 2D layer selection palette. If you choose a layer, it becomes the active layer; choosing the "+" option creates a new layer and selects it as the active layer. Once layer painting is enabled, you may move objects to the active layer by clicking or clicking and dragging over the color swatch for several objects.

See section **3.2.2** for details on manipulating layers within the Scene Editor.

Fit Zoom

The Fit Zoom function adjusts the time scaling of the animation view to fully contain all the visible animation tracks within range of the view.

Play Mode

This serves the same function as Play Mode in the Animation Parameters panel described earlier in this chapter.

Cli **Clip:** Only play animation for the current clip.

Ob **Object:** Only play animation for the current object.

Sc **Scene:** Play animation for the entire scene.

Reverse

Clicking the Reverse button flip-flops the action selected in the Object Listing. If keyframes are selected in the animation view, only those keyframes will be reversed. If one or more objects are selected in the Object Listing, all the animation tracks for all those objects will be reversed.

Scene Editor Animation Controls

The row of buttons on the far right of the Scene Editor are a VCR-style set of controls for browsing keyframes or previewing your animation. Many of these buttons are also found in the Animation tool group on your toolbar, but they've been incorporated into the Scene Editor for enhanced ease of use. Apart from the exception noted below, these controls work just like they do in the Animation Panel.

Note: It's good to be aware of the different functions of the Advance to Next Keyframe and Reverse to Previous Keyframe buttons. Whereas you can still jump through all keyframes on the currently selected object with each click by selecting the object itself from the Object Listing, you can also home in on a specific action and use these buttons to jump through those specific keyframes.

For example, say you've keyframed both movement and Inverse Kinematics (IK) actions on your object. By clicking on the Kinematic parameter in the Object Listing, the Advance to Next Keyframe and Reverse to Previous Keyframe buttons will only cycle through the Kinematic keyframes. This is an extremely handy way to focus strictly on those properties that need your attention. See "Using the Object Listing" for more specific information.

Additionally, the Record function can also be used to record keyframes for specific actions by selecting the appropriate action in the Object Track.

To choose whether trueSpace updates only the selected object or the whole scene, right-click on the Play button to bring up the Animation Parameters panel, and choose either Object or Scene. With the Object option selected, the Advance, Reverse, and Play buttons affect only the selected object. If, on the other hand, you choose Scene playback, trueSpace updates the entire scene as you advance or reverse. Scene playback makes it easier to see how all your objects interact with each other; however, for more complex scenes, you may find it easier to work with only the selected object until you're happy with your results. This is because playing the entire scene demands more processing power and may become choppy on more complex scenes.

Record

Clicking the Record button records a keyframe at the current frame number. If you've selected the object itself in the Object Listing, all animated parameters are recorded. If you have only selected one parameter in the Object Listing, trueSpace records a keyframe for that parameter and no others.

Note: You can also enable the Autorecord function, which sets keyframes automatically every time you make a change. Right-click on the Record button to open the Set Keyframe panel; here, you can toggle

AutoRecord while in Key View or Clip View by clicking in the box. An ‘X’ in the box indicates that AutoRecord is enabled, whereas a blank box means AutoRecord is turned off.

You can also select which keyframe types will be recorded when the SE record button is left-clicked.

Return to Start

Clicking the Return to Start button sets the current frame number to 0. Its twin, Advance to End, shoots the current frame number to the last set keyframe in your scene.

Reverse to Previous Keyframe

Jumps the Current Frame Number to the next previously-set keyframe for a scene or object. If a specific animation parameter is selected, clicking this button jumps to the next keyframe for that parameter, ignoring the other animation tracks.

Reverse to Previous Frame

Steps the Current Frame Number back by one frame.

Current Time Box

Nestled between the Reverse To Previous Frame and Play buttons is the Current Frame Number box. The value in this box represents the current time. To change this value, type a new number into the box, or click and drag the time marker at the top of the Track Pane.

Play

Click the Play button to preview your animated object or scene. You can restrict playback to certain frames by clicking and dragging the Start and End frame markers at the top of the Track Pane, or by setting numerical values in the Animation Parameters window (brought up by right-clicking the Play button). **Note:** The Animation Parameters window appears in the main workspace, so it is possible that it might be hidden behind the Scene Editor.

Advance to Next Frame

Increases the Current Frame Number by one.

Advance to Next Keyframe

Clicking the Advance To Next Keyframe button shifts the current frame number ahead to the next key-frame of your selected object or action. The currently selected keyframe will light up in the Track Pane.

Advance to End

Moves the Current Frame Number to the last keyframe

Audio Controls

Audio Compiler

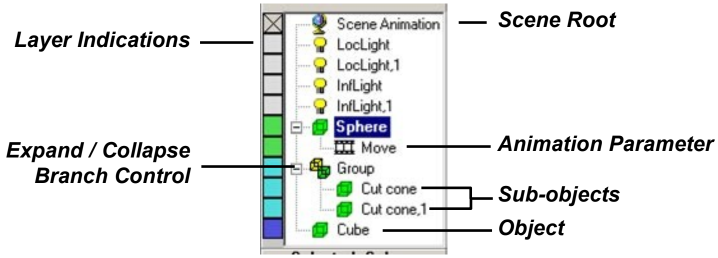
See section **6.1.4** for details.

Show/Hide Scene Sound Clip List

See section **6.1.4** for details.

The Object Listing

The Object Listing, found in the left pane of the Scene Editor, is a hierarchical representation of all your scene's objects and their associated animation properties.



As you work with the Object Listing, you'll find that it's the ideal way to manage the objects in your scene - whether they're animated or not. The expandable and collapsible tree control makes it easy to view hierarchical information, and narrow in on the object or animation parameters you wish to edit.

Note: Scenes with many objects can cause the response time of the SE to be slow.

The scene tree works along the same conventions found in Windows, allowing you to make changes by right-clicking on your target, or using drag-and-drop techniques to glue and unglue these objects.

Expanding and Collapsing Branches

Grouped objects can be expanded by clicking the small box with the plus (+) sign inside next to the group's name. This box is your Expand/Collapse Branch control button. Whenever Expand/Collapse Branch button contains the + sign, clicking the button reveals all sub-objects or animation parameters. If, on the other hand, the control button contains a minus (-) sign, clicking it hides the underlying branch. Load any sample scene and take a few moments to expand and collapse various branches to get a feel for navigating the Object Listing.

In an animated scene, some of the lines in the Object Listing represent objects themselves, while others represent animation information. You'll be able to tell which is which by examining the icon to the left of the object or animation name. Click on any object or sub-object in the Object Listing to select it.

Viewing an Object's Animation Tracks

All animated objects in your trueSpace scene contain at least one Animation track as a child of the object, represented by a small icon resembling a filmstrip. To the right of the Animation track is an action bar, which represents keyframed properties for the current animation type. Notice that the object itself also has an action bar, representing the entire collection of keyframed properties attached to the current object; manipulating this bar affects all keyframes under this branch.

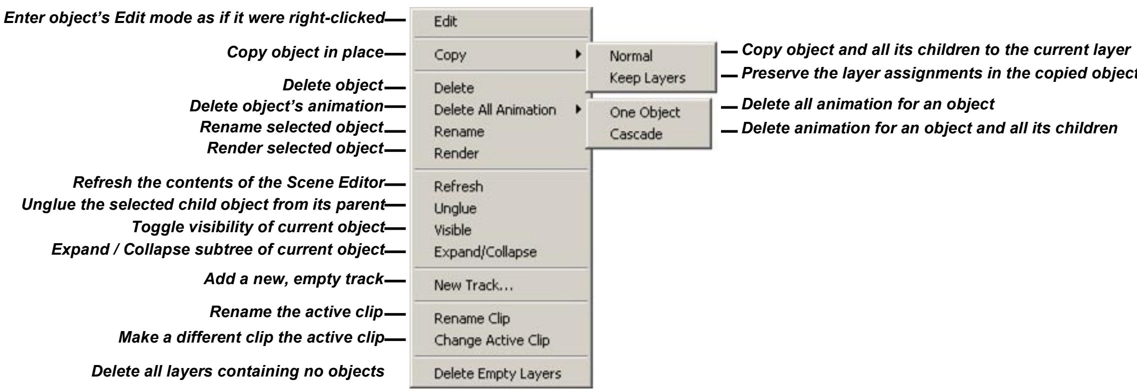
To select an action, click on the name of the action in the Object Listing. Now, when using the Advance To Next Keyframe or Reverse to Previous Keyframe button, the Scene Editor will only cycle through the keyframes set for that specific action.

To discover how to manage keyframes for your objects, see "Using the Track Pane" later in this documentation.

Right-clicking for Other Scene Editor Functions

Right-clicking on any object or action in the Object Listing brings up a popup menu which allows you to choose from several options for that object, including copying, deleting, renaming, rendering, or applying the Repeat/Reverse functions.

While most of these functions can be applied to objects and actions, the Copy command from the popup menu will make a copy of the currently selected object (and its attached animations), even if an action is currently selected.



Gluing and Ungluing

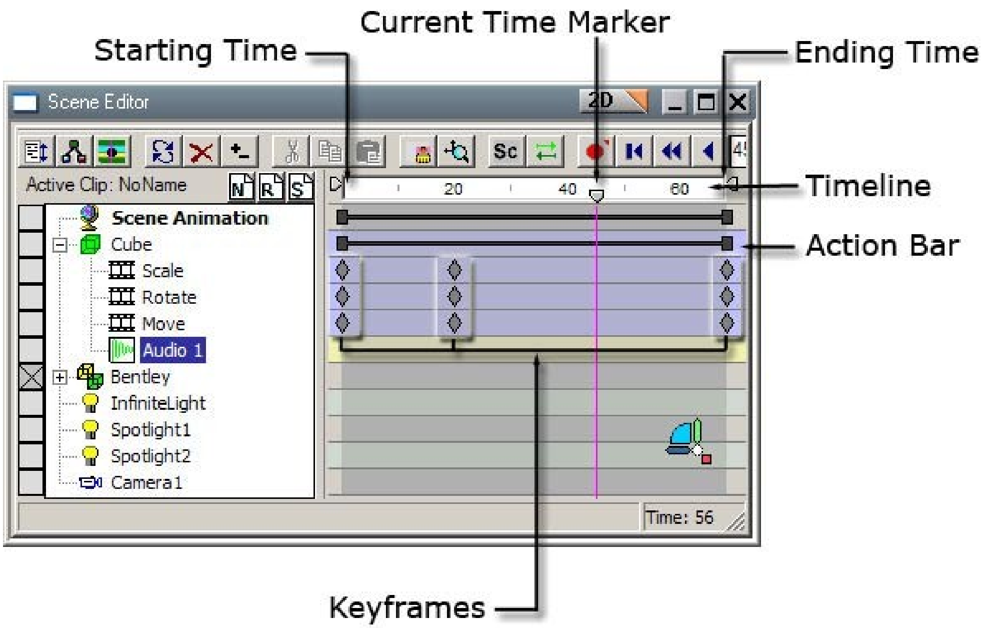
You can use the Object Listing in place of the Glue As... tools found in the main group tools. To glue two objects together, click and drag the first object over to the second, and release the mouse button. (In other words, "drag and drop.") This creates a glued hierarchy that can be managed normally from the Object Listing. To unglue an object from the hierarchy, drag and drop the object back into the scene icon.

Scene Keyframes

Scene keyframes include plugins, fog, ray tracing, backgrounds, and global environments. Scene keyframes are set by changing global scene parameters at a frame other than 0.

- **Fog Keys:** Color, fog extents, and maximum percentage can be keyframed, as well as the on and off state.
- **Background Keys:** A background bitmap may be keyframed on and off as well as changed during an animation. Changing image files does not create a smooth transition. Background color can also be keyframed.
- **Global Environments:** An environment bitmap may be keyframed on and off as well as changed during an animation. Changing image files does not create a smooth transition. Environment color can also be keyframed.
- **Ray Tracing Keys:** Ray Tracing may be keyframed on and off during an animation. The reflect value can also be keyframed.

The Track Pane



The Track Pane is a timeline that gives you an overview of all animation information of the objects in your scene. Here, you have the ability to view or manage entire actions, or dig in and record or edit individual keyframes (or groups of keyframes).

In the next section, you'll find an overview of ways to use the Scene Editor to achieve more specific goals; in the meantime, here is a quick overview of the components you can see in the above figure:

Timeline

The timeline, found at the top of the Track Pane, measures your scene's time in frame numbers or time units, depending on the Time Controls settings in the SE's Preferences. To compress the timeline (in other words, to view more frames within the same space), click the red (zoom) portion of the SE's view control, and drag to the left. To expand the timeline (focus in on a specific segment of the timeline), drag your mouse to the right. (Zoom can also be performed by right-clicking and dragging on the blue portion of the Scene Editor's 3D control.) The timeline is colored white between the Starting and Ending Frame markers, and gray outside.

Current Time Marker

The downward-pointing arrow at the bottom of the Timeline is your Current Time marker. A purple vertical line shooting down from this marker helps you determine the relative position of your individual keyframes. Clicking in the timeline will set the Current Time Marker to that position.

Starting/Ending Time Markers

Click and drag the Starting and Ending Time markers to set the begin and end limits for previewing your animation via the Play button. This duplicates the function of setting these limits numerically in the Animation Parameters panel.

Action Bars

Action bars extend to the right of the Scene line, animated objects, and/or glued objects. They are book-ended by two blocks which indicate the starting and ending time of that particular action. You can shorten, lengthen, or delay the timing of the action by clicking and dragging on these blocks. You can also shift the entire action to occur sooner or later by clicking and dragging on the bar itself. By modifying the action bar of the Scene line, you affect the timing of all animated objects.

Keyframes

The diamond-shaped markers in the Keyframe tracks represent individual keyframes. No doubt you are going to find it much easier to edit these keyframes using the wide array of options now available to you. You can easily shift the position of keyframes on the timeline by clicking and dragging left and right; you can also cut, copy, and paste these keyframes using the tools in the Scene Editor toolbar. Selecting keyframes is a relatively simple affair; click on a keyframe to select it You can also select groups of keyframes by clicking and dragging a rectangular area within the Track Pane. Selected keyframes are highlighted in white, while unselected keyframes are colored gray.

6.1.3 Setting and Editing Keyframes and Actions

The Scene Editor offers an enhanced set of tools for editing keyframes you've set, giving you a visual overview of all individual animation parameters. In this section, you'll discover a few ways to animate your objects and edit those actions later on.

Recording Keyframes

To animate an object:

1. Set the current frame number for your target location. You can see the current frame number in one of two ways:
    a. By typing the number in the Current Frame Number box and pressing ENTER, or
    b. By dragging the Current Frame marker at the top of the Track Pane to the desired frame on

the timeline.

2. Place your object in its new position. In other words, move, rotate, scale, deform, point-edit, or use Inverse Kinematics so your object ends up where you want it. (Note that to animate an object's material, you must use the Paint Over tool to apply the new surface.)
3. If the AutoRecord feature is turned off, you must hit the Record key to set the keyframe manually.

Note: Scaling also creates Move keyframes, since the center point of the object may move during scaling, like when dragging one of the corners of the selector box control.

Selecting Keyframes

In its collapsed state, an object's Animation line only allows you to decrease or increase the time taken to perform the action, or shift the action to occur sooner or later in your animated scene. (More on that below.) To select an individual keyframe, you must first expand the main branch of your object to view the keyframe tracks which contain the individual keyframes.

As you move your cursor over the keyframes and action bars, note how it changes shape depending on where you are pointing. This helps give you a visual indication of what will happen if you click and drag at that particular point.

To select an individual keyframe, click on the diamond-shaped marker which represents the keyframe. It should now turn from gray to white. To deselect a keyframe, click anywhere in the white space of the Track Pane, and the marker reverts to gray.

To select multiple keyframes, hold down the CTRL key and click on the desired keyframe(s). Alternately, you can click and drag out a rectangular area that encompasses the keyframes you would like to select. Start your selection away from the keyframes themselves - your cursor should appear in its normal, unaltered state. Otherwise you may inadvertently drag the keyframes themselves to a new position on the timeline.

The selected keyframes may now be shifted left and right by clicking and dragging, or copied to the clipboard using the Cut or Copy buttons. You can also unset these keyframes by pressing the DEL key on your keyboard.

Moving Keyframes

Simply click and drag on any selected keyframe to move all selected keyframes forward or back on the timeline.

Cutting and Copying Keyframes

Once keyframes are selected, you may move them from the animation track to the clipboard by clicking the Cut button. Or, you can leave the keyframes in place and simply copy them to the clipboard using the Copy button. As with many other Windows applications, these commands are complemented by the Paste button.

Pasting Keyframes

Use the Paste button to insert keyframes into an animation track. You can paste keyframes into the same object from which they were copied, or into another object entirely. One hitch: You can only paste the same type of keyframe into any keyframe track. (For example, if you have a series of Scale keyframes sitting in the clipboard, you can only paste them to a Scale track.) If you have selected an object line in the Object Listing, the keyframes will automatically nest in their proper tracks. Here is the best method for making sure pasted keyframes end up in the right place:

1. Select the proper object or keyframe track in the Object Listing. If the particular track is not visible on the target object, expand the Animation line on that object using the Expand/Collapse control button.

2. Click and drag the Current Frame marker to the frame number you want to paste the keyframe(s). If you have copied a series of keyframes, the position of the Current Frame marker is where the first keyframe in the series will be pasted.

3. Click the Paste button to place the keyframes into the selected object or track. The keyframes now show up where they've pasted them. (You may not see them right away; click the Refresh button to update the display in this case.)

Note: If you select multiple objects in the Object Listing, the Paste operation will be performed on all of them at once.

Erasing Keyframes

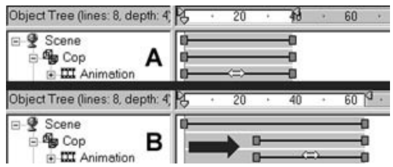
Selected keyframes can be removed by pressing the DEL key on your keyboard, or by right-clicking in the right-hand pane and selecting “Delete” from the popup menu.

Re-timing Actions

As mentioned earlier, an action is a collection of keyframed parameters. The Scene Editor represents actions with a solid bar (known as an action bar) in the Track Pane. If, for example, you have animated an object that both moves and rotates, manipulating the Action Bar controls both movement and scaling. There is even an action bar at the Scene level which allows you to re-time actions for everything in your trueSpace world.

To make an action occur sooner or later, click and drag anywhere on the bar itself to slide it left and right, using the timeline as your guide. (Before dragging, make sure your cursor has turned into a double-arrow that points left and right.) This is perhaps the best way to delay the start of an object's animation.

In example A below, the action of object Cop starts at frame 0 and ends at frame 40. After clicking and dragging the Animation action bar (the one on the very bottom of the example with the double arrow), the action now begins around frame 25 and ends at frame 65 in example B.



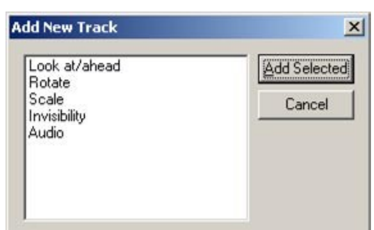
To stretch out or compress an action (in other words, shorten or lengthen the amount of time it takes), click and drag on either of the blocks which bookend the action bar, using the timeline as your guide. (Your cursor will turn into a line with an arrow pointing left or right as your cursor hovers over the blocks.)

If you happen to drag an action so that it extends beyond the last frame of animation, trueSpace automatically updates the Scene action bar and the End Frame marker to reflect these changes.

Note: When the Snap to Integer Frames option is checked in preferences, keyframes will not be spaced closer than one frame. This has a big effect on physics animation, where keyframes are already one frame per keyframe, so you cannot compress this kind of animation while in snap-to-integer mode. With the Snap to Integer Frames unchecked, even physics animation can be compressed, since the keyframes are no longer constrained to integer frame times.

Adding new (empty) tracks

You can add empty tracks to an object in the Object Listing by right-clicking on the object and selecting "New Track…" which will open the dialog below:



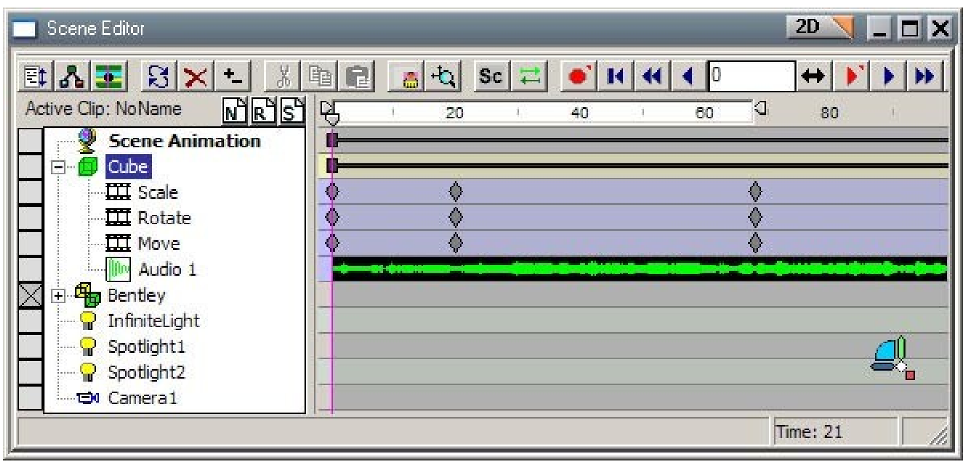
You can select as many of the track types as you like, and clicking "Add Selected" will cause the selected track types to be added to your object. You can also double click on a track type in the list to add just that track type without having to click the "Add Selected" button.

Multiple audio tracks (described in the next section) may be added to an object. There is no set limit on the number of audio tracks that may be assigned to an object.

6.1.4 Audio Tracks

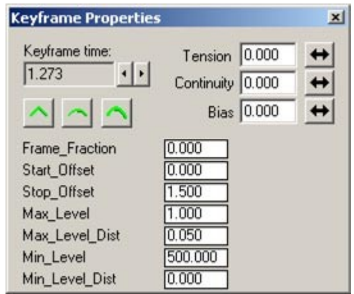
Audio clips are handled in the following manner: the scene maintains a set of audio clips; these clips may be used any number of times in the scene (including zero). To view the scene clip list, click on the **Show/hide sound clip list** button. To add an audio clip to the scene clip list, click on the **Add New…** button on the scene audio clip list dialog. This will bring up a standard Windows dialog from which you can select a WAV file (or multiple, using CTRL and/or SHIFT) to add to the scene. Once a clip is added to the list, you can either right-click an audio track in the track view and choose **New** to select and paste an audio clip from the list, or drag the audio clip from the Scene Sound Clips window to the new track. (**Note:** Audio clips in the clip list are saved as part of the scene file and add to the size of the file.)

Audio keyframes display a waveform of the clip, and may be moved about as any other keyframe type. A sample track with an audio keyframe is shown below:



Sample track with an audio keyframe

Audio keyframe properties can be edited in the KF Properties dialog, just like any other keyframe. Below is what an audio keyframe looks like in the editor, with descriptions of each field:



- **Keyframe Time:** The frame at which the clip's time 0 occurs. Note that audio keyframes are not restricted to integral frame times, as are other keyframe types.
- **Start Offset:** The time, in seconds, relative to the beginning of the clip, when the sound will begin playing.
- **Stop Offset:** The time, in seconds, relative to the end of the clip, when the sound will stop playing.
- **Max Level:** The maximum amount by which a clip's sound level will be multiplied (in the example above, the sound will become no louder than 1.5 times its original level, no matter how close it is to the observer).
- **Max Level Dist:** The distance (in meters) at which the maximum sound level occurs.
- **Min Level:** The maximum amount to which a clip's sound level will be diminished (in the example above, the sound will become no quieter than 5% times its original level, no matter how distant it is from the observer).
- **Max Level Dist:** The distance (in meters) at which the minimum sound level occurs.
- **Tension/Continuity/Bias:** These are currently meaningless for audio keyframes, and are therefore disabled.

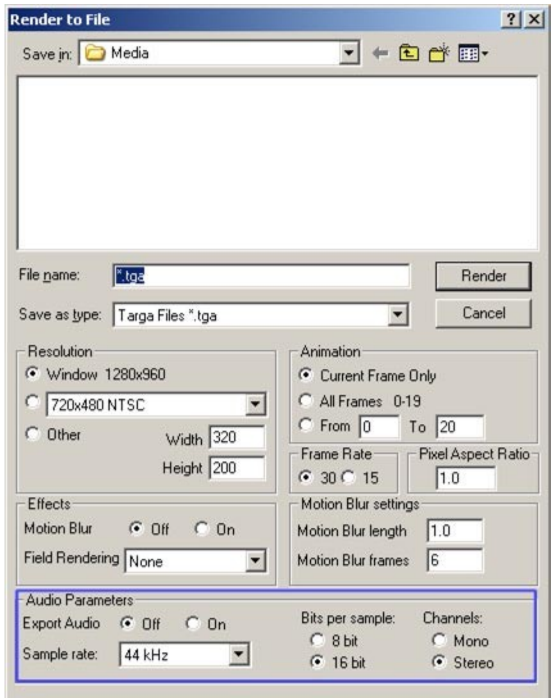
Start offsets <0 and stop offsets >0 will cause looping of the sound in the final clip, while start offsets >0 and stop offsets <0 will cause portions of the clip to be ignored. Below are some screen shots to demonstrate how keyframes with various start/stop offset values are displayed in the SE (screenshots forthcoming). The time range that will be rendered by the audio compiler is displayed with a black or gray background (black corresponds to the original clip, gray corresponds to looped sections of the clip), while ignored areas of the original clip have a red background.

Clip at frames 15-85, with start offset of -5, stop offset of +5 (the sound will be looped so that frames 10-90 will be included in the rendered audio).

Clip at frames 15-85, with start offset of +5, stop offset of -5 (only frames 20-80 will be included in the rendered audio).

Saving Audio

Audio clips are compiled and saved during the Render to File process. The compiled audio is stored either in a separate .WAV file (in the case of rendering to sequential files), or directly in the generated .AVI movie file (in the case of rendering to .AVI). All audio compiler options are accessible directly from the Render to File dialog:



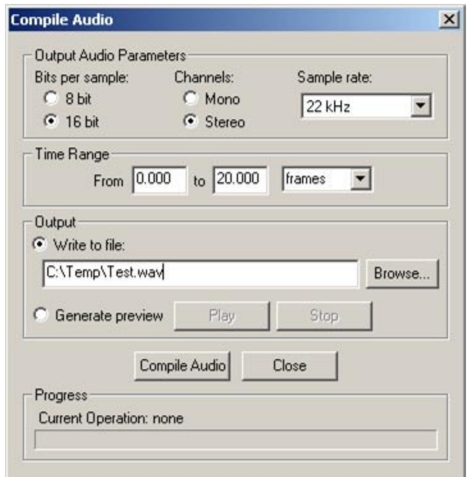
- **Export Audio:** Export audio (if available) or ignore it
- **Sample Rate:** 11kHz, 22 kHz or 44 kHz audio output
- **Bits per sample:** 8 or 16 bit audio output
- **Channels:** Mono or Stereo audio output

Note: The bit rate, audio sample size, channels (mono / stereo), and audio sample rate can be obtained in some versions of Windows by right-clicking the WAV file and clicking on the properties/summary tab.

Compiling audio

Once you have one or more audio keyframes in your scene, you may compile the scene's audio into a single WAV file. To open the audio compiler dialog, click on the audio compiler button in the SE taskbar, shown below:

When the audio compiler dialog comes up, you may specify the audio options, and choose whether you want to save the compiled audio to a file or just listen to it as a preview. The final audio will be 'rendered' for the range specified in the Time Range section.



Once all your parameters are set, click the "Compile Audio" button to begin the audio compilation. The compile operation consists of up to four steps:

- **Construct audio compiler network:** During this time, the Scene Editor sets up the audio render and steps through the scene animation frame by frame to get object locations.

  Tip: *Scenes with complex animation should be switched to wireframe mode so that it won't take as long to go through this step.*

- **Collect samples:** The compiler network is executed to add together all the audio clips. The progress speed may vary according to the number of audio clips and their overlap.
- **Normalize samples:** The collected samples are normalized so that they will all fit into the range you specified in your options (either 8 bit or 16 bit). Progress speed through this step should be reasonably constant (and fast).
- **Preview (if Generate Preview is enabled):**The progress bar updates, and the current operation shows "Playing audio preview" as the preview plays.
- **Save to file (if Write to File is enabled):**This will be another reasonably smooth step, although it's slow at the moment. Future optimizations should greatly improve the speed. (This step will be skipped if Generate Preview is selected).

If audio clips are attached to an object other than the scene, the audio compiler will calculate time delay and Doppler shift effects based on the distance from the object to the camera/eye. Audio clips attached to the scene will be merged into the final audio with no attenuation for distance. The left/right balance is determined by the position of the object relative to the eye (or camera object, if you have one); the audio compiler assumes that your ears are located at the eye/camera position.

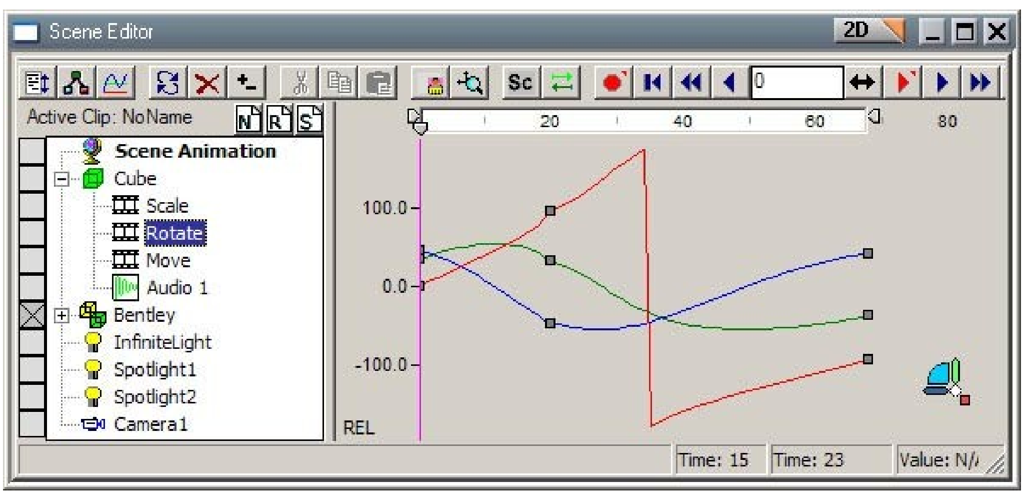
6.1.5 Function Curves

Function Curves tool

The Function Curves setting of the Animation mode menu turns the Track Pane into a function curve editor to allow you to fine-tune movement, scaling, and rotation keyframes for animated objects.

While in Function Curves mode, the Object Listing remains much the same, except that additional X, Y, and Z values are accessible for movement, scaling, and rotation. While in Function Curves mode, use the Expand/Collapse icons next to the object name to access these X, Y, and Z values. (Note that these will be easier to find if you use the Scene Editor filter to omit objects that are not animated.)

Below is a sample shot of the Function Curve mode of an object that has had movement keyframes assigned to it. In this example, the squares at frames 0, 10, and 20 represent movement keyframes for the cube object. The line that connects the keyframes is known as the function curve spline.



The spline curve represents the rate at which the object approaches the keyframes over time. The meter at the left side of the Track Pane displays the range of values for the keyframes. In the example above, the cube object has an X value of 5 at frame 10, and 0 at frame 20.

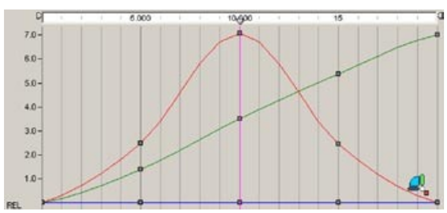
Editing Function Curves

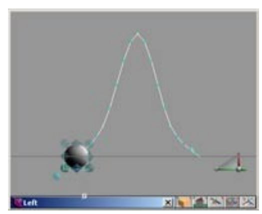
You have several options for the editing of function curves. Briefly, you can click and drag the keyframe to change its vertical position (thereby changing the value of the selected parameter); holding the Control key while you click and drag allows you to move the keyframe horizontally (thereby adjusting the frame number of the selected keyframe); or, you can double click on the keyframe to bring up control handles to adjust the tension, bias, and continuity of the spline curve at the selected keyframe.

Adjusting Keyframe Value

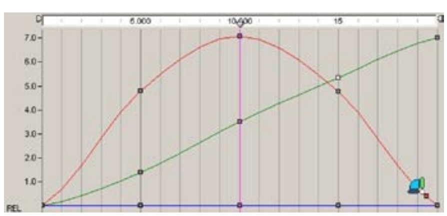
You can change the value of a keyframe, or multiple keyframes, by selecting the keyframe(s) and dragging up or down within the Function Curve editor. The meter on the left side of the Function Curve editor represents the floating point value of the keyframe.

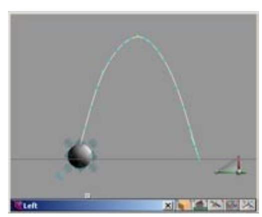
In the following example, we have an object that rises very quickly toward the middle of the animated sequence, "spiking" at frame 10. In the screen shot below, the Path command was used to make the animation path visible. Note the hump toward the middle of the path. The goal is to make the up-and-down motion more gradual over the course of the animation, turning the path into an arch. The first step is to select the keyframes immediately before and after frame 10. This is done either by using the CTRL key while clicking on each keyframe, or clicking and dragging a rectangular area within the Function Curve editor. Note how the keyframes are highlighted white.





With the keyframes thus selected, it is now possible to drag them further up to create a smoother curve going in to frame 10. As seen in the sample figure below, the path now takes a less dramatic up-and-down motion.





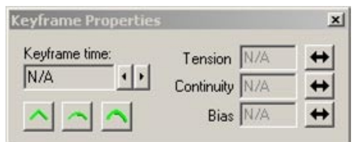
If no keyframes exist at points you would like to smooth, you can add additional keyframes by right-clicking at the desired frame position to bring up a menu window. Select "New" from this window, and the keyframe is added at the appropriate point on the spline curve.

Adjusting Keyframe Time

Sometimes, in order to create a smooth function curve, it is necessary to shift the frame number of an existing keyframe. To do this, hold down the Control key while dragging on a selected keyframe. Drag the mouse right to advance the keyframe to its new position, and drag left to move it back. Note that this also affects the position of the keyframe in the other two axes.

Keyframe Properties

Right-click on any keyframe and choose Properties to bring up the Keyframe Properties panel. Here, you can enter keyframe values manually, as well as set the standard Sharp, Smooth, and Very Smooth spline corner settings.



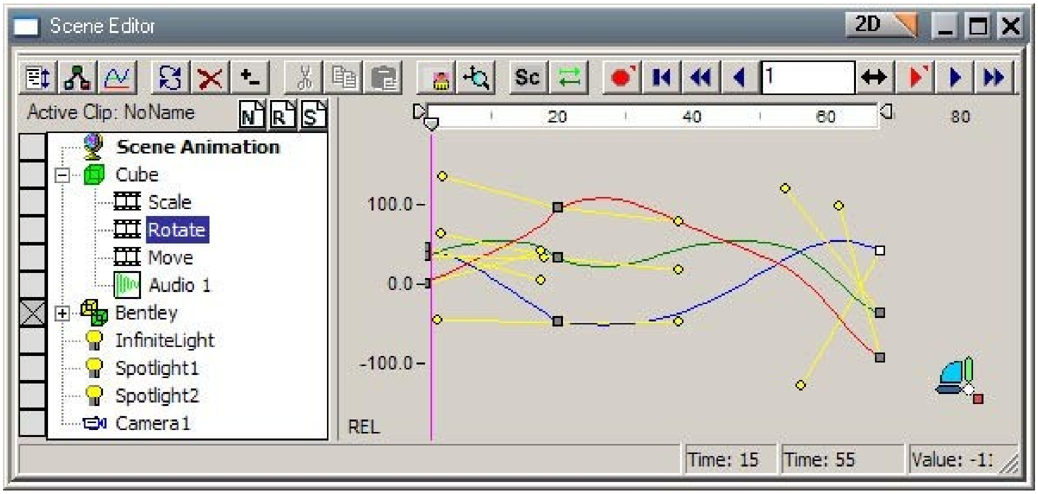
- **Keyframe Time:** This represents the frame number of the currently selected keyframe. Note that this field can only display one keyframe at a time; when multiple frames are selected, the keyframe time (and all other parameters) are displayed as N/A. The arrows to the right of the KF field allow you to jump from keyframe to keyframe.

- **KF Value: The**Keyframe Value displays the movement, scaling, or rotational value of the keyframe at the frame number displayed in the KF field. You can enter numbers directly into the box for a specific value.

- **Tension, Bias, and Continuity:**These values determine how smooth or how sharp the function curve spline eases into or out of the specified keyframe. Note that changes in these values will affect movement, translation, or scaling along all axes. Tension/bias/continuity can also be adjusted by double clicking on a keyframe and using the control handles that appear (see below).

Adjusting TBC Values Visually

Right-click a keyframe and choose Curve Handles to bring up control handles that work similar to those found in Spline Edit mode. Using these handles, you can change the shape of the curve as it enters and exits the selected keyframe.



Changing the orientation of a handle adjusts the bias value for the selected animation parameter - in other words, use handle direction to adjust the angle of the entry and exit trajectories for the selected keyframe.

Changing the length of the handle (by dragging the handle bar closer to or farther from the keyframe) controls the tension value for the selected parameter. In other words, use this value to adjust how quickly or how slowly the rate of movement, scaling, or rotation changes as frame numbers approach the selected keyframe. Lengthening the handles means that movement, scaling, or rotation occurs at a more rapid rate around the keyframe, while shortening the handles slows this rate.

Note: Because adjusting Tension, Bias, and Continuity values can often lead to over-dramatic variations along other axes, it is a good idea to right-click on a keyframe to open the KF Properties panel. By clicking on the Sharp, Smooth, or Very Smooth parameters, you can easily rectify problems that can occur when editing TBC values.

6.1.6 Layer Manipulation within the Scene Editor

The SE can display and modify object/layer associations. To turn the layer indications on and off, use the "Show Layer Controls/Indications" checkbox in the SE preferences.

To move objects to another layer, turn on layer painting by clicking and releasing the Layer Paint /Selection button, shown in the left image below. When the button is down, layer painting is enabled. Clicking and holding this button displays the 2D layer selection palette. If you choose a layer, it becomes the active layer; choosing the "+" option creates a new layer and selects it as the active layer.

Once layer painting is enabled, you may move objects to the active layer by clicking or clicking and dragging over the color swatch for several objects.

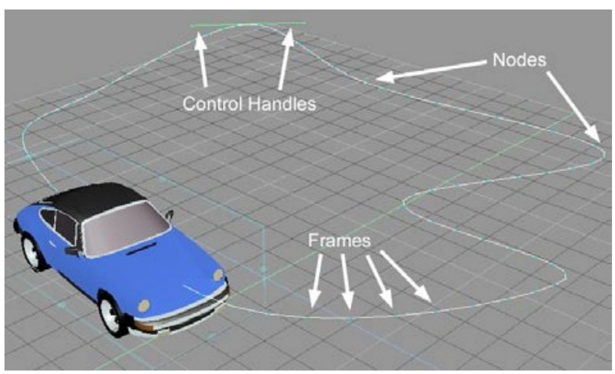
Alternatively, in the Objects by Layers tree mode, simply drag and drop objects onto the layers listed in the Object Listing.

For more information on manipulating layers within the SE, see section **3.2.2** in **Artist Guide Chapter 3: Modeling.**

6.1.7 Animation Path

Path tool

An animation path is created the same way as a spline. To draw an animation, select the target object and then select the **Animation Path tool**. Left-click in the workspace to create a path vertex. Each point you define sets a new spline node for the path, with an intermediate number of interpolated frames in between (there are thirty frames per second of animation). New spline paths can be created from any view. Points drawn in either Perspective or Top view are placed flat on the grid. While drawing a spline path, you can switch between adding new points and editing existing ones using the **Draw** and **Edit** controls. Spline paths are edited in the same way as spline. To close a spline click the right mouse button, which connects the last point with the first. At this point, you can edit the polygon or add the spline to a paths library for later use, but you cannot add new spline points to a closed spline path except by adding position key-frames with the SE. It is not necessary to close an animation path; exit by de-selecting the **Animation Path** tool. Also, you can add or change a path spline to the workspace by selecting a path (closed or open) from the **Path library** after selecting the **Animation Path** tool.



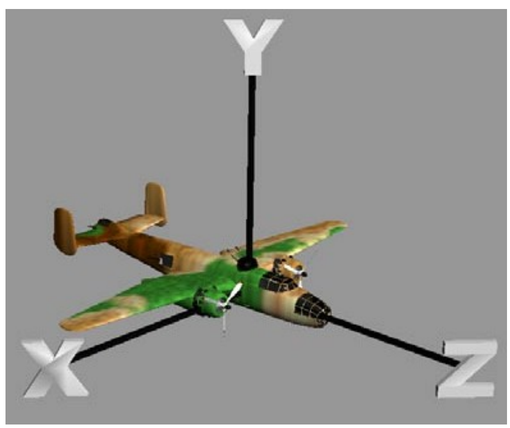
Look At

This command is used to make an object, camera, or light continually realign itself during an animation to remain pointed at another object. The program accomplishes this by automatically pointing the original object's Z axis at the second object's axis location (see the Axes tool). To use **Look At**, select the object that is to be constrained, select **Look At**, then select the target object. When an object is under **Look At**'s influence, it is constrained and cannot be rotated manually. **Look At** and **Look Ahead** are mutually exclusive operations.

The **Look At** tool has two states that can be keyframed: enabled (icon pushed in) and disabled (icon in default state). When enabled the constrained object will continually update its orientation during an animation to that of the target object. In disabled state the object will cease to update itself at the “disable” keyframe. Subsequently the object can then be keyframed to look at a new target during the course of the animation. Note that keyframing a new target on the same frame as the old one will not create a smooth transition; rather, the constrained object will jerk suddenly to re-orient itself. The proper action in this situation is to skip ahead many frames and then keyframe the new Look At target.

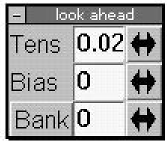
Look Ahead

This command is used to make an object point forward along its motion path during animation. The program accomplishes this by automatically pointing the object’s Z axis along the path, re-aiming for each frame. To use **Look Ahead**, select the object to be constrained and then the **Look Ahead** tool. Objects like text, cameras, and spot lights will work automatically with **Look Ahead** but other objects will need to be adjusted to work properly (see the Axes tool). The **Look Ahead** tool requires that the object’s Z axis points to the “front” of the object and that the Y axis points towards the “top” of the object. When an object is under the influence of the **Look Ahead**’s command, it is constrained and cannot be rotated manually. **Look Ahead** and **Look At** are mutually exclusive operations.



Look Ahead has two modes that can be keyframed: enabled (pushed in) and disabled (default state). When enabled, the constrained object will orient itself to the path until a “disable” keyframe is set. Subsequently the object can then be keyframed to look ahead again or to look at a target with the **Look At** tool. Note that keyframing a **Look At** action on the same frame as **Look Ahead** was disabled will not create a smooth transition, rather the constrained object will jerk suddenly to re-orient itself. The proper action in this situation is to skip ahead many frames and then keyframe the **Look At** target. To set **Look Ahead** parameters, right-click on the **Look Ahead** button.

Look Ahead Control Panel



- **Bias:** Determines the amount of "overshoot" when the object moves around curves.
- **Tension:** Determines the degree by which the object tries to follow its original orientation.
- **Bank:** When enabled, the animated object "leans" into curves in the motion path. The bank amount can be specified numerically by degrees.

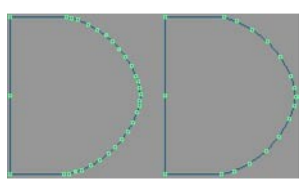
Curve to Path

This tool converts a NURBS curve to an animation path. Select the object you wish to attach the path to, and then activate the tool. trueSpace will prompt you to select a NURBS curve in the workspace. The selected curve is converted to animation path and attached to currently selected object, and the animation path editor is then activated.

Right-click this tool to open the Curve Conversion Options panel:

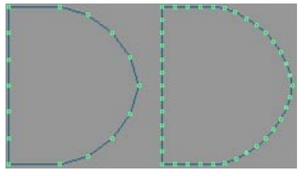
There are three ways how to convert curve to animation/sweep path:

- **Track curve:** When Track curve is selected, the Angle threshold value is used. Curve curvature is tracked, and the frame is created each time the tangent deviates from last tangent more than Angle threshold. This yields more frames in curved parts of the curve and less parameters in straight parts. This is a useful way to create a macro sweep path.



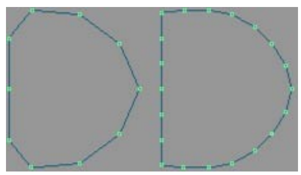
Tracked curvature for Angle 5 and 10 degrees

- **Homogenous path:** When Homogenous path is selected, the Subdivisions value is used. The curve is homogeneously divided into frames, and sharp corners are preserved. Higher Subdivisions values yield larger distances between frames. As in the previous mode, the number of frames depends on curve curvature.



Homogenous path for Subdivisions 1 and 2

- **Divide curve:** When Divide curve is selected, the Segments value is used. The parametric space of curve is divided into a preset number of segments. This allows you to create a specific number of frames (Segments + 1), but sharp corners are smoothed.



Curve divided into 10 and 20 segments

Note: The preset options in the Curve Conversion Options panel are used for automatic conversion when you add a NURBS curve to the Path Library. The curve is converted to path and can be used for animation or with the Macro Sweep tool.

6.2 Non Linear Editor

6.2.1 Introduction to Non-Linear Animation (NLA)

NLA is the process of composing an animation by recording and blending independently-created local animations, called clips. Clips take the concept of working with keyframes a step further by treating groups of keyframes as abstract entities that can be easily manipulated within the Clip View of the Scene Editor. Overlapping clips can be blended together in relative or absolute modes, making complex animations much simpler to create than by working with keyframes alone.

For example, imagine that you have a character that you want to animate walking across a room. You could record each individual keyframe, posing the character and moving it in small steps until it reaches the other side of the room, as in traditional stop-motion animation. Then you may want to go back and add in a hand wave. You would have to place that hand wave exactly where you want it. This is a feasible but laborious way to create the animation.

On the other hand, you could create the same animation in a non-linear fashion by recording a single step (just the legs) as one clip, and the whole character's path of movement as a second clip. You compose the animation by repeating (looping) the "Step" clip and adjusting its relative scale to the "Path" clip so that the character walks smoothly across the room. Then you want to add in the hand wave. With a clip you can add it in at any time and move it anywhere, and have the waving motion naturally blend with the hand swing motion of walking.

NLA adds a greater level of flexibility to traditional keyframe animation. Even while blended, clips remain independent entities that can be copied, flipped, scaled, moved, and edited. The next section will discuss working with clips in more depth.

A note on the distinction between Key View and Clip View: "Key view" refers to the hierarchy and layer views. Both of these views allow direct manipulation of keyframes, and are therefore considered key views. In Clip View, objects only appear in the view if they have animation.

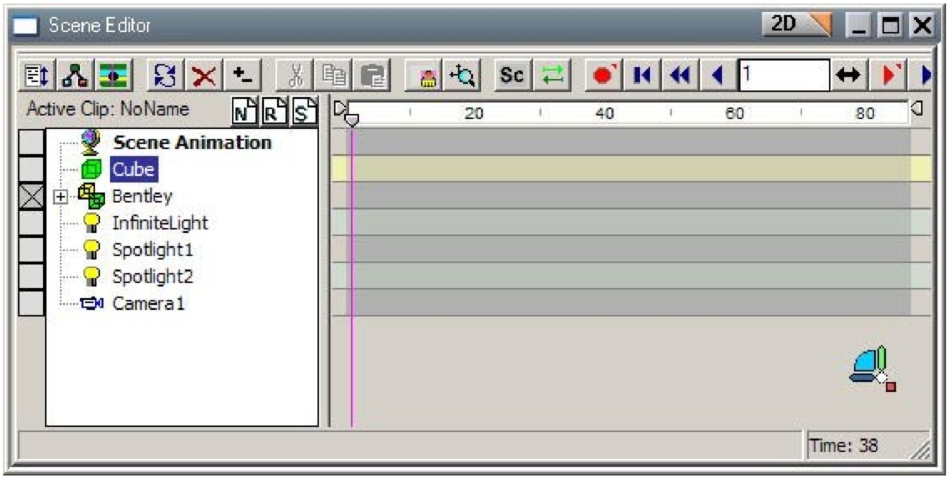
6.2.2 Working with Clips

By default, all animation you create will reside in the default clip. It is not necessary to create additional clips, but doing so allows for greater flexibility than working with all keyframes in a default single clip. Without creating additional clips, the Scene Editor will continue to work as it has in previous versions.

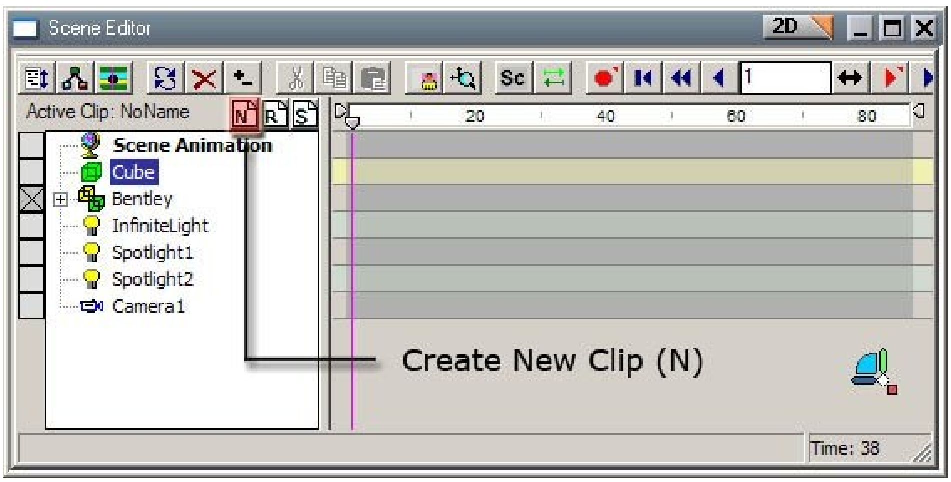
To view the clips, select Clip View from the Tree Mode popup.

Note: While in Clip View, the right pane of the Scene Editor may only be in clip view mode. Also, if Play Mode is set to Clip in the animation parameters panel (right-click Play), selecting a clip will change the play range to just that clip.

To create a new clip, select the object in the Object Listing that the new animation will be assigned to. Click the **Create New Clip (N)** button above the Object Listing, and enter a name for the new clip in the dialog that appears. After pressing **Okay** or ENTER, only the top-level object and all its children will be shown. For example, if you have the hand of a character selected, and you create a new clip, the SE will "crawl" back up the tree to find the top-level object (in this case, the group object that contains all the objects that make up the character), and will display only that object and its children.



Select the object that will be assigned the new clip



Click Create New Clip (N)

Clip View mode shows only top-level objects with one or more clips. There is a node under each object for each clip, and each clip will have a clip bar in the right pane indicating the span of the animation in that clip.

Grabbing the center of a clip and dragging will slide a clip, while grabbing the left or right blue/green tab and dragging will stretch or compress the clip. Double-click a clip to enter a special “clip-edit key-view” mode, in which the SE will display the keyframes for the current clip and filter out all other objects. Notice that the keyframes are displayed in “local” time, starting at frame 0, no matter where the clip is currently located on the timeline. Since clips are created and composed in a non-linear fashion, the actual time at which keyframes are recorded is not as important as the relative flow of the clip’s animation.

The image above shows several different attributes of clips:

- The “Slide” clip is selected (white).
- The “Spin” clip is not selected, but enabled (light gray).
- The “Sink” clip is not selected, and is disabled (darker gray).
- The sizing tabs on the “Spin” clip are green, indicating that the clip is relative. Blue tabs indicate an absolute clip.
- The “Spin” clip is looped, as indicated by the gray “ghost” clips to the right of it. Looped clips repeat forever. The tiny arrow in the upper right corner of the clip indicates that it toggles between playing forward and reverse while repeated.

The right-click menu for a clip contains four items specific to clips:

- **Enable:** Enable or disable the clip and its animation. This prevents the clip’s animation from playing or blending with other clips but does not delete the clip from the scene.
- **Relative:** When enabled, the relative animation of the clip, instead of its absolute parameters, will be blended with overlapping clips. For example, if one clip records an object’s forward movement, and a second clip records upward movement, the object’s forward movement would be shortened in the final animation because the absolute positions of both clips are blended. However, changing the upward motion clip to Relative causes the relative upward motion to be used in the blend, rather than its absolute positions, so in the resulting animation, the object would complete its full forward motion while still moving upward.
- **Toggle:** When enabled, the clip is repeated once, with the repeated portion being reversed. If

Loop is also enabled, then every other instance of the repeated clip is reversed.

- **Loop:** When enabled, the clip repeats for the entire length of the animation.

While not in clip view mode, the SE can be switched to show keyframes for the various clips that are available on objects. To switch the active clip, select an object and click the **Change** button above the object listing.

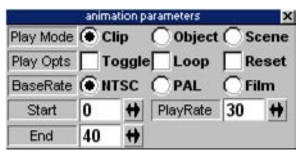
> **Note:** When trueSpace starts, the default clip is active. So if none of the objects in the scene have animation stored in the default clip, the SE will show no keyframes until you change to a clip that contains animation.

For example, if an object has two clips, Action1 (containing Move animation for Object1) and Action2 (containing Scale and Rotate animation for Object1), setting Action1 as the active clip will refresh the SE, and Object1 will have only a Move track displayed under it. Setting Action2 as the active clip will refresh the SE, and Object1 will have only a Scale and Rotate track under it.

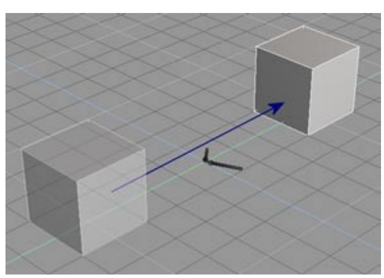
Absolute versus Relative Clips

To see the difference between absolute and relative clips, try this example.

1. Right-click the **Play** button to bring up the Animation Parameters panel. Set Play Mode to Clip.

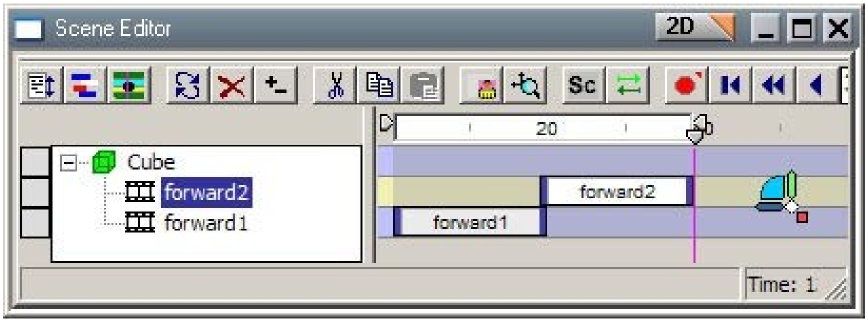


2. Place any object in the scene. In this example, we have placed a cube at X = 4, Y = 0, Z = 1. Make sure your object is highlighted in the Object Listing, and click the Create New Clip (N) button. Give the new clip any name. Advance to frame 20, and move the object a few spaces in any direction.

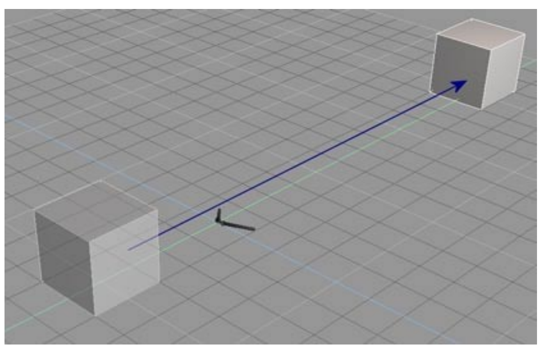


3. Now, repeat step 2, essentially creating a duplicate clip.

4. Set Play Mode in the Animation Parameters panel back to Scene. Switch to Clip View, and move the second clip so that it starts when the first clip ends, and play the animation. You will see the object move, return to its starting point, and move again, as if the animation were repeated once.



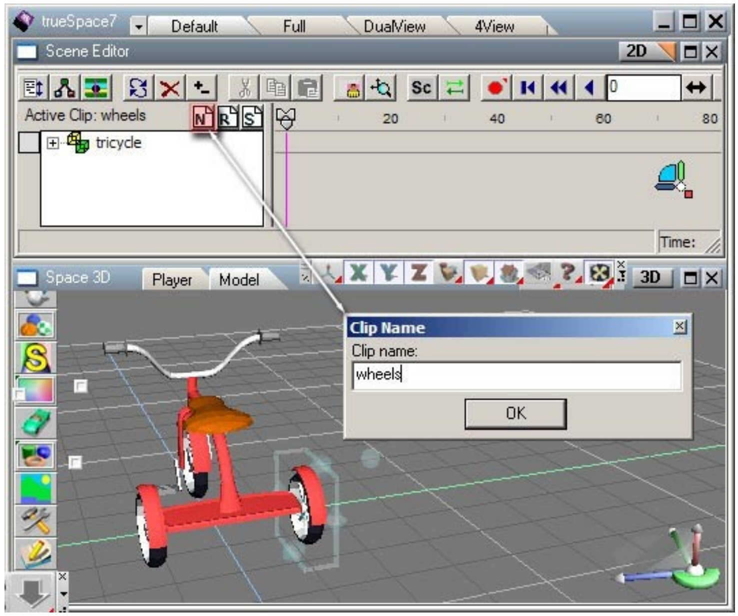
5. In Clip View mode in the SE, right-click the second clip and enable Relative. Play the animation again. This time, when the frame marker reaches the beginning of the second clip, the object continues its motion instead of jumping back to its point of origin. By placing the clip in Relative mode, its relative forward motion (in this case), rather than its absolute positioning, is used.



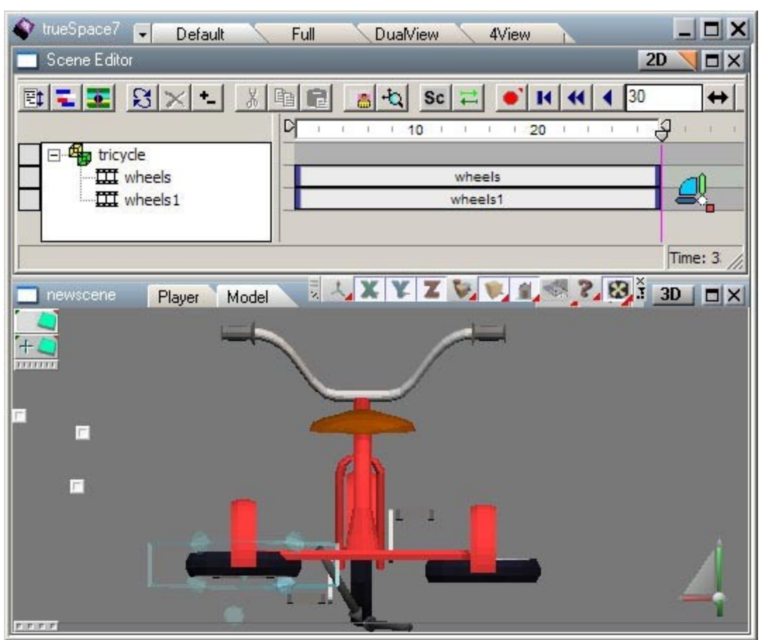
Clip Animation Tutorial 1: The Flying Tricycle

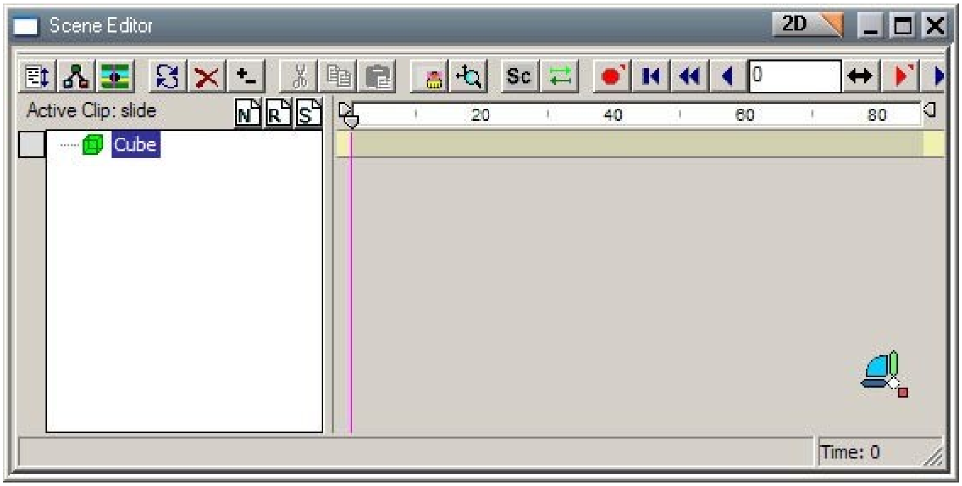
Note: This tutorial assumes you have AutoRecord enabled. To enable AutoRecord, right-click the **Record Key**, and enable AutoRecord. Otherwise, you will need to click the **Record Key** to create new keyframes.

1. Create a new scene, and load the Tricycle object from the "objects" library. Make sure the tricycle object is selected in the left pane of the SE, and click the New Clip button. Name this clip "wheels".



2. Select one of the rear wheels of the tricycle, move the time to frame 30, and rotate the wheel 90 degrees. Repeat this rotation for the other rear wheel.

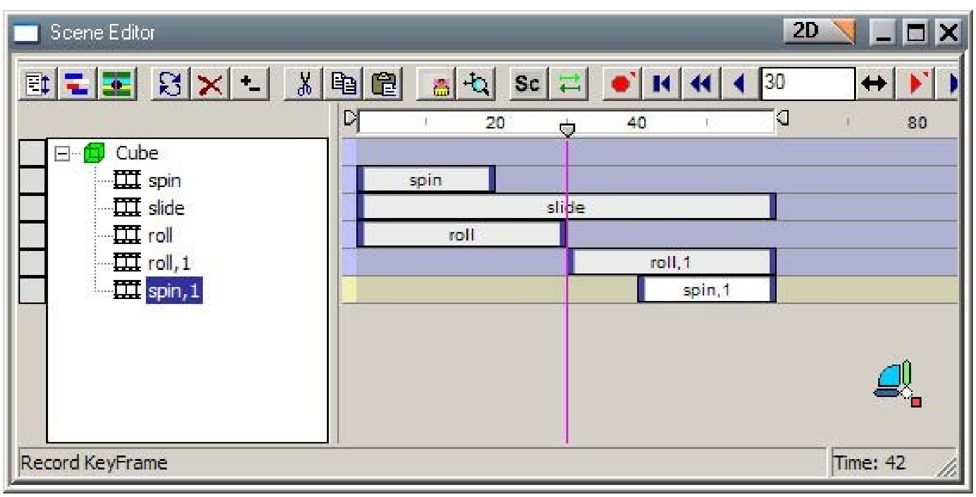




View changes to show only the current clip

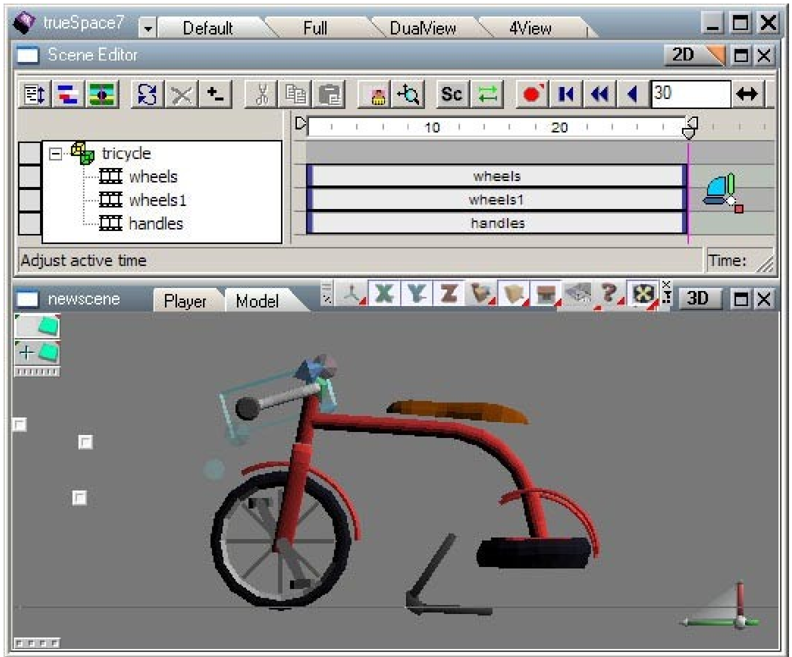
From this point, you may animate the object as desired. All keyframes captured will be stored in the active clip. You can switch to another clip by clicking the **Change Active Clip** (S) button and selecting a different clip from the list, rename the active clip using **Rename Active Clip** (R), or start another clip for the current object using **Create New Clip** (N). Changing the active clip will restore the SE to viewing all objects, and the animation you created will be stored in the new clip. This process may be repeated as many times as desired to create multiple clips.

When you want to change how the clips are played out and blended together in the animation, switch the SE to Clip View mode using the tree mode popup again.

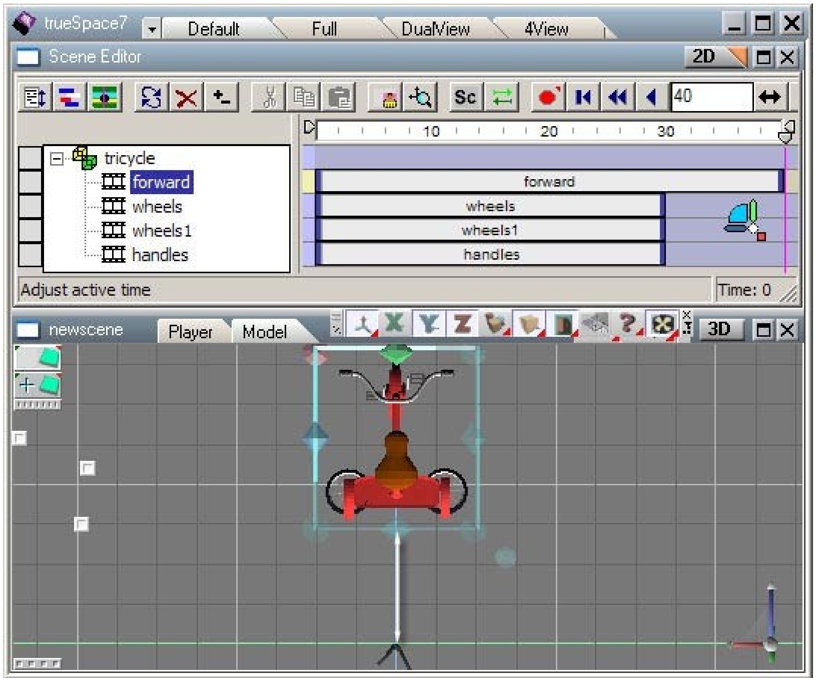


Clip View mode showing multiple clips

3. Select New Clip again (you do not have to save previous clip), and name it “handles”. Select the handlebars subobject, go to frame 30, and rotate handlebars forward.

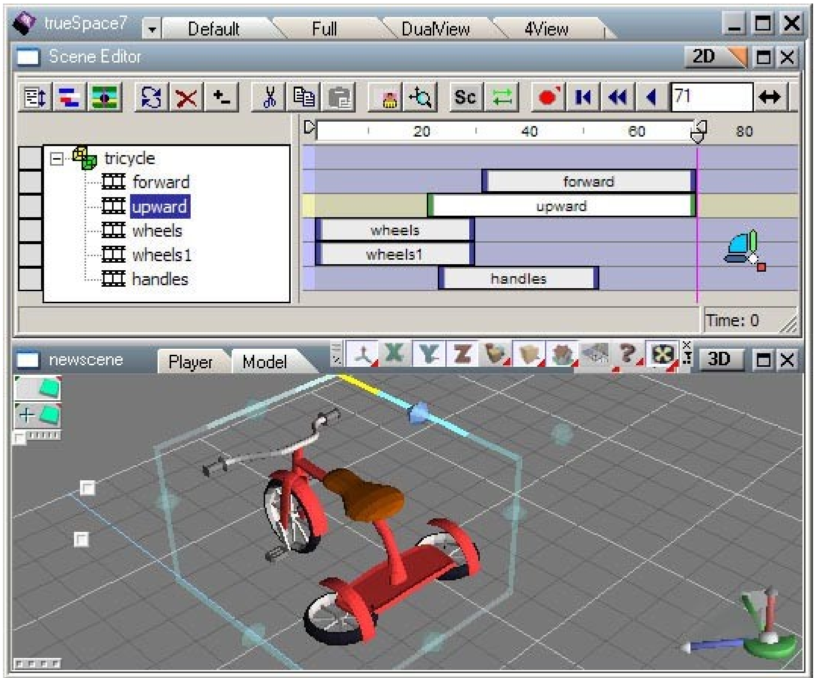


4. Select New Clip again, name it “forward”, and this time move the whole tricycle forward in 40 frames.



5. Select New Clip one last time, and name it "upward". Right-click the Play button, and select Clip mode. This will ensure that the "forward" clip we created earlier does not interfere with this new clip, which we will make relative instead of absolute. Move the whole tricycle vertically over 50 frames, and set the Mode in Animation Parameters back to Scene. You just created 4 clips for tricycle object.

6. Select the Clip View icon from the tree mode popup. You should see all four clips associated with tricycle in the right SE pane, all starting from frame 0. Drag the "handles" clip a little bit to the right so that it still overlaps with "wheels" clip. Drag the "forward" clip all the way to the right so that it starts after both other clips finished. Drag and resize "upward" so that its action takes place within the same time frame as the "forward" clip. Right-click the "upward" clip, and set it to Relative.



7. Click Play to watch the animation. Notice how the clips blend together to create the animation, and how the "upward" clip, despite being created with Move keyframes, does not add extra distance to the forward motion. Because this is a relative clip, the animation uses the relative upward movement of the tricycle, rather than its absolute position recorded in the clip.

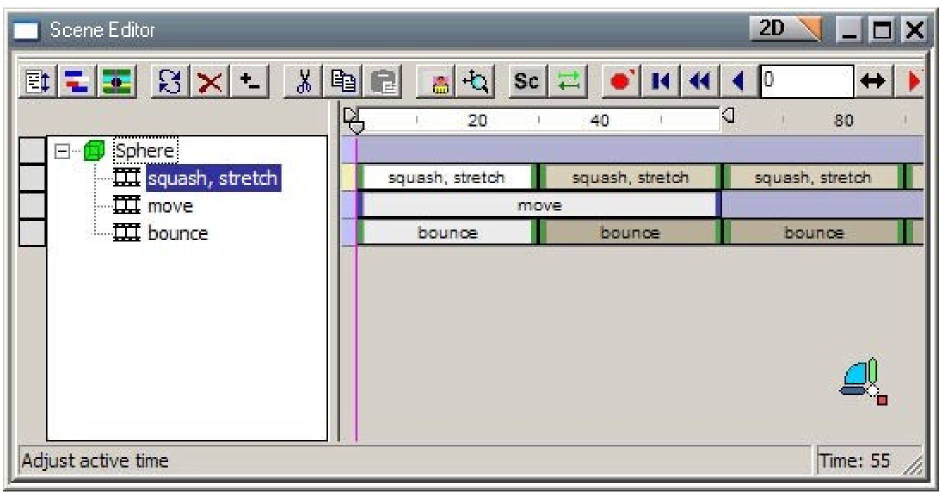
Clip Animation Tutorial 2: Squash and Stretch

This tutorial shows you how to use looped relative clips to create a "squash and stretch" animation of a bouncing ball. The same concept could be used for the walk cycle of a real character, where certain motions would be repeated throughout the cycle.

1. Create a new scene, and add a default-sized sphere primitive. Move it to 5,0,1.
2. In the Scene Editor, make sure your sphere is highlighted in the left pane, and click **Create New**

Clip (N). Call this new clip “move”. Advance to frame 60, and the move the ball to X = -5, Y = 0, Z = 1.

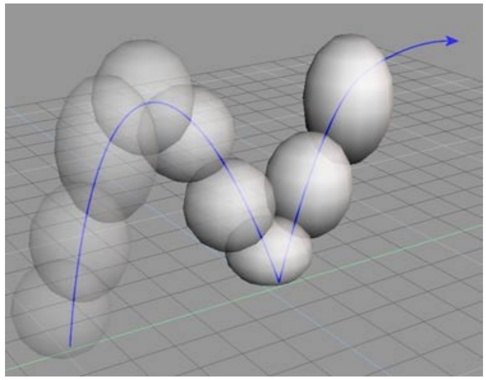
3. Create a new clip, and call it “bounce”. Right-click **Play** to bring up the animation parameters, and change the Mode to Clip. Advance to frame 15, and move the ball vertically 4 units. Advance to frame 30, and then move the ball back down 4 units to its starting position. Set Mode back to Scene in the animation parameters.
4. Using the tree mode popup, change to Clip View. If you play the animation now, you will notice that the Bounce clip has the effect of shortening the distance the ball travels in the Move clip because, by default, the clip animation is treated as absolute. Right-click the Bounce clip, and enable both Relative and Loop. Play the animation again, and you will see that the ball now bounces up and down while it travels the distance recorded in the Move clip.
5. The next step is to add some cartoon-like squash and stretch to the ball as it bounces. Add a new clip called “squash, stretch”. With the Object Scale tool active, record the following keyframes:
   a. At 12, change the ball’s size to X = 2, Y = 2, Z = 3
   b. At 16, change the ball’s size back to X = 2, Y = 2, Z = 2.
   c. At 27, click to record another keyframe (without changing parameters).
   d. At 30, change the ball’s size to X = 2, Y = 2, Z = 1.5
6. Return to Clip View, and set this clip to Relative and Loop. Your clips should now look like this:



When you play the animation, the ball should bounce across the grid, “stretching” at the peaks and “squashing” when it hits the grid.

Some of the ways in which you could further tweak this animation are by

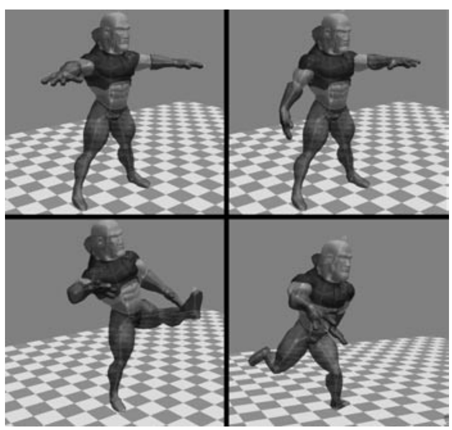
- Manipulating the keyframes and f-curves within each clip.
- Looping the “move” clip. The entire animation would repeat until it hits the end frame marker.
- Adding another clip to rotate the ball as it bounces.



6.3 Bones and Skinning

With Bones and Inverse Kinematics, trueSpace has a powerful set of tools to help you create complex animations, easily.

Bones in trueSpace function very much like bones in vertebrates, controlling the movement of surrounding skin and muscle. After you create an object in trueSpace, you can simply and quickly build it a skeleton and bring it to life. With bones you no longer need to build segmented characters and try to hide the seams. Now you can simply build the character how you like, and add a skeleton to get natural joints and muscle flex.



Not all situations however, call for smooth joints. If your project calls for a robot, crane, a knight, or any other rigidly jointed object, you still have the option of using IK to speed up your animation. You even have the option of combining rigid and seamless joints in one model.

As an added benefit, Bones are tightly integrated with trueSpace's Inverse Kinematics tools, the benefit of which is easier animation.

With traditional or Forward Dynamics you position each segment of a character individually. To move a character's arm to his face, you would need to move the shoulder, then the elbow, the wrist, and so on, right down to the individual finger joints. This is very time-intensive and arduous work. With Inverse Kinematics (or IK) you can pull the character's body into poses very much like you would an articulated puppet. For example you can take a character's finger to its face, and the rest of the arm will follow naturally.

This section will introduce you to all the tools related to Bones and IK. By the time you finish this section you will know how to create skeletons for smooth jointed characters, assemble rigid jointed IK models, and animate them both. The focus of this section is on creating skeletons for animating characters, while the Inverse Kinematics section focuses on general joint connections such as hinges on a door, etc.

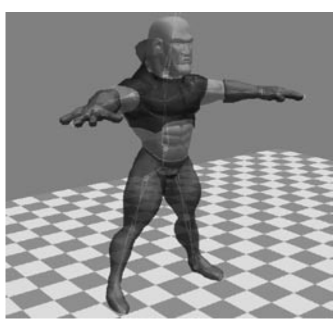
6.3.1 Terminology

Most of the terms used in reference to the Bones and IK system in trueSpace correspond to real world anatomy terms. This makes it easy to quickly learn their functions.

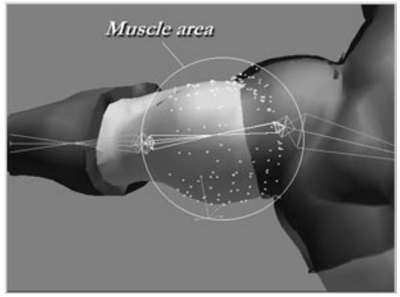
Bone - The most basic part of a skeleton, bones control the deformation of surrounding skin.

Joint - Joints connect bones together and control how they move in relation to each other.

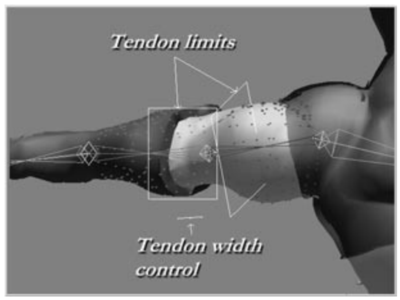
Skin - An object that is deformed by a skeleton. Any polyhedron object you create in trueSpace can be used as skin. Some types of objects, such as NURBS and Plastiform will need to be converted to polyhedron objects before they can be used. trueSpace will prompt you when this is the case.



Muscle - Flexing in trueSpace is accomplished with muscles. Muscles are controlled through a dialogue box, and have no dedicated visual representation.

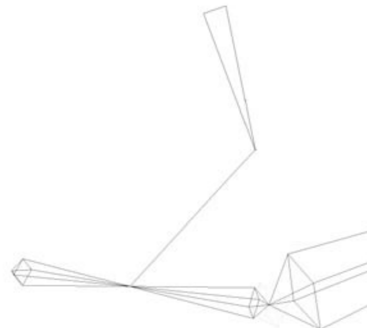


Tendon - The area around a joint is the tendon. Tendons control how skin bends and moves.

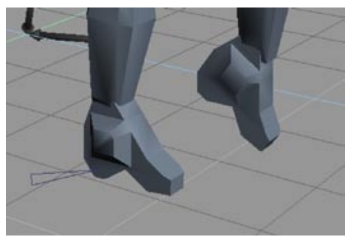


Skeleton - A skeleton is simply an object made up of bones and joints.

Nail - The nail is the little blue triangle you see hanging onto each unique skeleton in a scene. It controls which part of a skeleton will remain stationary when using Inverse Kinematics. The nail is created automatically whenever you create a skeleton. To move a skeleton use the move tool and either drag the bone with the nail, or drag completely off the object.



Note: It is possible to create a complex skeleton object consisting of more than one IK branch. These types of complex objects have nails for each branch. A nail cannot be moved outside of its branch. If you attempt to do this, the nail automatically returns to its former position.



A robot's foot nailed in position

6.3.2 Manipulating IK and Bones Objects

This will be covered in detail later in the chapter, but this should serve as an overview of the process:

To manipulate an IK or bones skeleton, the entire object should be selected (i.e., you should be at the top of the hierarchy), and Object Move mode should be active. Click on the end of the chain of component objects you wish to manipulate, and drag the mouse. One way that IK manipulation differs from normal object navigation is that you can manipulate any component object of an IK object without first selecting it. You can grab any part of an IK object and move it directly.

You can move a selected IK object with or without Inverse Kinematics. If, when you drag the mouse, the mouse pointer is outside the object or it is on a nailed component object, the whole object moves according to how the mouse moves. If the mouse pointer is on another component object of the IK object (called an effector), Inverse Kinematics controls the motion of all component objects between the object and the nailed object. A chain of component objects from the effector to the nail behaves according to the degrees of freedom set for joints along this chain and according to how you move the mouse. The mouse motion is translated into a directional force, which is depicted as a yellow line emanating from the effector at the point on which you click. If the Use Torques option is active (see later in this chapter), the mouse motion is also translated into a rotational force about the center of the effector. Thus, clicking on different parts of the effector causes varying object behavior.

Branches of component objects that are connected to some other component object in the chain from the nail to the effector are moved and rotated with the component object they are linked to. Any part of the object “above” the effector is moved and rotated with the effector.

You should now have a good understanding of the basics behind IK and bones. We will now move into the core of the subject - the tools.

6.3.3 The Tools

Build Skeleton

The **Build Skeleton** tool does just what it says it does. Using this tool you will create the skeleton that you will use to manipulate your object.

Attach Skin To Skeleton

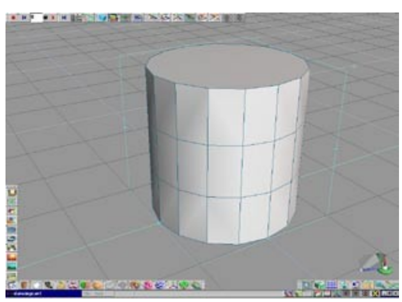
The **Attach Skin To Skeleton** tool also does just that. Once you have created the skeleton, you will use this tool to attach it to its skin, or vice-versa.

Let’s start by building a very simple skeleton and doing a quick animation with it.

Tutorial: Dancing Soda Can

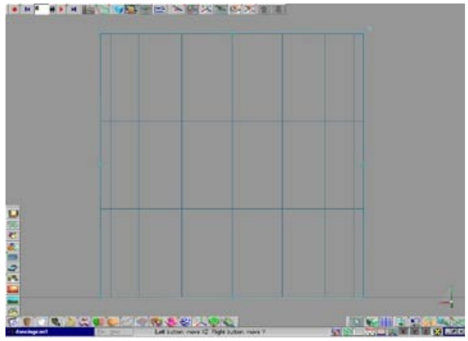
You may have seen tutorials on creating a dancing can in any number of other places, and for good reason. It is a very simple way to introduce the basic concepts of bones animation.

Step 1: First you need to create your can. To do this, just create a cylinder with latitude of 3, and a longitude of 16. (See **Artist Guide Chapter 3: Modeling** if you are not sure how to do this.)



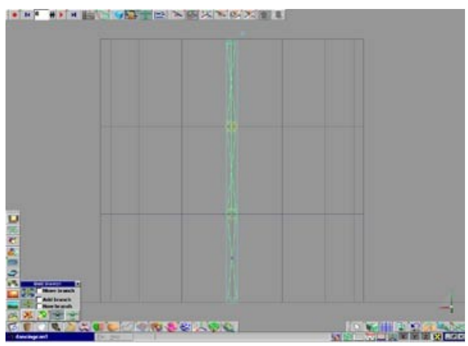
Step 2: Now we will build a very simple skeleton for our can. Go to a side view and switch to wire-frame. Select the **Build Skeleton**  tool; note that the Build Skeleton flyout appears.

Click in the middle of the can, on the third row of vertices from the top. Your scene should look something like this now.



Note that trueSpace creates a joint where you clicked and a bone going off in either direction. trueSpace will try to center the skeleton inside the skin as you build. If you now switch to front view, you will see that it did just that.

Switch back to side view and click again on the second row of vertices from the top. Note how trueSpace keeps the bones from protruding out of the skin.



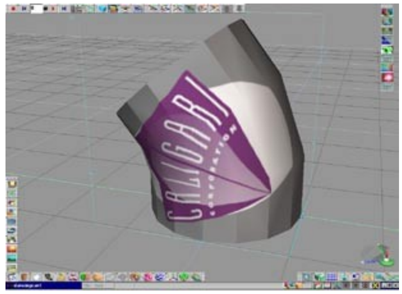
You have completed the skeleton for the can; now all you need to do is attach it.

Step 3: With your skeleton still selected, select the **Attach Skin To Skeleton**  tool. The familiar glue bottle cursor will pop up. Click on your can to attach it to the skeleton. You are now ready to start animating your dancing can.

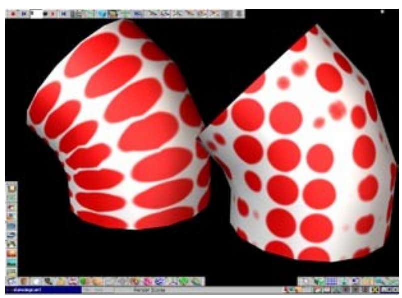
Step 4: Select the object tool and switch to a 3D view. To make the can move, click and drag the top most bone. You can see already how easy to use and powerful the Bone and IK tools can be. To animate your can, go to another frame and move the can in the same way you did before. Click play, and watch your can move.

You could also use the DoF tool to move and animate the joints. More on that later.

Step 5: Let’s paint our can. In the texture directory is an image named caligari_can.jpg. Use it as a texture map, make the material shiny, and apply it to the can. Run your animation again in Solid Render mode. As you can see, the texture moves with the can, as you would expect.



Note: Boned objects work very well with texture-maps and wrapped procedurals that make use of UV-space; non-wrapped procedurals, however, work poorly with animated boned objects. In this image, the object on the left was painted with ‘Wrapped Polka,’ and the object on the right was painted with ‘Solid Polka.’ Clearly the ‘Wrapped’ procedural is most often the desirable choice. This goes for all vertex animation, bones or otherwise.

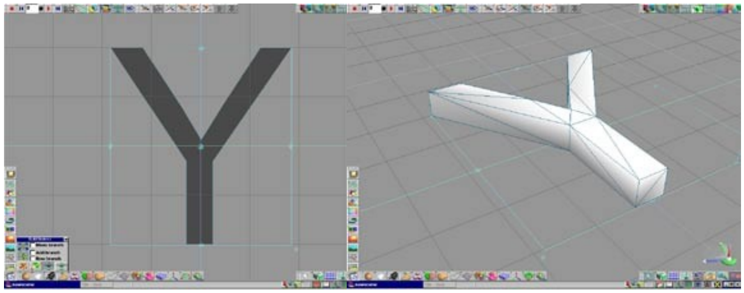


Tutorial: Using Branches

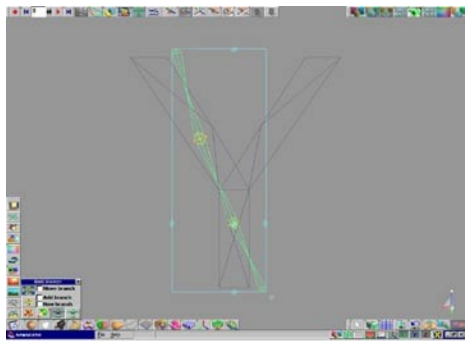
So far you have learned to create simple skeletons, great for animating cans, snakes, etc. Very quickly however, you will need to build branched skeletons. A branch is simply a bone that connects to more than one joint. A simple example is a pelvis. This one bone connects the base of the spine to the top of each leg.

For this tutorial you will build a very simple branched skeleton, and use it to animate a capital letter ‘Y.’

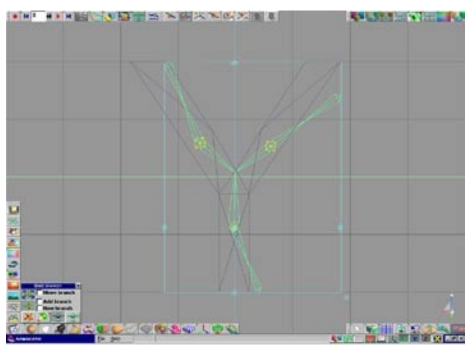
Step 1: Open a top view, and, using the horizontal text tool, type a capitol ‘Y.’ Switch to a 3D view and sweep once. Lastly, triangulate the object.



Step 2: From a top view, select the**Build Skeleton** tool. Place a joint in the main body of the 'Y,' and another in the left hand branch.



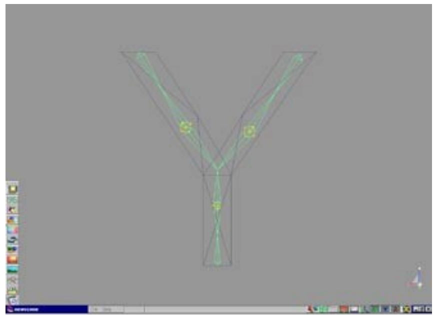
Step 3: To add the bone for the right hand branch, hold the CTRL key and click in the right hand portion of the letter.



By holding the CTRL key you told trueSpace to create a branch, as opposed to appending another bone to the end of the chain. Alternately, you could have selected the **Add Branch** and accomplished the same thing. The branch will come off of the bone you click closest to. As such, you may need to click close to the bone you want to branch off of, and then, without releasing the mouse, drag to where you want the joint to be.

Step 4: You could attach the skin at this point, but the joints are not yet where they should be. Select the **Move Joint** tool. You can now move bones and joints around easily. Simply click and drag parts of the skeleton until they are where you want them. If the Bounding Box Widget gets in the way, press TAB to temporarily disable it. TAB will also bring it back.

Starting with the joints, manipulate your skeleton to look more like the one below.



Once the joints are in the proper place, you can move the bones into the proper position. Its important to move the joints first, as moving a joint will affect the position of the bone.

> **Note:** Grab the bones at the end otherwise they will bend. This can be advantageous in some circumstances, not here though. That will be covered later.

Step 5: Using the **Attach Skin To Skeleton**  tool, attach the letter.

Note: A single bone can branch into any number of joints. This is useful to keep in mind when creating hands, feet, spiders, etc.

Move Branch

The **Move Branch** tool allows you to move entire segments of a skeleton at once. When this tool is on and you select a joint to move, all joints farther from the nail than the selected joint will also move.

New Branch

The **New Branch** toggle allows you to begin building another skeleton easily. When you select this toggle you are telling trueSpace that you no longer want to add bones to the current skeleton, rather you want to start a new one.

Delete Joint

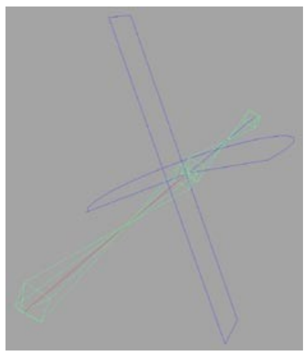
The **Delete Joint** tool does exactly what the name implies. To use, just depress the Delete Joint button, then click the joint you want removed. This will not break your skeleton apart, it will just remove the offending joint.

Edit Joints Directly

Each joint can rotate in any or all of the three directions, yaw, pitch and roll. Each joint can also slide in any three axes, X, Y, or Z. You can also set stiffness levels for each joint, making it easier for example to move a finger than to move an elbow.

When you select the **Edit Joint Directly** tool and select a joint you are presented with a graphical representation of that joint's range of movement, direction of rotation and/or translation (slide), stiffness, orientation, and value.

If it proves too troublesome to edit a given joint to be how you want it, you have the option to flip the joint. To do this, double click on a joint. If a given joint is not symmetrical, a flipped joint will often prove to be the better choice.



With the **Edit Joints Directly** tool you can manipulate these joints by dragging the different graphic elements the arch represents the given joint's range of motion; the bars represent the range of translation. The arches and bars are colored to make them easier to understand. Blue is for pitch rotation and X axis translation, red is for yaw rotation and Y translation, and green is for roll rotation and Z translation.

The distance between the spoke of the arch or length of the bar graphically shows the range of movement that a given joint is allowed to make. A joint will only pivot or slide within the constraints of the bars or arches. To change the range of motion, simply drag the edges of the bars or the spokes of the arch to the desired setting.

The diameter of the arches, and the height of the bars represent stiffness. The default is 0, or no stiffness. The bigger or taller the arches and boxes become, the more stiffness they have. When using IK, a joint with no stiffness will bend less easily then a joint with more.

Lastly, the simple lines projecting out from the center of the joints show the current value. Left drag this line to manipulate a skeleton while building it.

> **Note:** You can also right drag this line to change the orientation of the joint. If, for

example, none of the axes of rotation align with the desired rotation you want, you can correct it this way. There is no numerical way to change this value. For example, use this feature to get a knee to bend the correct way.

For more control you can also edit the joints numerically by right-clicking the **Edit Joints Directly** icon to access the **Joint Attributes panel**.

This panel presents you with numerical information for the selected joint. This is also where you would enable or disable any or all of the different axes of translation and rotation. Here you can also disable the min/max rotation and translation values, thus allowing unlimited rotation. To create a joint with no movement at all, simply disable all the axis of rotation and translation.

Build Skeleton Properties Panel

You can access the Build Skeleton Properties panel by right-clicking the build skeleton tool. This panel allows you to tweak the way the **Build Skeleton** tool works and the color of the components.

Link With

This tool allows you to pick one of six types of starting joints that will be used when you build the skel - eton. The default is a 1D hinge, or spherical joint. The other options are, 0D Fixed Joint, 2D slide joint, 1D Slide Joint, 2D Spherical Joint, and Shaft Joint, a 1D slide joint and a 1D spherical joint combined. These six are available as presets, but there a lot of other types of joints that you can create using the **Edit Joints Directly** tool and the **Joint Attributes panel** after you have created the joint.

Build Vertically

This option works just the way you would think. When it is selected you can build vertically instead of horizontally, or on the ground plane. This is useful when creating skeletons without the final skin in the scene.

Hide Leaf Bones

Leaf bones are the bones that only have one joint associated with them and are found at the end of branches. Occasionally while building a skeleton they can occlude your view. When this occurs simply check this box and they will be made invisible.

Attach Skin

This option will cause trueSpace to associate the skin that you click with the skeleton you are building as soon as you begin to build it. This is useful when building simple objects that you want to finish quickly.

Bent bones and 0D fixed joints

Although it is entirely possible to tweak the alignment of joints, and use different angles in one or more axes to get a desired bend from a joint, it is rarely the best answer. trueSpace takes into account the direction from which each bone initially connects to the joint to determine how to align the axis of that joint.

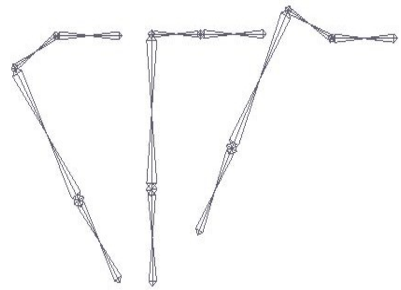
Sometimes this will be the way you had in mind, often however it will not. To correct this you have several options.

The first is a bent bone. A perfect use for a bent bone is the upper arm. Think of your own humerus, or upper arm. From the elbow up it is mostly straight, but just before connecting to your scapula it takes a turn toward your body, effectively connecting at the side of the bone.

To emulate this in trueSpace, build your skeleton as you like, and position the joints. Then, using the **Move Joint**  tool, click and drag the center of the bone to where you want it. You should do this before tweaking joint attributes to make things easier. Bending a joint can radically affect the way a bone will move.

Alternately, you could use a 0D fixed joint. This is simply a joint with all of its translation and rotation disabled. In some instances you will not be able to, or even want to, use a bent joint. In the arm bone example you could put a joint in the place where you moved the center of the bone to, then make it 0D fixed by opening the Joint Attributes panel and disabling all the translation and rotation.

Moving Joints



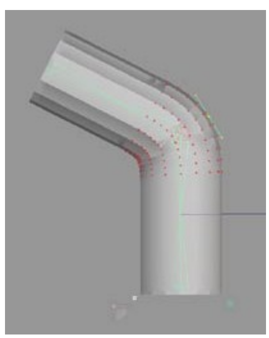
Using 0D Fixed Joints

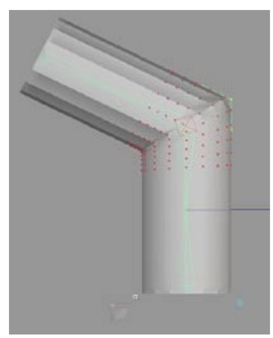
6.3.4 Edit Muscles

Edit Muscles tool

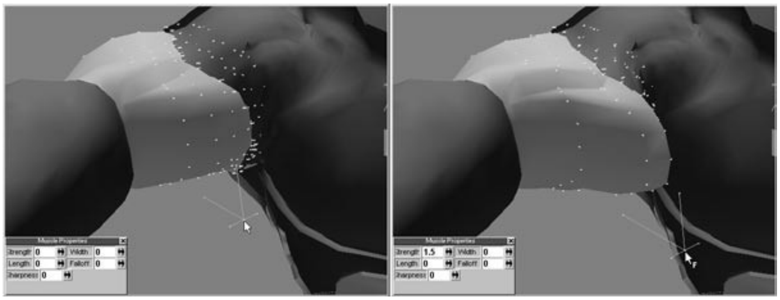
The **Edit Muscles** tool is extremely useful for many different tasks. Primarily it is used for editing the way muscles flex and move, but it also can be used to control how the skin reacts to the movement of the skeleton.

One function of this tool is to set the sharpness of the muscle tendons. Selecting a joint will highlight its vertices in red and provide control handles allowing you to change the sharpness:

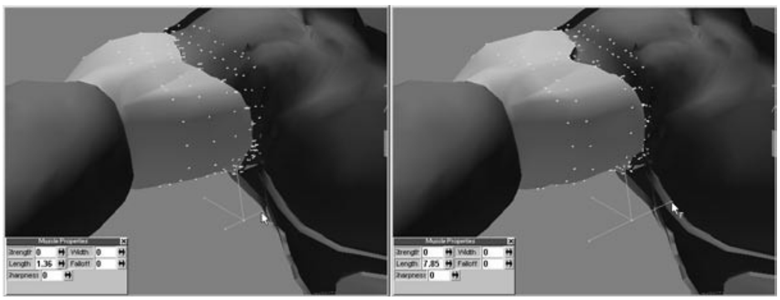




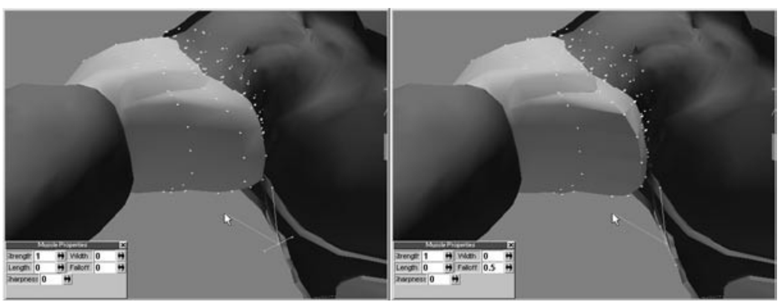
The **Edit Muscles** tool also allows you to set the strength, length, falloff, and width of muscles. Selecting a muscle will highlight its vertices in dark blue and provide control handles for manipulation of these settings:



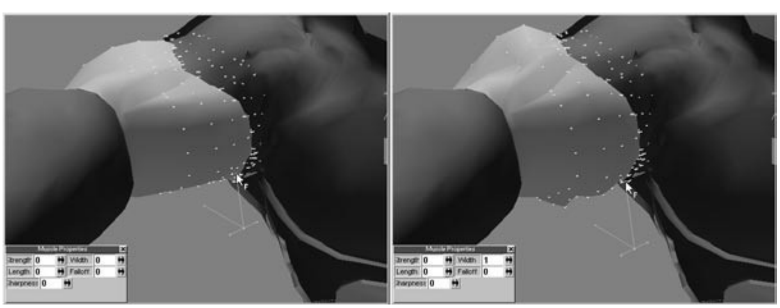
Editing Strength



Editing Length



Editing Falloff



Editing Width

Add Vertices to active Muscle or Tendon

When you associate a skin with a skeleton, trueSpace automatically associates all the vertices of the skin with the nearest bone or joint. This is not always the desired result. With the Add Vertices to active Muscle or Tendon tool you can use any of the standard point selection tools to re-associate vertices to your liking. The points you select will then react the way you want them.

Note: There is no need to remove a point from one joint or bone before adding it to another. The **Add Vertices to active Muscle or Tendon**tool overrides previous associations.

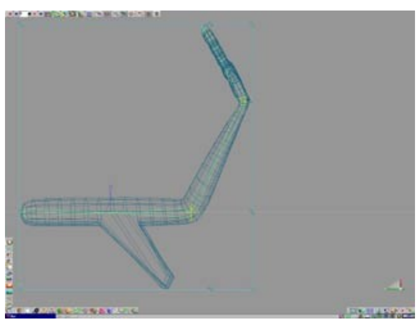
Remove Vertices from active Muscle or Tendon

This tool does the exact opposite of the **Add Vertices to active Muscle or Tendon** tool, leaving points associated with the nearest joint or bone.

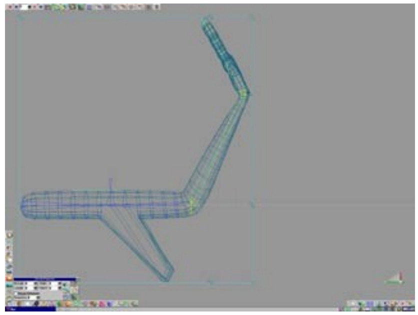
Tutorial: Flexing Some Muscle

Using the **Edit Muscles** tool and panel we are going to fix a miss-associated set of vertices, and add some strength to a simple arm without going to the gym.

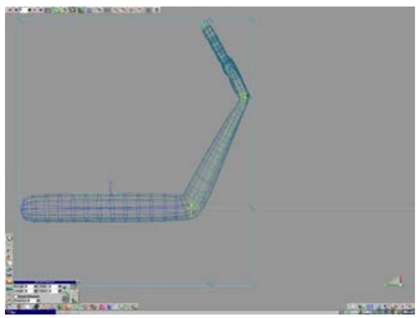
Step 1: Load flex.scn from the scenes directory, and spend a moment moving the hand around. Very quickly you will notice two things are wrong. First, there is no muscle flex anywhere, which is not very impressive. Second, there is a set of vertices in the upper arm that are moving in tandem with the forearm causing serious tears in the skin. Before we can add any flex to the muscles, we need to be sure they are where we want them to be. Bend the arm so that the elbow is at no less then a 45-degree angle.



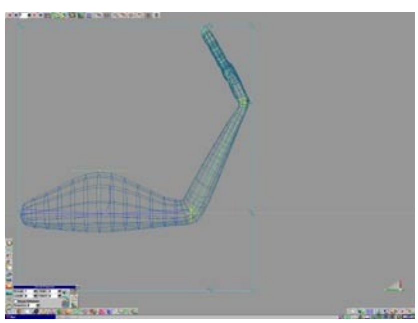
Step 2: Select the **Edit Muscles** tool; the Muscle Properties fly-out opens up automatically. Select any joint or bone and all the associated vertices will highlight in the color that represents what selected object is. This color depends on how you have them set in the Skinning panel. Select the upper arm bone.



Step 3: The set of points that are protruding from the bottom of the arm should be associated with the upper arm bone, which you can see with that bone selected that they are not. This is easy to fix. Using any of the selection tools simple pick the points you want re-associated. They will be added to the selected bone as muscle. If you were working with a joint, they would be associated as a tendon, more on that later.

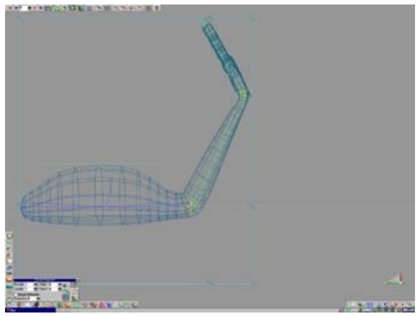


Step 4: With all the points properly associated you can now add a little flex. There are of course two ways to do this: graphically and numerically. With the upper arm bone still selected, and the arm still bent, change the Strength setting from 0 to 1. As you change the value you can instantly see the results; the muscle becomes apparent. If you had done this with the arm straight you would not have been able to see the flex because of course the muscle would be relaxed.

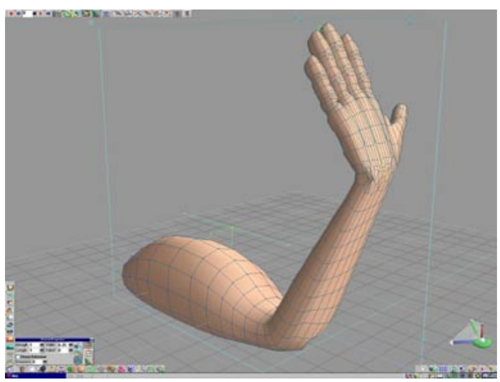


Note: Don't be afraid to move your model's skeleton around while setting the muscles. You can always restore the original pose by selecting the build skeleton tool.

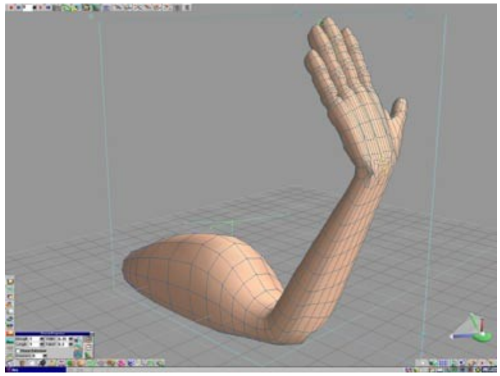
Step 5: Now we have some muscle, but it is not flexing realistically. Real muscle expands in more than one direction while flexing, and in different ways along their length. With the same muscle still selected, change the Length to 1. You can see that the muscle expands along its length to appear a little more natural. 1 may be too large or small so feel free to change any of these settings to your taste.



Step 6: You are starting to see a more realistic muscle forming, but still there's something missing: flex along the width. Change the Width setting to 0.35. This should be starting to look really good. There are just two more settings to tweak.



Step 7: Muscles bulge and flex differently depending on a great number of variables; it's up to you to determine where the majority of the bulge should be. Use the Falloff setting to do this. Changing this setting will affect whether the bulk of the flex will happen on one half the arm or the other. Change the Falloff setting to 0.3, and then change it to -0.3. Note how the muscle reacts, and pick a setting you like.



Step 8: You have now tweaked all the settings that relate to Muscles. The last setting is related to Joints and is the only thing the Edit Muscle tool can do to them. Select the elbow joint. Now tweak the Sharpness setting and note how the elbow becomes pointier.

Keep in mind that a bone can either be a contractor or muscle. Only muscles will flex and bulge. You can control which bones are which with the Skinning panel, discussed later.

All/Bi-directional Contraction

A muscle can have one of two type of contractions, which can be selected from the pop-up menu.

By default muscles are set to All-directional contraction, which means the muscle can bend in any direction and all the muscle will behave the same.

The bi-directional setting allows you to apply two different sets of muscle settings to one bone. Think of your upper arm. You have a bicep in the front, and a triceps in the back. They behave independently and usually the triceps are less developed than the bicep. All bones in trueSpace can make use of the contraction setting except leaf bones.

When you turn on Bi-directional contraction, a set of lines will appear from the center of the bone, one red, and one blue. They can each have their own settings. The lines denote the orientation of the muscles and can be set in any way. trueSpace assumes you want to edit the muscle that would be flexing, which is why it is important to have the joint bent when you edit.

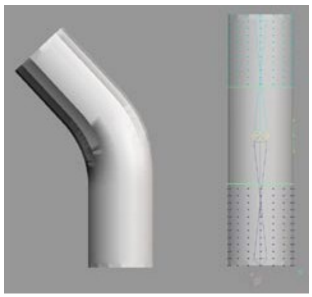
After turning on bi-directional contraction, change the settings for the first muscle. You then have two options for editing the second muscle. You can leave muscle-editing mode and bend the muscle the other way. Alternately, you can click and drag the red or blue line around to set the orientation of the muscle in the other way, edit the second muscle, and then reset the orientation.

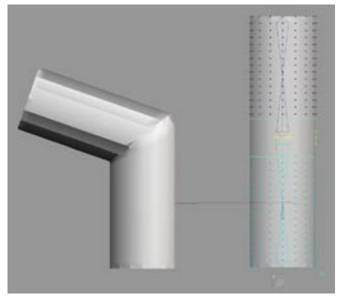
The muscle will flex differently now depending on the direction the joint bends.

6.3.5 Edit Tendons

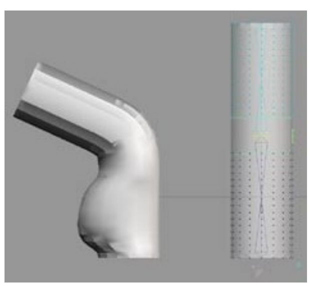
Edit Tendons tool

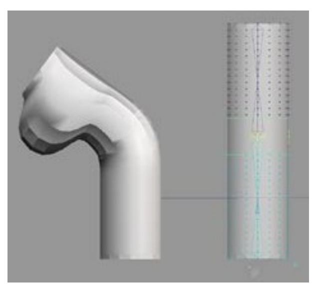
Whereas the points associated with a Bone act as Muscles, the points associated with a Joint are referred to as Tendons. The **Edit Tendons** tool simply affects the ratio of Muscle to Tendon. By dragging the green handles, more points will be associated with a given Joint, and the more the skin will bend. This is referred to as the "rate" of the tendon, and can also be edited in the Skinning Panel, which will be detailed shortly.





Note: While it will be necessary to bend a joint to see the effects of a change made to a tendon, you can only change tendon settings to a skeleton in its original pose. The **Edit Tendon** tool will automatically return the skeleton to that position, and the skeleton will return to its previous pose when you are finished editing the tendons.





Another feature of this tool is that it gives you a way to specify which of the two bones associated with a tendon is the Contractor - the muscle that will flex. While you are in Edit Tendon mode, simply click either of the attached bones. The contractor vertices are displayed in dark blue.

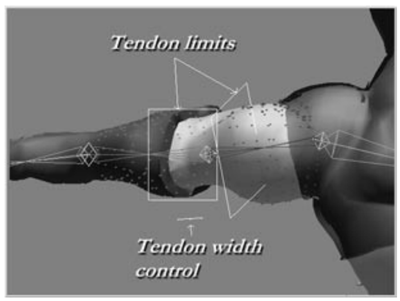
One important thing to remember about the **Edit Tendon** tool is that it will erase any manual changes that you have made with the **Edit Muscle** tool. To avoid this, always edit your tendons before editing muscles.

Skinning panel

Right-clicking in the **Edit Tendon** or **Attach Skin to Skeleton** tool accesses the Skinning panel. This panel controls how skin is associated with a skeleton at first association, and how that skin reacts after it is attached.

Tendon Rate

This is the numeric adjustment for the **Edit Tendon** tool. Higher values will increase the influence of the tendon and the number of points associated with the joint.



Beyond last Bone

The default for this setting is on, which in almost all situations is how it should stay. When this is on portions of skin that protrude past a leaf bone are associated with the bone, when it is off, these points are simply not associated with any bone.

Before Nail

When this option is selected, any portion of skin that does not envelope a bone is associated with the bone that has the nail.

Contractor

Specifies which of the two bones associated with the selected joint is the Contractor, or the bone whose muscle will be able to flex. By default the check box to the left is selected, this represents the bone closer to the nail. If you want the other bone to be the Contractor, just check the other box.

Realtime Update

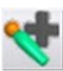
Some slower machines will bog down when editing the Tendons of models with a great many points, so the option of disabling the **Realtime Update** has been added. When this box is unchecked, only the bounding boxes for the joint will move; the re-association of points will take place after the mouse is released.

Tendon, Muscle, and Contractor Color

Like the other color selectors found throughout the trueSpace interface, these control the color of their associated object.

Link IK Object

The Link IK Object tool only works when you click on or in a group of objects that have been glued together. trueSpace will automatically attach the object you click to one bone, and that object's nearest neighbor to another bone. This tool is an excellent way to build rigidly jointed objects that utilize trueSpace's IK.



Add New Bone

The **Add New Bone** tool places a bone with no joints into the scene. Move the bone to where you want it, and continue to do this until you have all the bones where you want them. Use the Add Joint tool to connect them together.

Add Joint

The **Add Joint** tool is another way to build rigidly jointed models. Unlike the Link IK Object tool, the **Add Custom Joint** tool can be used on multiple objects whether or not they have been glued together. An added benefit of this tool is the ability to choose a type of joint before you build it by simply choosing the type you want from the fly-out. You have the same options for joints that you have in the Build Skeleton Properties panel.

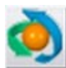
To use the **Add Joint** tool, choose your first object, then select the **Add Joint** tool of the desired type of joint. Next, navigate down the resulting hierarchy to the object you want the next object to be attached to. Repeat these steps until you have completed your object.

Add 0D fixed joint

Add 1D slide joint

Add 1D hinge joint

Add 2D slide joint



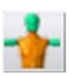
Add 2D spherical joint

Add shaft joint (1D rotation, 1D translation)

Extract Skeleton

This tool is very useful for transferring or recycling skeletons. Simply select any skinned or rigidly jointed object, then select this tool. A copy of the skeleton will be created with all objects and skin removed. You can then take the skeleton and apply it to another object.

This tool is also very useful for transferring animations, since the resulting skeleton maintains its animation.



Attach Object To Bone

The **Attach Object To Bone** tool is another way to build rigidly-jointed models. If you build a skeleton using any of the available techniques, you can then use this tool to attach an object to the skeleton that will not be deformed.

Note: This object has several other very important uses. It is with this tool that you can create mixed models; that is, a model that contains both rigid objects, and deformable skin. Imagine a human head, for example. The lips could be deformed using skin, while the teeth and gums are rigid.

Also, objects that are attached to bones can use the **Look At** feature. This is very useful for a character's eyes. Attach them to the bone, then assign them to look at a target, and you now have very fine control over them.

To use the **Attach Object To Bone** tool, navigate through the hierarchy to the bone you want your object associated with (you can use the cursor keys for this), then select the **Attach Object To Bone** tool. The glue bottle will then appear. Click the object you want attached. You can still move the object to be attached after clicking it if it is not exactly where you want it since the object is not attached until you release the mouse button.

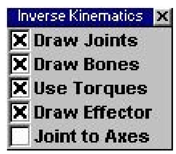
Alternately you can use this tool with the entire skeleton selected and trueSpace will attach the object to the nearest bone.

Note: You can attach as many objects to a bone as you like, for example a set of teeth to a bone that functions as a jaw. To access individual attached objects you need to use the arrow keys or the Scene Editor.

To remove an object from a bone you can also use the **Attach Object to Bone** tool. With the tool selected you can either hold CTRL and click the object to be removed, or hold CTRL and drag the object to remove it.

Inverse Kinematics panel

Right-clicking the **Attach Object To Bone tool** accesses the Inverse Kinematics panel.



Draw Joints

This toggle determines whether or not joints are displayed.

Draw Bones

This toggle determines whether or not bones are displayed. This is useful when editing joints that are occluded by a complex bone structure.

Use Torques

When this toggle is active, trueSpace calculates rotational forces as well as directional movement from your mouse input. This causes your skeleton to turn and rotate more realistically.

Draw Effectors

The effector is the yellow line that appears when you manipulate a skeleton, visually representing the forces you are exerting. This toggle enables and disables the line.

Joint to Axes

This toggle affects the **Add Custom Joint** tool. When it is unchecked, the joints created will be in the middle of the two objects. With this option selected, the joint will appear at the axis of the second object, thus maintaining its center of rotation.

Change DoF Values

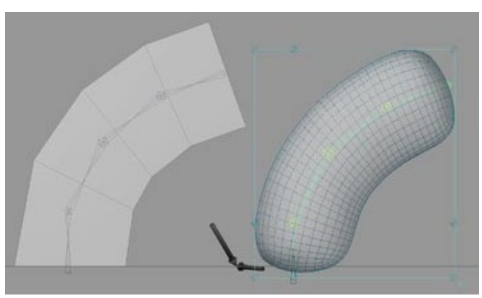
The **Change DoF Value of Joints** tool allows you to select an individual joint and manipulate any of its six degrees of movement independently of the other degrees, and of all the other joints. This tool is perfect for making those tweaks you just cannot seem to get done using IK.

To use, simply select the tool, then the joint you want to tweak. The familiar graphic representation of joint ranges appears. Now, however, you are not using it to edit joint properties; rather, you are using it to adjust the joints movement with in the constraints you had previously set. Resultant changes will not affect the properties of the joint in any way.

6.3.6 Using Surface Subdivision with Bones and IK

Surface subdivision is an extremely powerful tool. This is especially true when you integrate SS with Bones and IK. To combine these great features, first complete your SS control mesh and get the subdivision settings how you like them. Then simply associate your skeleton as you normally would. You now have far fewer points to associate with bones and joints. Because only the control mesh is actually associated with the skeleton you need only deal with those vertices; all the vertices of the subdivided object will then tend to themselves.

You also have the benefit of more realistic stretching in some situations. Where trueSpace bones are very good at averaging movement around joints and at muscles, SS takes a step further by providing averaging of points in the control mesh when creating the resultant mesh. You also can animate using the low-resolution control mesh saving time, and then subdivide the object prior to rendering. (SeeArtist Guide Chapter 3: Modeling for more information on the SS tools.)



Animating Movement

There are two ways to animate movement using trueSpace IK and Bones. You can either move the object and animate the bones accordingly, or you can do it the other way around.

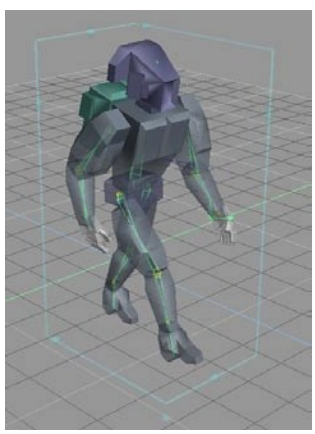
The first option, moving the object as a whole and animating the skeletal movement accordingly offers the most versatility. At any time you can change the objects rotation and movement independently of the IK animation.

Alternately you can use the skeleton itself to create the movement. For example, you can take a biped and place the nail in the left foot. Move the right foot forward at frame ten and move the nail to the right foot, then repeat on the other side. The skeleton will have moved as if walking. The downside is that the axis didn't move in correlation to the character. Rather it stayed where it was at the start of the animation, making rotation and movement more difficult. On the up side, however, you will have avoided foot sliding.

Conclusion

With this information, you should be well on your way to creating animation using Bones in trueSpace. You have learned how to create skeletons for both skinned and rigidly jointed objects, and control their behavior, range of motion, and movement. With practice you will be creating skeletons as second nature.

6.4 Inverse Kinematics



An IK Robot

Although Inverse Kinematics has already been touched upon in the Bones and Skinning section, it is an important feature in trueSpace that can be used for more than just animating characters. An industrial designer may find it useful for creating hinges on a CD player model, for example, so that he can demonstrate how the CD door opens and closes.

Some of the information here is duplicated in the bones documentation, but by reading both sections you will have a complete understanding of the IK process.

The basic idea behind IK is that you have one or more separate objects that are grouped together by joints. The different types of joints available allow movement of the joined objects in specific ways, with different "degrees of freedom." This means that a joint between two objects can move along and/or rotate on any combination of the three axes (X, Y and Z).

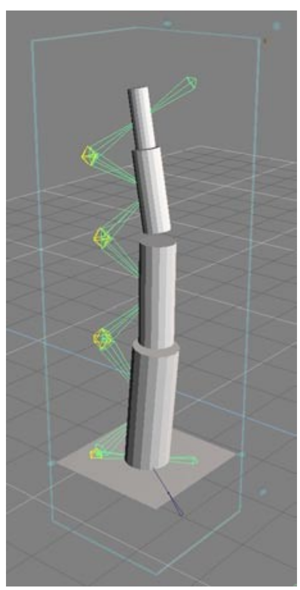
Three possible axes of movement and three possible axes of rotation equal six possible degrees of freedom of movement. With trueSpace's sophisticated IK controls, you will find it is easy to place limits on any of the six degrees.

6.4.1 IK Tutorials

Before you start learning the intricacies of IK, it is helpful to experience some of the basic procedures by following these three short tutorials:

Tutorial 1: Using IK

1. From the program directory, load the object file ivR1T1.cob. This consists of a base and four cylinders, each connected with a 1D translational and 1D rotational joint. In this model, the joints have been moved to the side of the model for better visibility, although normally they appear between component objects.



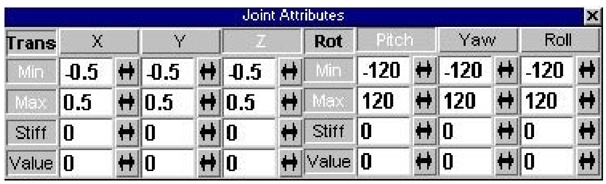
2. In **Object Move** mode, left-button drag the topmost section in various directions. Use the World coordinate system (right-click on the **Move Object** tool), switching between left and right mouse buttons to see the difference. Note that all parts of the object move except the base.

3. Repeat step 2 with other parts of the object.

4. Change the nail, which is the thin blue triangle initially attached to the square pad. Use the mouse to drag the nail onto another part of the object, noting how it snaps to the nearest component object. Then manipulate other parts of the object, as in step 2. Notice how the nailed segment remains immobile while all other component objects between the nailed segment and the one you manipulate move.

5. To move the entire object, drag with the mouse pointer outside the object or on the nailed component object. However, do not drag on the nail itself as this will only reposition the nail.

Tutorial 2: Joint Parameters



Joint Attributes Panel

1. Load the ivR1T1.cob object.

2. Left-click on the **Build Skeleton** tool. You will see a panel with the name "Build Skeleton." One of joints is selected (rendered in brighter colors). You may need to right-click a joint if none are selected.

   Note that the joint has three two-dimensional controllers: a green rectangle that represents the active translational degree (the Z axis), and a blue partial circle that represents the active rotational degree (the pitch).

3. Right-click on the **Edit Joints** tool. This opens the **Joint Attributes panel** (see above), which displays numerically the parameters of this joint.

4. Left-button drag any of the joint's green or blue lines (curved or straight) while checking the panel for changes to learn which line controls which parameters. For details on this panel, see **Changing Joint Parameters** later in this chapter.

5. Select different degrees of freedom (i.e., X, Y, Z, Pitch, Yaw and Roll) in the **Joint Attributes panel**, and then set different Min and Max values. The joint reflects these changes by displaying

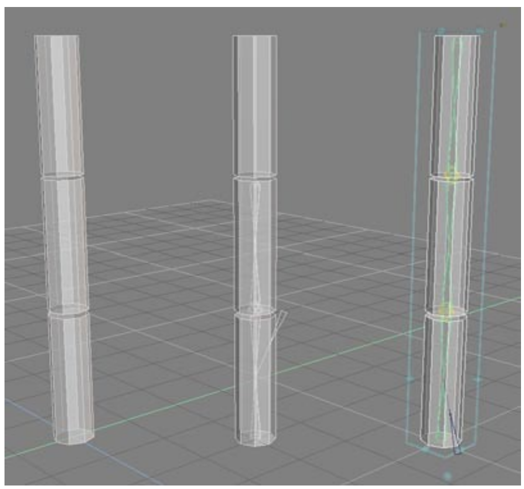
different controls.

6. Select any other joint by clicking on it, and repeat steps 4 and 5. The Joint Attributes panel changes to reflect the current joint's parameters.

Tutorial 3: Building a Finger Object with Joints

1. Add a primitive cylinder , scale it down in all three axes, and then raise it up on the Z axis to make a "bone." Copy it twice, and place the copies end-to-end above the original to form a vertical column. (The illustration below shows this object as well as the joints from steps 2 and 3.)

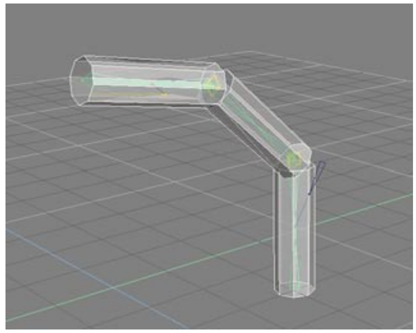
2. Select the middle bone, then activate the **Add 2D Spherical Joint**  tool and click on the lower bone. A joint with two rotational degrees of freedom will be created between two objects. Joints are represented by yellow octahedrons.



Finger after step 1, 2, and 3 (from left to right)

3. Select the upper bone. Activate the **Add 2D Spherical Joint** tool again and click on the middle cylinder. This creates a hierarchical IK structure in which the topmost cylinder is the child of the middle cylinder, which in turn is the child of the lowest one.

4. You've just created a "finger," which, unlike our fingers, can bend in any direction. Drag the uppermost fingerbone in various directions to demonstrate this. Because of the default rotational Minimum and Maximum settings, you can bend the joints only 120 degrees from the vertical, but you can easily change this with the Joint Attribute settings.



Bending a finger

6.4.2 Kinematics Tools

Add Joint Tools

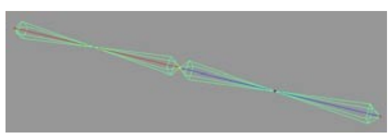
There are six default joint tools available. They are known collectively as Add Joint tools:

Add 0D Fixed Joint

Connect with zero degrees of freedom.

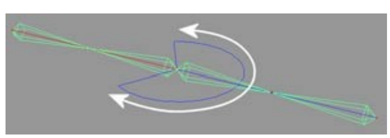
Add 1D Slide Joint

Connect with a translational joint that can move along only one axis.



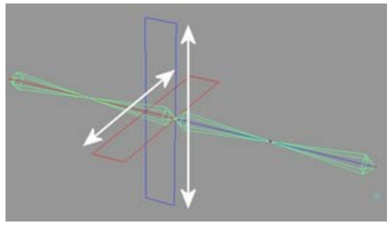
Add 1D Hinge Joint

Connect with a rotational joint that rotates only on one axis (like our fingers, or elbow).



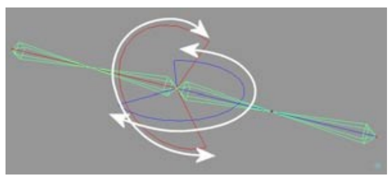
Add 2D Slide Joint

Connect with a translational joint that can move along a plane.



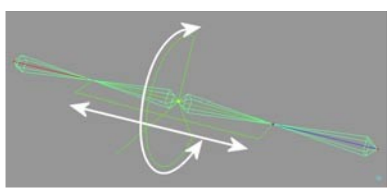
Add 2D Spherical Joint

Connect with a rotational joint that rotates on two axes (a “ball-and-socket” joint, like our shoulder).



Add Shaft Joint

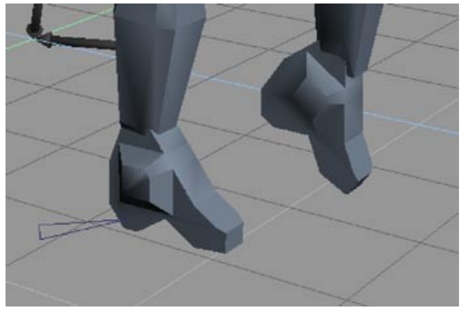
Connect with 1D translational and 1D rotational joint.



Add Custom Joint

Connect with a joint whose degrees of freedom you define (see Building the IK object).

The Nail



A robot's foot "nailed" to the ground plane

You must specify one fixed component object, or "anchor," in each IK object using the "nail." trueSpace creates this entity, displayed as a thin blue-outlined triangle, automatically when you create the first joint and places it on the second object. The nail remains permanently attached to the IK object. You can manipulate it at any time by dragging the Nail object onto a different component object. Its location on the component object to which it is attached is inconsequential.

Note: It is possible to create a complex IK object consisting of more than one IK branch. (This is described later in this section, under Building IK Objects.) These types of complex objects have nails for each branch. A nail cannot be moved outside of its branch. If you attempt to do this, the nail automatically returns to its former position.

6.4.3 Working with IK Objects

Manipulating IK Objects

To manipulate an IK object, the entire object should be selected (i.e., you should be at the top of the hierarchy), and Object Move mode should be active. Click on the end of the chain of component objects you wish to manipulate, and drag the mouse. One way that IK manipulation differs from normal object navigation is that you can manipulate any component object of an IK object without first selecting it. You can grab any part of an IK object and move it directly.

You can move a selected IK object with or without Inverse Kinematics. If, when you drag the mouse, the mouse pointer is outside the object or it is on a nailed component object, the whole object moves according to how the mouse moves. If the mouse pointer is on another component object of the IK object (called an effector), Inverse Kinematics controls the motion of all component objects between the object and the nailed object. A chain of component objects from the effector to the nail behaves according to the degrees of freedom set for joints along this chain and according to how you move the mouse. The mouse motion is translated into a directional force, which is depicted as a yellow line emanating from the effector at the point on which you click. If the Use Torques option is active (see Kinematics Property Panel, below), the mouse motion is also translated into a rotational force about the center of the effector. Thus, clicking on different parts of the effector causes varying object behavior.

Branches of component objects that are connected to some component object in the chain from the nail to the effector are moved and rotated with the component object they are linked to. Any part of the object "above" the effector is moved and rotated with the effector.

Animating an IK object is done the same way as animating any other object. Going to a frame in which a key is to be set, and then manipulating the object, automatically sets an animation key if AutoRecord is enabled. (Right-click the Record Key button to turn AutoRecord on or off.)

Building IK Objects

You build IK objects with the **Add Joint** tools. These tools connect two polyhedra with a joint. The process consists of three steps:

1. Select the polyhedron that is to be the child object of the one it will be attached to. For example, in an arm, the forearm would be the child of the upper arm, so you would select the forearm first. The first object you select can be a standalone polyhedron or a component object of a previously built IK or hierarchical object. In the latter case, you need to first use the arrow keys to navigate through the hierarchy and select the component object.

2. Activate the **Add Joint** tool you want to use (any of the predefined joints or Custom Joint).

3. Pick the second polyhedron. Again, this can be a standalone polyhedron or a component object of other IK object. In this case, you need not pre-select the component object.

When you join two objects, the joint is placed at the intersection of both objects. If objects intersect each other, the joint is placed in the center of intersected parts; if objects do not intersect, the joint is placed at the vertex (or in the center of the vertices of the first object, e.g., the end of a cylinder) closest to the second object.

A joint's location can be changed with the **Build Skeleton** tool. The Edit Joints mode has to be OFF. The **Move Joint** operation of the **Build Skeleton** tool can then be used.

Important note: In addition to combining discrete objects into an IK structure, the Add Joint tools can link pairs of polyhedra that already belong to a standard hierarchical object. Thus, an existing hierarchical object can be transformed, step-by-step, into an IK object without the need to first split the object into separate objects.

You can also use this method to link only a part or parts of the object, maintaining the remainder as a whole, but without connection by joints. This lets you create an object with more separate IK branches. Each branch carries one nail, and can be manipulated separately by dragging on the nailed component object, while still letting you manipulate the whole object. IK objects can be glued together without linking by the joint. This is another method for creating objects with multiple IK branches, in addition to those detailed in the Editing IK Objects section, below.

There is an alternative way of building the IK object in trueSpace. It is the **Link IK Object** tool. It allows direct creation of joints inside a hierarchical object by clicking on desired places. Each click creates a new joint and links two closest objects with this joint. The type of joint that is created can be specified in the **Build Skeleton Properties panel** or the context-sensitive toolbar that Link IK Object provides. Although the click has to be on the surface of the objects to determine the 3D position of the joint, the joint can be moved afterwards, or even in the same step by dragging instead of just clicking. To do this, the Move Joint mode has to be enabled together with the Create Joint mode.

There is also another tool, Attach Object to Bone. It allows not only attaching objects to a particular bone in a Bones object to carry the object without deformation, but it also works for the regular IK object. It allows detaching, attaching and repositioning of objects linked with joints. This tool works much like the Attach Skin to Skeleton tool. Clicking on an object while the Control key is pressed detaches the object from the IK object while keeping all joints that were linking this object to the rest of the IK object. Click ing on the other object on the same place attaches this object to the IK object and links it with all relevant joints with neighboring objects. Dragging of any sub-object replaces this object without destroying the existing joint linkage.

The Joint

In trueSpace's IK implementation, a joint is an object that connects two objects and specifies degrees of freedom of this connection. The joint is not visible when the scene or object is rendered. Each joint can have three rotational and three translational degrees of freedom, of which any combination is possible.

Each degree of freedom has its own parameters that include stiffness and optional lower and upper limits, respectively called Min and Max.

The joint has two graphical representations. Basic representation consists of just a simple octahedron. A more complex representation is evident when you are in the Edit Joints mode of the Build Skeleton tool. The graphical representation during editing consists of connectors, limits and stiffness. The connector is a straight line from the joint, normally pointing to the center of a connected object, drawn in yellow (or red, if unconnected to an object). (Connectors can be rotated, in which case they may not point to the center of a connected object.) Limits, combined with stiffness, are represented as one or more rectangles for translational degrees of freedom, and one or more partial or complete circles for rotational degrees of freedom.

The position of the joint determines its rotational axes. A line from the position of the joint to the center of the first object (called a connector) defines the Roll axis. The axis for Yaw is perpendicular to the second connector, which is a line from the joint to the center of the second object. The axis for Pitch is always perpendicular to the Roll axis. When you create a new joint and after the joint is moved, the rotational axes are in a default orientation, where the Pitch axis is perpendicular to the second connector. You can change the axes' initial orientation with the Edit Joints tool. Moving the joint in the Build Skeleton tool

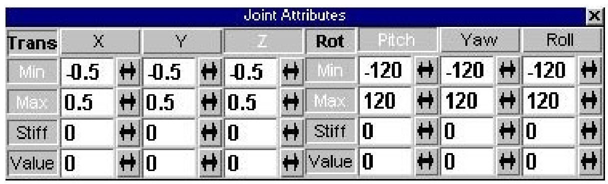
(Move Joint operation) always updates connectors and the orientation of all rotational axes. (See the Bones section for more).

The following color scheme is used to distinguish between the various degrees of freedom:

Blue: X translation and Pitch
Red: Y translation and Yaw
Green: Z translation and Roll

Changing Joint Parameters

You can change a joint's parameters both before and after creating the joint.



Joint Attributes Panel

Right-clicking either the Edit Joints tool or Edit DOF Values Directly tool opens the Joint Attributes panel, which contains default parameters. It shows the active translational and rotational degrees of freedom and optional Min and Max limits and a Stiffness setting for each degree of freedom. You can change any displayed parameter before adding the joint.

You can modify the joints of an IK object at any time with the Edit Joints mode of Build Skeleton or Link IK Object Tool. Activating this mode selects the first joint of the IK object.

With the Edit Joints mode active or by using the Edit Joints tool, you can change a joint's parameters interactively with the mouse by dragging various parts of the joint controller:

When a limit is changed, one of linked objects (the one farther from the nail) rotates or moves to show in real-time what this limit represents. The whole branch of objects linked with joints to this object rotates or moves with it. This object returns back to its original position just after the mouse button is released.

Dragging the whole connector with the right mouse button rotates the connector about its length, thus rotating the joint.

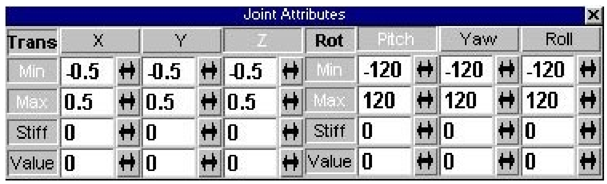
Dragging the whole connector with the left mouse button changes the value of relevant rotational degree thus rotating one of linked objects (the one farthest from the nail) relatively to the second one. The whole branch of objects linked with joints to this object rotates with it.

These changes are updated also in the Joint Attributes panel. You can also change joint parameters by entering new values in the panel. The joint's graphical representation will reflect those changes as well.

Note that you can enter the Edit Joint mode by right-clicking a joint.

When editing joints, you can save the currently selected joint with the Save Object command. This lets you keep a library of joints for building IK objects without having to set up the joints individually each time.

Joint Attributes Panel



Right-clicking either the **Edit Joints tool** or **Edit DOF Values Directly tool** opens the Joint Attributes panel

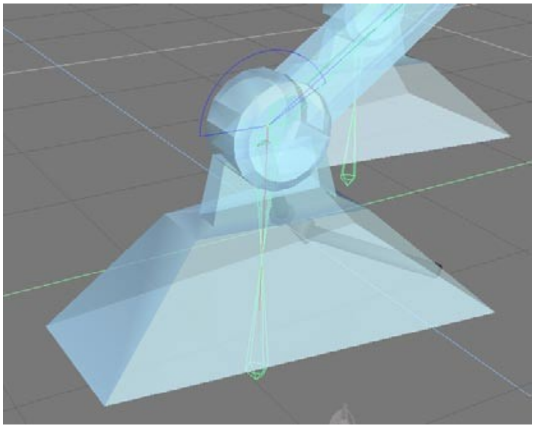
The panel specifies the current joint's degrees of freedom and parameters of all six potential degrees. There are X, Y, and Z buttons for translational degrees of freedom, and Pitch, Yaw, and Roll buttons for the rotational degrees. You can enable and disable each degree separately by selecting or deselecting its button in the panel, and specify each degree's limits numerically with the Min, Max and Stiff settings.

Min/Max Joint Limits

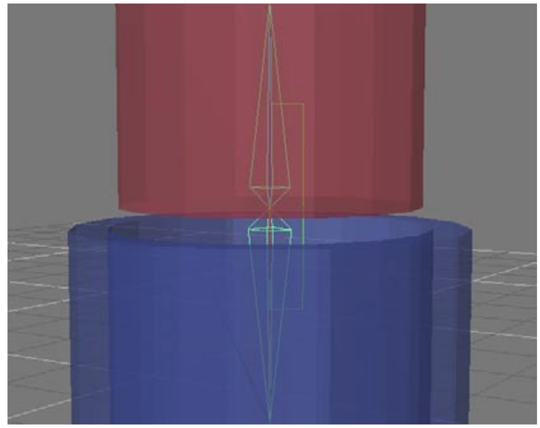
The Min and Max settings let you set the "bottom" and "top" limits of allowable motion for each degree of freedom. You can disable and enable these limits separately for each degree of freedom by clicking the corresponding Min/Max label. Each label is a toggle and is active when "pressed in."

For translational joints, the Min setting can be -10 to 0 units, and the Max setting can be 0 to 10 units. In other words, an object attached to a translational joint can be moved up to 10 units away from the joint's center along any of the three axes.

For rotational joints, the Min setting can be -360 to 0 degrees, and the Max setting can be 0 to 360 degrees.



Rotational joint with a Min setting of -60 and a Max setting of +90



Translational joint with a Min setting of -0.5 and a Max setting of +0.5

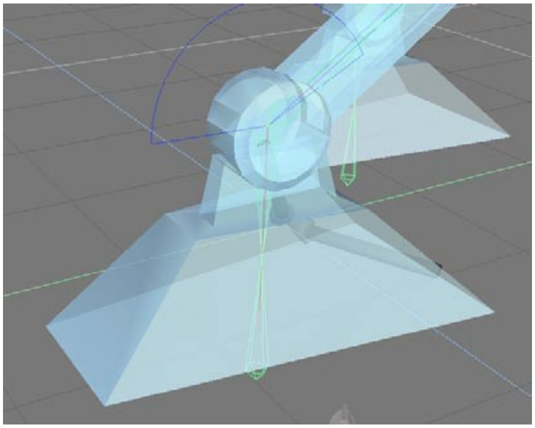
Stiff(ness)

The Stiffness setting lets you determine the amount of “friction” of the respective degree of freedom. The greater the friction, the more relative effort is required to modify the joint when manipulating the IK object. A setting of 0 means no friction, and the higher the value, the higher the friction. There is no upper limit for this value; it goes to infinity, which you can also set by de-selecting this degree of freedom from the joint.

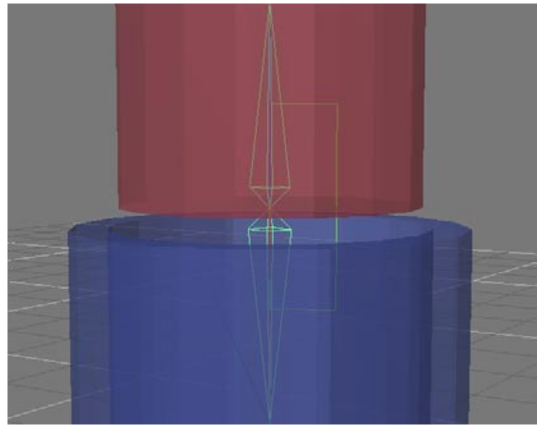
If the **Edit Joints** tool is active, the Joint panel shows the parameters of the selected joint. In this case, the panel settings reflect any change of the joint by directly manipulating the joint’s handles. Conversely, changing numerical entries in the panel causes the selected joint’s handles to change.

If the **Add Custom Joint** tool is active, the Joint Attributes panel shows parameters for the default joint used by this tool. You can change the parameters for this joint before adding the joint. Otherwise, the tool uses current settings, and you can change the joint later with the **Edit Joints** tool.

If you open the Joint panel by right-clicking on one of the predefined joint tools, it shows the parameters for that joint. Although you can change the parameters in the panel, the predefined joint is not altered.



Same translational joint as before, but with a Stiffness setting of 1.5.

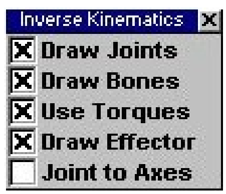


Same rotational joint as before, but with a Stiffness setting of 3.

Value

Joint Attributes panel also contains a setting for "Value" for all six degrees of freedom. It specifies a relative rotation and movement of two objects linked with this joint.

Kinematics Property Panel



Open the property panel for Inverse Kinematics by right-clicking on **Build Skeleton** or **Link IK Object** tools.

There are five check boxes in the Kinematics panel:

Draw Joints

This toggles the display of an object's joint(s).

Draw Bones

This toggles the display and hiding of bones.

Use Torques

This transforms mouse movement into rotational force as well as into the implicit directional force. This behavior is closer to physical behavior.

Draw Effector

This toggles the display of an effector's force (i.e., the yellow line that shows direction) during IK manipulation.

Joint To Axes

This toggle affects the **Add Custom Joint** tool. When it is unchecked, the joints created will be in the middle of the two objects. With this option selected, the joint will appear at the axis of the second object, thus maintaining its center of rotation

---

6.5 Facial Animator

Facial Animator

The Facial Animator is a powerful trueSpace extension for creating facial animations.

It allows you to:

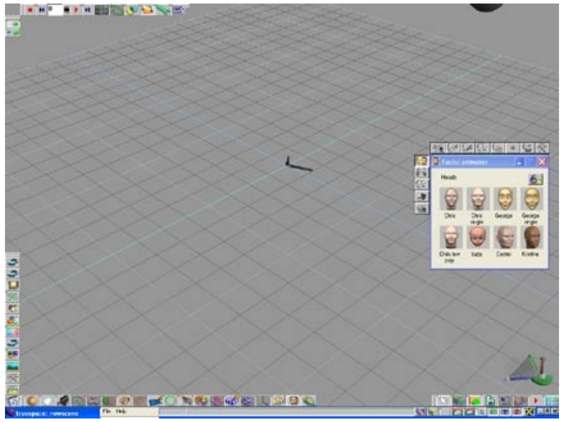
- Make heads speak desired text (using synthesized speech) or your own speech (using a previously recorded .wav file).
- Set various emotions for talking heads.
- Photorender animation using key frames as well as add sound to rendered video automatically.
- Change head shapes or textures using two photos.
- Add custom heads using a native wizard. Custom poses can also be defined.
- Record / restore scenes with talking heads.

NOTE: The Facial Animator requires MS Windows 98 or higher. Microsoft Speech is not supported by Windows 95. DirectX 3.0 or later is also required, as is a sound card.

6.5.1 Getting started

Step 1: Starting the Facial Animator

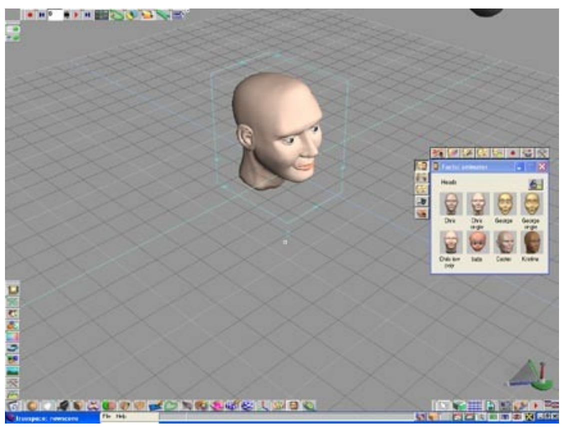
After the Facial Animator is started, its control panel appears on the right side of your trueSpace window.



Note: It is recommended that you clear the trueSpace scene before starting to use the Facial Animator.

Step 2: Adding a head

Now you can add the first head to the scene. For example, simply click on the “Chris Single” icon. The head will be added.

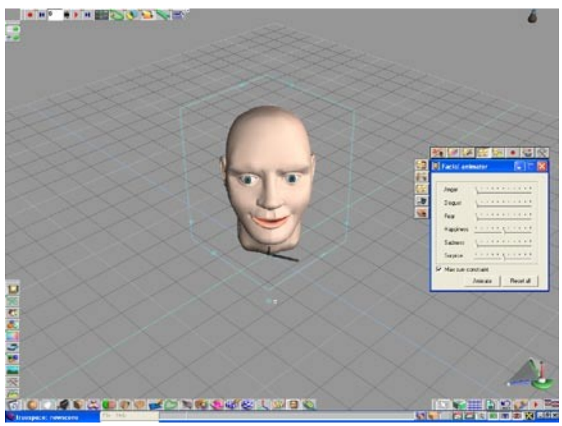


Step 3: Adding speech

After adding the head, click on the **Speech**  icon in the upper toolbar. Make sure “MS Speech synthesis engine” is selected under **Select speech engine**. Also, rotate the current view to see the head details using trueSpace’s navigation widget (see picture above). Add the desired text in the **Text of speech** box. Now press the **Talk** button to start real-time synthesized speech of the head.

Step 4: Setting emotions

To define emotions for the head, click on the**Expression setup**  icon. A tab with many control sliders appears. Use the sliders to determine the head's emotions. For example, if you move the "Happy" slider to right, the head starts to smile.



6.5.2 Basic Concepts

Facial animation and expression setup

Once a head is added to the scene, you can create two basic kinds of animation:

- Speech animation
- Expressions and gestures animation

All animations are displayed in real time.

If you want to record the animation (make keyframes), press the **Record** icon on the upper toolbar. After you click **Record**, all animation will be recorded. With **Record** unchecked, you will see only a preview of facial animations.

Each pose you can assign to the head consists of four types of head deformation:

- Head shape
- Expression
- Gesture
- Viseme (specific mouth shapes during talking)

You are allowed to define each part of a pose. Head shape is usually set only once per head; however, you can also animate a head shape change. Visemes are automatically animated during speech (or can be set manually for special animations). Expressions and gestures can be changed frequently during speech.

Editing a head's shape and texture

When you want to edit the shape of a head model, there are two tools to accomplish this:

- Use the **Head texturing editor** for setting up head geometry as well as texturing, using two photos;
- Use the **Head geometry setup** panel for setting the head shape. (This option is less precise and sophisticated, but you don't need any photos.)

Saving and loading of created heads

You can save and restore heads just like any other trueSpace object. (I.E. you can create a trueSpace library of talking heads.) Also, if applicable, keyframes will be recorded / restored.

Custom heads and custom poses

You can enhance the versatility of the Facial Animator by adding your own head models to it. After a custom head has been added, it can be animated just like a built-in generic head.

Custom heads can be added using **Custom Head Wizard** (see **Custom Head Wizard** section).

Notes regarding the addition of custom heads:

- Currently, you can add single mesh objects as well as hierarchical objects (e.g. a head with glued eyes, teeth, and hair). (**IMPORTANT NOTE:** In the case of a hierarchical object, the main portion of the head must be named "Head" in order for it to be recognized by the wizard.)
- There is also a limit for the number of vertices in the head (adding heads with more than 20,000 vertices is impossible for most computers due to the huge memory consumption that would occur during the weight computation process).

General rules regarding custom poses:

- All custom poses are defined only for the current head being used (i.e. when you define a new pose for the "George" head, it will not be available for the "Chris single" head).

- You can override built-in poses by specifying names of existing poses during an “expression deposit” session. This is especially useful to fine tune poses (e.g. eye blinks) for custom heads.

Applying SubDivison to heads

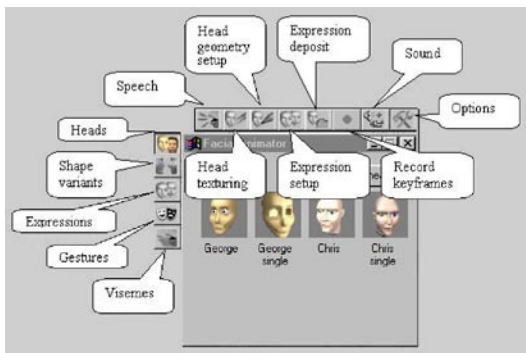
You can smooth your facial animations by applying **SubDivision** to the heads. You can subdivide a head:

- before any facial animations are keyframed.
- or after keyframing as well.

It is also possible to revert to the non-subdivided state of a head by using the “extract control mesh from subdivision object” function.

6.5.3 User interface reference

Toolbars layout

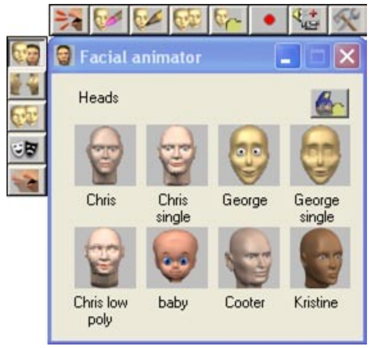


There are two toolbars for the main Facial Animator panel:

- Horizontal (tools) toolbar
- Vertical (libraries) toolbar

6.5.4 Libraries

Heads library



The **Heads library** is the default panel of the Facial Animator. You are not allowed to switch to a different panel until you add (at least) one head to the scene.

The **Heads library** contains icons of generic heads available in the Facial Animator. There are two kinds of basic heads :

- Built-in heads (e.g. George, Chris)
- Custom heads

Custom heads (if any) are always displayed after built-in heads (see example above).

To add a built-in head to the scene, simply click a head icon. You may add several instances of the same head. (This is similar to trueSpace object libraries).

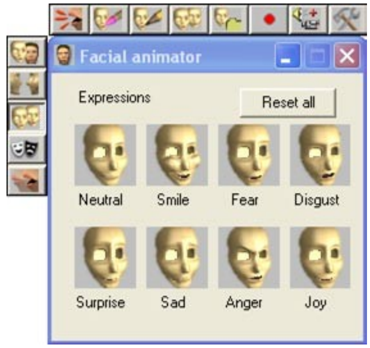
Click the **Custom head wizard** icon to start the **Custom Head Wizard**.

Custom heads management

Right-clicking on the panel brings up the "custom heads management" popup menu. It contains:

- **Delete** … Deletes the selected custom head from library.
- **Edit** ... Invokes the **Custom Head Wizard** again to re-edit the custom head.

Expressions library



The **Expressions library** contains icons of saved expression poses. There are built-in expressions (which can be optionally overridden or extended by custom poses).

To apply an expression to the head, simply click on the icon. You will then see animation to the target pose.

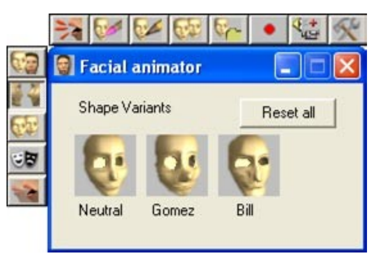
Note: if you want to apply several poses at once, right-click on the pose icon and pick "Apply over" from the pop-up menu. This feature is especially useful for creating gestures, e.g. to combine a left smile with a right eyeblink, etc.

Note: You can fine tune expressions using the **Expression setup** panel.

Custom poses management

- Right-clicking on the panel will open a pop-up menu allowing you to delete a selected expression from the library.

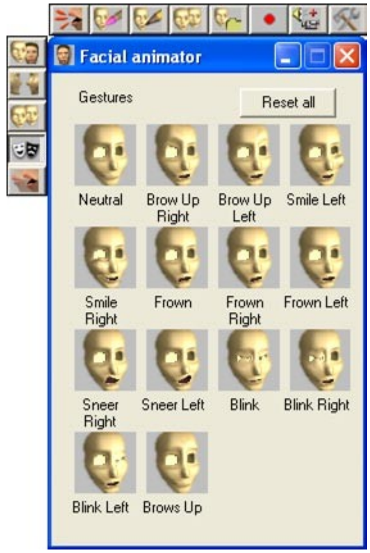
Shape variants library



The **Shape variants library** contains icons of basic head shapes.

For functionality and custom poses, see **Expressions library** description.

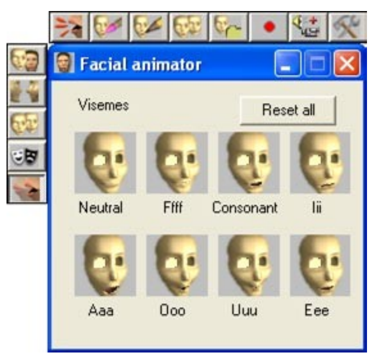
Gestures library



The **Gestures library**  contains icons of basic gestures.

For functionality and custom poses, see **Expressions library** description.

Visemes library



The **Visemes library**  contains icons of basic visemes.

For functionality and custom poses, see **Expressions library** description.

6.5.5 Speech

The **Speech**  panel allows you to make the head talk without detailed editing of face points.

Speech Engines

The Select Speech Engine list allows you to choose the speech engine to be used for talking. Currently, there are two speech engines:

- MS Speech synthesis engine - this is a speech synthesis engine based on Microsoft Speech. It uses text to be synthesized as a source.
- Internal speech recognition engine - this engine uses a .wav file with previously recorded speech as a source for making talking animations.

Speech Files

In the Internal speech recognition engine, you can select a file to be spoken. There are predefined .wav files in the \speech subdirectory, and you can add your own recorded files as well.

- Use the **Load** button to add .wav files to the list.
- Use the **Delete** button to erase files from the list.

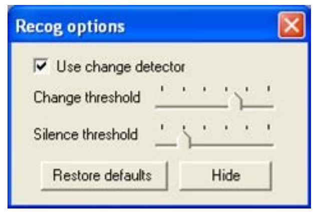
Text of speech

In the MS Speech synthesis engine, you can type in text to be spoken in the edit field.

Advanced users can write special XML tags for the MS Speech engine to adjust pitch, make pauses, etc.

Speech Control

- Click the **Talk** button to start speech and head animation.
- Press the **Stop** button to stop the speech currently being animated.
- Press the **Generate wav** button to generate a .wav file: if the Speech synthesis engine is selected, you can generate a .wav file with the current text synthesized to the disk (a save dialog box appears). This can be useful when you want to add sound manually to rendered video (when using the Sound functionality, this is not needed).
- Press the **Options** button to show the current speech engine's options panel:



- MS Speech properties: you can specify the speech rate, volume, voices (currently three), and check or uncheck the **Process XML** checkbox.
- Recog options: you can uncheck **Use change detector**, or adjust sliders to tune sensitivity for viseme changes. Use the **Silence threshold** slider for adjusting the silence threshold (useful for eliminating speech detection during noisy pauses).

Animation of Speech

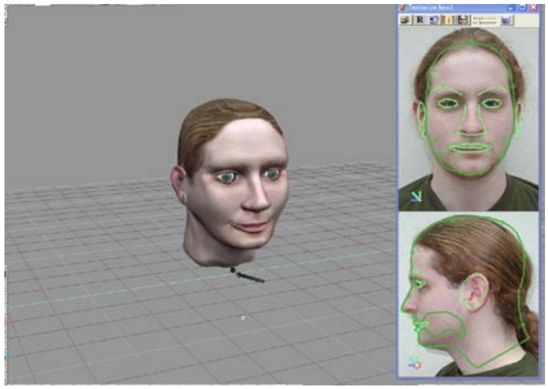
You can make the head talk using the **Speech library**'s features (see the **Speech** section of the user documentation).

Some notes:

- Real-time animation of speech runs on a variable frame rate, depending on your computer's speed. You can check the current frame rate on the **Options** panel, in the **Average preview FPS** field.
- When storing keyframes, the speech versus animation timing may appear strange - the speech may be at normal speed, but the animation may be much more slower and not appear to match the sound. THIS IS OK. The asynchronous manner allows you to get 30 frames per second in the rendered animation regardless of your computer's speed.

6.5.6 Head Texturing

Head texturing editor



The **Head texturing editor** is a powerful tool, allowing you to:

- Edit the shape of the head using two orthogonal photos.
- Create a texture for the head by blending source photos.

When the editor starts, the main **Facial Animator** window is hidden.

You can resize the head texture editor to better see the photos' details. Use the **maximize** icon to display full screen photos.

Usage

Set images (or use default) at the start of the editing session. (See the **Change images** icon reference.)

Use the navigation controls to resize (and also to rotate) the green curves representing the head outline to best match to your photos (use outer control points, like the top of the head, as criteria). You can move the head by clicking and dragging in the head area outside of the green curves.

Control reference:

- **Cyan bar:** uniform scaling.
- **Green bar:** vertical scaling.
- **Blue bar:** horizontal scaling.
- **Red ring:** rotation (only for side view)

With the green curves resized properly, try to move the control points to best fit to your photos. You will see the deformations on the head immediately.

There are two feature point editing modes:

- **Fast curve mode (default):** There are green curves (eyes, nose, ears) displayed on the images. You can move whole curves to create a basic head geometry setup. The green editor curves may not match the image exactly. They are to be used primarily for changing the 3D head shape and adjusting textures.
- **Point editing mode:** For this mode, you can adjust any feature point. The mode is intended for fine-tuning head geometry.

Right-click on images to switch between edit modes.

Once the geometry has been (more or less) set, apply the texture by pressing the **Apply Texture** icon. The texture will be computed and set, and you can continue with geometry editing as well as with texture tuning such as adjusting the seam slider, or changing UV mapping type.

You can resize the editor's window to better see a photo's details. Use the **maximize** icon to display full-screen photos.

All changes you make are permanent. You can use the **Revert state** button to undo all changes made in the current editing session. You can also reopen the texture editor for the head several times to make further adjustments.

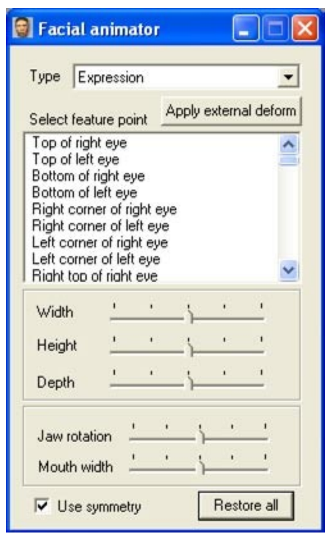
Toolbar reference

- **Default photo display:** The Facial Animator will initially display your images side-by-side, as illustrated above. To display your images vertically, click and drag to narrow the window. Return to the horizontal display by clicking and dragging to expand the window horizontally. Enlarging or reducing window size will cause the image sizes to be adjusted proportionally. Both images are always displayed in the window. Right-click to toggle between feature and point modes when editing.
- **Display photos full screen:** Use the **maximize** icon to display photos side-by-side at full screen size.
- **Change images** : The image browser will be displayed. Select front and side view photos

(in this order).

- **Reset feature points** **:** Reset all feature points to return the head to its basic, non-deformed state. This is especially useful when changing images.
- **Revert state** **:** Return to the state (geometry, images) of the head when the editor was started. (This is a simple variant of the “Undo” feature, especially useful when re-editing the head).
- **Toggle symmetry / no symmetry mode** **:** With the symmetry mode turned on (default state), all feature points or curve movements are done in a symmetrical manner (i.e. when you change the left eye, the right eye will also be changed).
- **Apply texture** **:** Creates and sets texture from both photos. Once you have clicked the button, the texture is set.
- **Texture seam depth slider** **:** As final textures are composed of 2 images, there is a border where the front image is connected to the side image. You can adjust the seam depth using the slider (reasonable values are about 1/3 of slider’s max).
- **Toggle cylindrical / planar UV mapping** **:** You can change cylindrical texture UV mapping (default) to planar and vice versa.

Head geometry setup



The **Head geometry setup** panel provides functionality for low-level editing of head shape and creating new poses (in conjunction with the **Expression deposit** tool).

Usage

First, pick the pose type using the **Type** combination to be edited. There are four types:

- **Expression**
- **Shape**
- **Viseme**

- **Gesture**

I.E. When you want to create a gesture custom pose, choose “Gesture” here.

Select feature points by clicking on the feature point list. With a point selected, edit its displacement using the sliders below (**Width**, **Height**, and **Depth** correspond to the x, y, and z object axes).

More complex operations

- **Jaw rotation:** Use the slider to create whole jaw rotation (useful especially for visemes or gesture designing).
- **Mouth width:** Expand / shrink mouth width with the slider.

Other controls

- **Use symmetry:** All feature point displacements will be done in a symmetrical manner (i.e. when you alter the left eye, the right eye will also be changed).
- **Restore all:** Reset all feature point placements (return to head’s non-deformed neutral state).
- **Apply external deform:** You can deform the head using trueSpace tools like **PointEdit**, **Deform** tool and so on. After editing, press the **Apply external deform** button (otherwise all the changes will be lost). **NOTE:** during tS editing there are no topology changes allowed (e.g. adding or erasing vertices).

Expression setup

With the **Expression setup** panel, you can compose expressions by mixing several basic emotions together.

Usage

Simply adjust the sliders' positions to create the desired expressions. You can combine, for example, ***Happiness*** and ***Surprise***, ***Anger*** and ***Disgust*** and so on. With **Max sum constraint** turned on, you must lower some sliders' positions before setting another.

Other controls

- **Max sum constraint:** With this feature turned on, the maximum sum of all sliders' values must not exceed 10. (I.E. when you set, for example, the Happiness slider to the max value, you are not allowed to set any other emotion. Or you can set the Happiness to value 5 and the Surprise value also to 5). This feature prevents you from creating exaggerated poses.
- **Reset all:** Resets all sliders to zero. (Neutral expression).
- **Animate:** Animate transition from original expression state to the target state. The feature can be useful for key framing expressions changes.

Expression Deposit Tab

The **Expression Deposit**  panel provides functionality for saving custom poses into the poses library.

Usage

You must deform the head (using the **Head geometry setup** tool) first. Then,

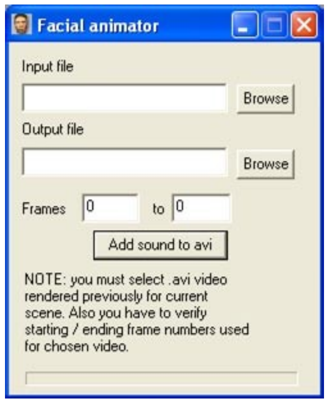
- Fill in name of the pose. (This must be a unique name for a new pose, or an existing pose name if you want to override it.)
- The **Type** combination value is set according to the **Type** combination in the **Head geometry setup page**. There are four types:
  - Expression
  - Shape
  - Viseme
  - Gesture
- Confirm the thumbnail (rotate the head to adjust it up).
- Click the **Record** button to record the pose to the library.

NOTE: The Expression Deposit Tab can be disabled for some reloaded heads (those that do not have a corresponding generic head in the Heads library)

Store keyframes

Store keyframes is a toolbar checkbox. If checked, all head animation (speech, poses) will be recorded into keyframes (this feature is similar to trueSpace's **Record** function).

Sound



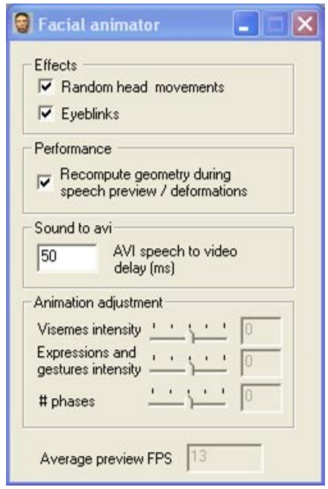
The **Sound panel** allows you to integrate speech sounds into a rendered .avi file. The Facial Animator uses information saved in special audio key frames to retrieve .wav files used and their starting positions in whole animations.

Controls

- **Input file:** Specify (or pick using the **Browse** button) an input .avi file. This must correspond to the .wav file used to render the current scene.
- **Output file:** Specify (or pick using the **Browse** button) an output .avi file. The name must differ from input file's name.
- **Frames:** Specify starting and ending frame numbers that have been used for rendering the input .avi. (Default values correspond to the entire animation.)
- **Add sound to .avi:** This button starts the video and sound mixing machinery. Animation playback will start in the scene. At the end of the process, you will get either a success or an error message. Errors can be caused by different .wav file formats being used.

  NOTE: Beginning with version Facial Animator 1.1, you can generate overlapping speech (e.g. several voices talking concurrently in conversation).

Options



The **Options**  panel enables you to set various Facial Animator options.

Controls

- **Random head movements:** The head will make random rotations during speech, as well as random brows-up / brows-down movements.
- **Eyeblinks:** There will be random eye blinks during speech.
- **Recompute geometry:** Check the checkbox to force trueSpace to recompute geometry changes for each frame during speech preview / head deformations in the texture editor. Performance will slow down by approximately 30 %.

  (NOTE: During photorendering, there will be always geometry recomputation, regardless of the checkbox state.)

- **.AVI speech to video delay:** Specify speech-to-video delay for photorendered animations. This value is in milliseconds. The idea behind this option is that video animations can start a little bit earlier than sound to increase realism. (I.E. the mouth can be already opened when the "a" phoneme start sounds.) You can tune this option to bring more realism to your animations.
- **Visemes Intensity Adjustment:** Use the slider to adjust the intensity of mouth movements during speech. This feature is especially useful for custom heads.
- **Expression and gestures intensity adjustment:** Use the slider to adjust the intensity of expressions and gestures. For example, you can eliminate exaggerated emotional effects using this slider.
- **# phases:** Use the slider to control the animation speed. This is very useful in creating fast or slow speech or gestures.

(NOTE: Adjustments have a global effect for all heads in the scene.)

Texture Naming Rules

- All textures are saved to the current project texture directory.
- If you have previously given your head a filename, the texture filename will be the same as the head's but with a .jpg extension. You may specify the head's filename with any of the following functions: **Load Object**, **Save Object**, **Save Object As**, **Insert to Object Library** , or **Load from Object Library**.

  (NOTE: The object file names are not saved in the scene files, so after a scene is loaded all objects will have unspecified names.)

- If the object and scene names are not specified, the texture name will be HeadTexture.jpg.
- When the object and scene names are specified, the texture name is assembled from the scene and object names. For example, a new scene with the Chris head would be: .\My FA Project\Textures\NewScene.Chris.jpg

  (NOTE: You can quickly locate the texture names by opening the **Material Editor**, choosing the **Inspect** tool, and looking for the filename in the **Texture Options**.)

Blending Head Textures

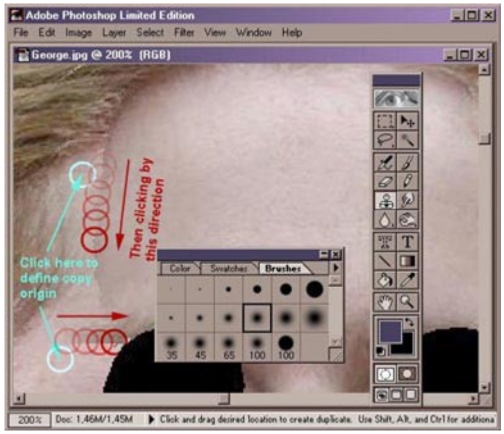
The Facial Animator allows you to join the front and side textures into just one texture. Although there are blending algorithms included, in some cases the blended texture seam may be too visible. To minimize the seam's visibility, use an image editor such as Adobe Photoshop. The process is simple:

- Project a head by Facial Animator and apply the texture.
- The joined texture will probably look similar to this one.

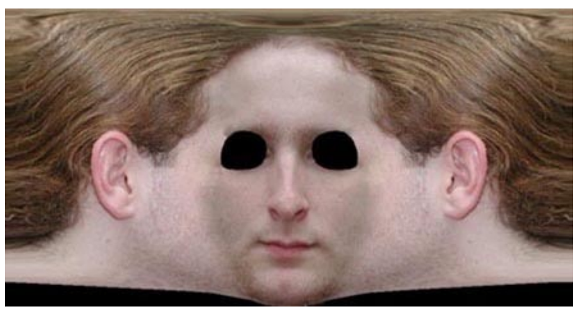
- Once you have created the texture, load it into Adobe Photoshop. The texture can be found in the trueSpace texture directory under the same name as the object it is mapped onto, e.g. Chris_Single.jpg.
- Select the **Rubber Stamp Tool** and choose an appropriate brush size (this will depend on the

texture size).

- Go around the whole seam and replace the seam edge as in the following example.



- **Note:** the copy origin is defined by holding the Alt key and the left mouse button.
- When the texture is fixed (generally it takes about 3-5 minutes), save the picture. The next rendering will use the corrected texture, which should look like this example.



6.5.7 Custom Head Wizard

The **custom head wizard** enables you to import custom heads into the Facial Animator.

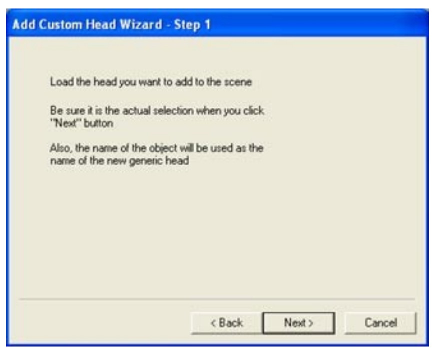
You can start the wizard from the**Heads library** panel (the start panel of the Facial Animator) by clicking the **Custom head** button.

You can also re-edit a custom head once imported by right-clicking on the head in the heads library and choosing **Edit** from the popup menu.

WARNING: After pressing the wizard's **Next** button, all of the heads in the scene will be erased. Please record your work first.

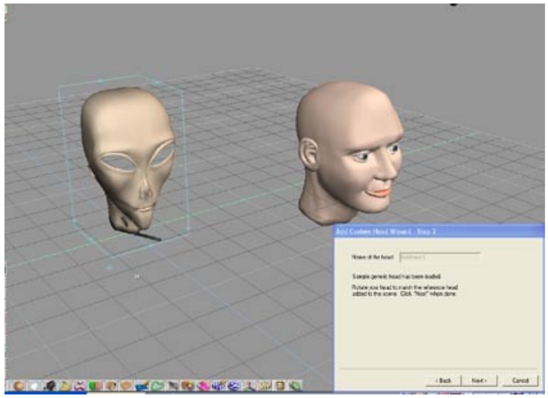
Step 1 : Load and name new head

In the first step, you are prompted to load the new head, and set the name of it.



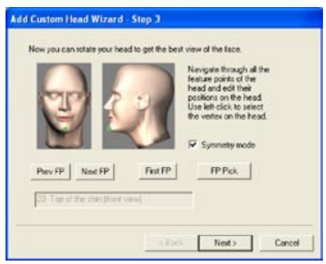
Step 2 : Rotate the head

During the second step, the reference head will be displayed. You must rotate the new head to match the reference head's rotation (see picture above). This step is critical for proper functionality of the new head.



Step 3 : Set up feature points

The third step lets you set up feature points for the new head.



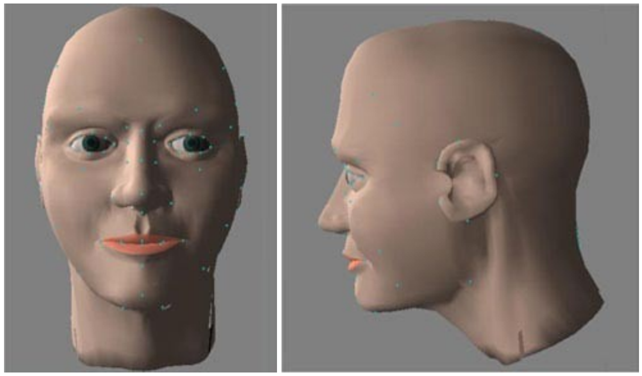
First, rotate the new head to get the best view of it (there is no need to preserve the rotation from step 2).

Navigate through all the feature points and set their positions on the new head (pick vertices using the mouse, similar to trueSpace's point edit mode). See the green points on the reference front and side pictures for a better understanding of intended point location. As there are 60+ feature points for the head, this process can be time consuming.

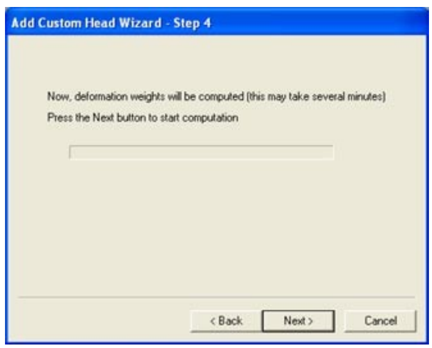
With "Symmetry mode" checked, you only have to define feature points on the right side of the head. Positions of corresponding points on the left side will be computed automatically.

You can use the **FP Pick** button to pick a feature point. After the button has been clicked, all feature points on the head are highlighted. Click on the desired feature point to select it.

Here is an example of properly assigned feature points:

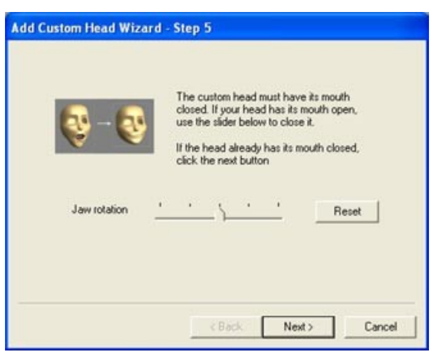


Step 4 : Weights computation



When you press the **Next** button on the fourth step's panel, deformation weights will be computed (for more complex heads this can be a very slow and memory-consuming process, so please be patient).

Step 5 : Mouth closing



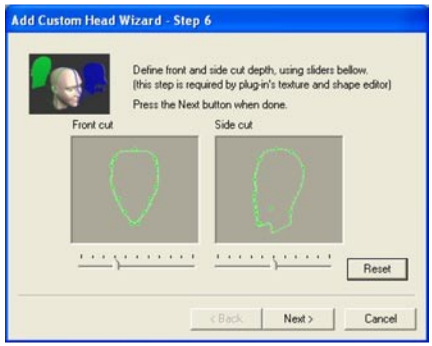
If the new head has its mouth open, you can close it during the fifth step.

A closed mouth is required for proper functionality of the custom head.

Use the **Jaw rotation** slider to close the mouth (you will most likely move slider to the left).

Use the **Reset** button to revert any jaw rotations.

Step 6 : OUTLINE CUTS Definition



This step allows you to define depths for the front and side outline cuts. These outlines will be used with the head texturing editor.

Use the **Reset** button to revert to the default depths.

Step 7 : Thumbnail Selection

Select the best view for the thumbnail.

As the head will be saved during this step, it is strongly recommended to move the head to the 0/0/0 position and set a neutral rotation for it before pressing the **Next** button.

Step 8: Finishing

When you can see a panel displaying a congratulations message, the head has been successfully imported into the Facial Animator generic heads library. Click **Finish** to complete.

6.5.8 Advanced Topics

Adding phoneme transcript to .WAV

The internal speech recognition engine can recognize some phonemes (smallest units of human speech) on your speech .wav file. If you want to improve the quality of the resulting animation, you can try the following:

- Adjust the controls in the **Recog options** panel (e.g. silence threshold, change threshold).
- Write a phoneme transcript file corresponding to your .WAV file.

Let’s look at the second possibility, writing a phoneme transcript, in detail:

- Create a new text file with same name as the .WAV file, using the “phn” extension, and place the file in the .WAV file directory.
- The .PHN transcript file consists of:
  - A 2 line header
  - Several lines in the format “<startTime> <endTime> <phoneme>” where <startTime>, <endTime> are times in milliseconds for start/end of a given phoneme.
- As you can see from the .PHN files included in the Facial Animator’s \speech subdirectory, phoneme names are quite complex. But for manual editing you can use following basic phonemes and their mapping rules:
  - “.” or “!” (will be mapped to “Neutral” viseme)
  - “a” (will be mapped to “Aaa” viseme)
  - “i” (will be mapped to “Iii” viseme)
  - “e” (will be mapped to “Eee” viseme)
  - “o” (will be mapped to “Ooo” viseme)
  - “u” or “w” or “q” (will be mapped to “Uuu” viseme)
  - “f” or “v” (will be mapped to “Ffff” viseme)
  - “m” or “b” or “p” (will be mapped to “Neutral” viseme)
  - “ai” (will be divided into “a” and “i” parts)
  - “ei” (will be divided into “e” and “i” parts)
  - “ou” (will be divided into “o” and “u” parts)
  - <any other character> (will be mapped to “Consonant” viseme)

Example of manually edited .PHN file (transcript for .WAV file with “3 3 4” numbers enunciated:

```
MillisecondsPerFrame: 1.000000
END OF HEADER
1 92 .
92 170 t
```

170 245 r
245 445 i
445 787 .
787 947 .
947 1020 t
1020 1090 r
1090 1231 i
1231 1384 f
1384 1548 ou
1548 1630 r
1630 1949 .

NOTES:

- The technique of manually editing of transcript files is very difficult and can be extremely time consuming. You can also use external recognition software to do the task for you.
- As writing transcript files was intended for advanced users only, there are no special supports or conveniences for it (changes or mistakes in .PHN file format can lead to unpredictable results).

Adding XML tags to MS Speech source text

The "MS Speech text synthesis engine" provides ways to enhance the quality of the text-based speech.

You can add special XML tags into the input text, which allow you to:

- Set or change the pitch of the voice within a single phrase
- Change volume of the speech within a single phrase
- Add pauses to a single phrase
- Etc.

Example:

"Hello <pitch middle = "-10"/>world."

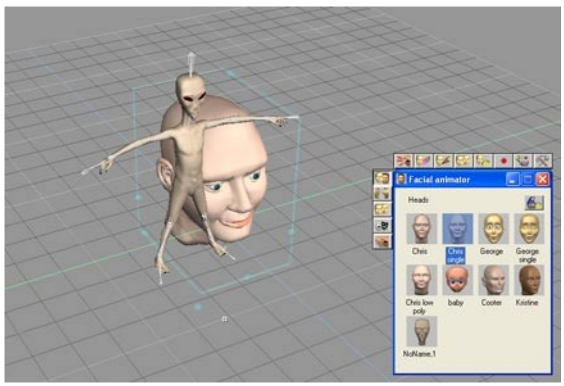
Adding the text above to the MS Speech engine will lead to "hello world" speech with the pitch lowered between the words.

Complete XML tags reference can be found in the MS Speech SDK documentation, at http://www.mi - crosoft.com/speech.

Adding a Head to a Skeleton

- Place the body into the trueSpace scene - in this example, the IK alien body was loaded.

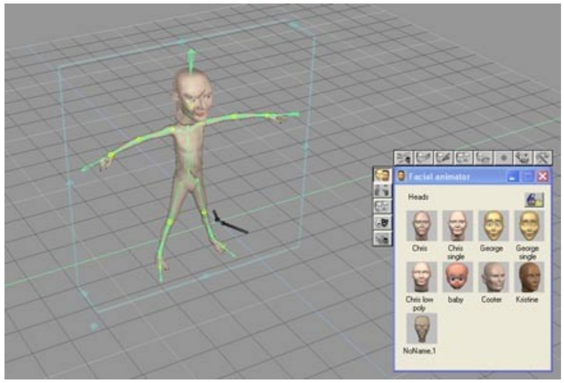
- Insert a head into the trueSpace scene - in this example, the Chris single head.



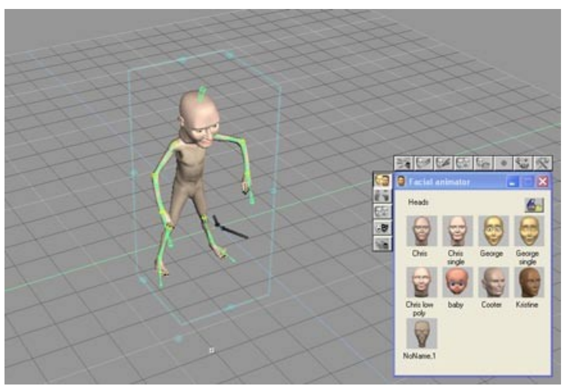
- Position the head to the proper position with respect to the body.

- If needed, create an additional bone ("head bone") in the skeleton by using the **Add new bone**  tool. This bone should stick out of the neck of the body in a vertical direction.

- Select the body.
- Next, start **Attach object to the bone**  tool from IK pop-up.
- Select the head and close the tool - now the head is attached to the corresponding bone of the skeleton. Note that the bone the head is attached to is selected automatically based on the proximity. When the head is successfully attached to the bone it will turn transparent before you close the tool.



- Once the head is attached you can manipulate the body using IK as usual - note that the head moves/bends with the rest of the body. You should have the whole body selected in order to manipulate it using IK.



- In order to work with the head again select the bone the head is attached to - you can do this by traversing the hierarchy of the object using the arrow keys (down arrow and right arrow until the Facial Animator interface is enabled) - or even better you can do it in the**Scene Editor**. The head is attached to the bone as a child object so you can locate the proper bone in the **Scene Editor** easily. The successful selection of the corresponding bone is indicated by the Facial Animator interface becoming active (the icons will no longer be grayed out).

(NOTE: The IK+FA example objects can be found in the trueSpace\Library\objects.obl library.)

Splitting Dialog Within a Scene

The Facial Animator allows you to create complex dialogs by splitting the dialog within a scene into separate non-overlapping sentences. (Or, also into overlapping speech since version 1.1.) You may also add expressions to the head that is listening while the other head (or heads) in the scene is talking. For example:

- Add two heads to the scene (here, Chris and George).
- Keyframe the phrase “How are you?” as speech for Chris (i.e. keyframes 0-65).
- Select the George head and set the frame number to 40 in the trueSpace frame selector.
- Keyframe the “smile” expression for George (i.e. frames 40-50).
- Set the frame number to 70.
- Keyframe the reply “I’m fine, thank you. And yourself?” as speech for George (i.e. frames 70-180).
- Set the frame number to 200.
- Keyframe the phrase “I’m fine, too!” as speech for Chris.

If you are using .wavs, create a separate .wav file for each sentence.

Limitations

- Speech synthesis - maximum length of text to be synthesized is approximately 1,000 characters.
- .AVI sound - all .wav files for one video must be in the same format (MS speech synthesis generates .wavs at 22KHz, 16 bit, Mono format).

6.6 trueParticles

trueParticles tool

trueParticles allows you to create complex particle effects using a variety of emission methods, including

- Point source: Particles are emitted from a single point in the emitter icon.
- Volume fill: Particles are created at random locations within the volume of the emitter.
- Surface distribution across any selected object in the scene
- Fragmentation, in which you create and explode into particles a copy of any object.

Once you have selected an emission method, choose from among a variety of particle types, including

- Five standard particle shapes: shard, box, star, streak, sphere
- Square tiles and round tiles designed for specialized mapping effects
- Metaparticles
- Object instances: particles as copies of any object or light in the scene

trueParticles also allows precise, detailed control of particle speed, direction, rotation, and bubble motion (vibration perpendicular to the direction of motion, like bubbles in a glass of soda). For added realism and power, we have also provided a complement of helpers, unique tools designed to multiply your particle power.

Helpers include

- Path Follow: a tool that makes particles follow any selected path
- Blast: a multipurpose, configurable tool that propels particles with the force of an explosion
- Bounce: a tool that turns any object in the scene into a particle reflector
- Gravity and Wind: for fast, easy simulation of these natural forces

With all these special settings, you will be glad to know that trueParticles comes with over a dozen configuration presets, complete particle set-ups that you can load with the click of a mouse. And you can add your own presets, to be accessed instantly whenever you want. You can also save your presets to separate files for any other trueParticles user to load and use.

6.6.1 Definition of Terms

The trueParticles interface is composed of seven layered menus referred to as a “Tab Panels” (or simply “Tabs”). The Helpers tab is further subdivided into five “Sub-Tabs”. Each Tab Panel is divided into a number of labeled boxes (“Groups”).

Controls that are not animatable appear slightly recessed into the dialog box.

> **Note:** The trueParticles interface uses Context-Sensitive Layout. CSL simplifies the user interface, hiding or disabling those controls that are not necessary for your current

settings.

"Variation" controls affect the amount of variance applied to a parameter. For example, a Variation of 50% applied to a Particle Size value of 10.0 would result in particles with random sizes between 5.0 (50% less than 10.0) and 15.0 (50% more than 10.0). For Variation to produce any results at all, the parameter it modifies must be set to a value other than zero.

The scenes for the following tutorials and their final results are included in the Particles Tutorials scene library. To load this library:

1. Left-click the Scene library icon.
2. Right-click within the library's window.
3. Select 'Load' from the menu, and 'Particles Tutorials' from the listing.

6.6.2 Tutorial 1: Blast/Bounce

The purpose of this tutorial is to explore the creation of Blast and Bounce effects using the respective helpers. The end scene, rather than being a finished animation, is a paradigm for exploring the different modes of the two helpers. The aim is to create a pair of particle systems approximating pebbles and dust, which are affected by Blast and react with the surrounding enclosure. While the result of the basic implementation here may not in itself be aesthetically pleasing, it is certainly chaotic!

Load Tut1blastbounce.scn from the Scenes library panel. The scene is set up with a camera, adjusted default lighting, and a texture-mapped enclosure consisting of a three part hierarchal structure of a front right angle wall, a matching back right angle wall, and a ground plane.

The first task is the creation of the first particle system, which is the source of the larger pebbles. Left-click the trueParticles icon to create an emitter at the default center location. Right-click the trueParticles icon to bring up the main trueParticles interface. On the Mode tab, select Box. The idea is to create an emitter parallel to, and just above, the ground plane, which will create the illusion of the extant ground plane being strewn with stationary pebbles from the first frame of the animation. We could elect to match the emitter to the inside dimensions of the enclosure, or make it somewhat smaller so that the effects of the collisions with the walls are more clear.

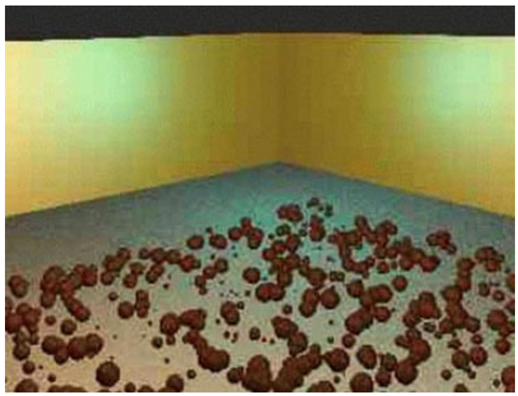
To that end, in the Icon group of the Motion panel, enter the following values:

Length = 5
Width = 5
Height = .001

While this creates a uniform square emitter, the consequence is that it produces a rather unnaturally symmetrical pile of pebbles. (To remedy this, you could get a bit fancy and choose to use an irregular polygon as the emitter, set to Custom Surface instead.) Go to either the front or side view, and translate the emitter up or down so that it is oriented just a bit above the ground, something along the lines of z = .200. This way, the pebbles will not be half buried in the ground plane. (If we were working on recreating a real-world scene, it would take an immense blast to dislodge them if they were half sunk in the ground.)

Now, we need to set the Timing and Motion for this first system. Since we want the pebbles to be present in the first frame of the animation, and we do not really want them to continue popping up as the animation progresses, enter the following values in the Timing and Motion groups:

Emit Start = 0
Emit Stop = 0
Display until 200
Life = 200
Rate = 700
Speed = 0
Spread off axis = 15
Spread off plane = 20



This will result in an animation of sufficient duration to see the effects at work, and a healthy bunch of pebbles to kick around.

We need now to set the parameters for the pebbles. In the Type panel, set the following values:

Sphere
Size = 0.2
Variation = 85%

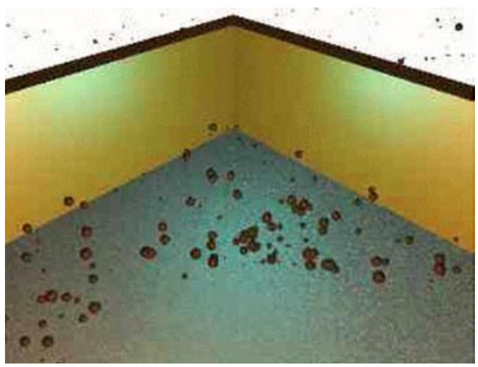
This creates an interesting mix of a broad range of sizes for the pebbles. If we were setting up a more sophisticated version of this scene, we could select a more irregular mesh of a pebble of our own construction via the 'Picked Object' selection, but let's keep it simple here. The actual appearance of our pebbles can be determined here too, via one of trueSpace's native shaders, a second party shader, or the use of a texture map. And, unless you want your particles to change their surface over the duration of the animation, set mapping to 'Distance' and set a high value, like 6000.

Also make sure that 'Spin Time' on the Spin tab is set to 0.

Preview the first few frames of the animation (believe me, the first few frames will be enough, because all you'll see will be the emitter and seven hundred pebbles sitting in the center of the enclosure, dead as, well, rocks.

Now we need to get some action into the scene using the Blast and Bounce helpers. Go to the Helpers>Blast panel and enter the following values:

Use Blast
Start Time = 19
End Time = 20
Strength = .4
Impact Chaos = 100%
Falloff type = linear
Range = 6
Shape = spherical
Center = 1,1,0



These are, of course, just suggested values, but after we've walked through this exercise you can fiddle with the variables to gauge the impact of different shapes, different ranges, etc. You may find that starting the blast at frame 20 leaves too long a period of inactivity at the front end of the animation. If so, move the start/end time closer to the start. But those extra 20 frames give you the space to insert whatever causes the ruckus in the first place into your animation: a UFO, a T-Rex foot, a shower of giant geckos, whatever.

After previewing the animation, you might find yourself distressed by a couple of things. First, all of your finely wrought pebbles have just sailed off into space, as if the enclosure walls weren't even there. And second, while the pebbles indeed have sailed right along, we've not set any spin value for them. The first is the easiest to rectify. The second could be easily set too, but a more satisfactory solution requires setting a key frame.

To keep the pebbles inside the enclosure, and to vastly increase the kinetics of your scene, go to the Helpers>Bounce panel. We're now going to set up the enclosure as one large reflector. Check the box

'Use Bounce' and hit the 'Pick Reflector Object' toggle. Then left-click on the visible wall of the enclosure. This makes the entire hierarchy of the enclosure (front wall, back wall, and ground plane) a reflector for the pebbles affected by the blast. Set the bounce values starting with some base values like those below:

Bounciness = 50%
Variation = 50%
Chaos = 6

This will cause the pebbles that hit the visible wall, as well as the pebbles that are blown out of view behind the camera to bounce back into the center of the enclosure and continue to mix it up as they bounce from the ground plane, other walls, etc. If you wish, you can return at some point and select only the front wall and ground plane to reduce the chaos (but why, why, would you want to do that?).

Let's leave the assignment of spin to your pebbles until a bit later, after you've created your second particle system, the system that generates the smaller fragments or dust that are kicked up by the blast at the same time as the pebbles. trueParticles allows you to make an exact copy of an existent particle system. Go to the Menu panel and click on the 'Copy Particle System' button. tP has created a second, identical instance of the system you created by the above steps, in the same location as the original.

Now, obviously, a second particle system in its present state is going to do you little good. The simple trick is to set about altering the variables necessary to make the second system (TrueParts1 in your Scene Editor) serve your purposes. Examine the table below to see what settings create the alterations we're looking for, and go ahead and make them:

	TrueParts	TrueParts1
Emitter Mode	Box	Box
Length	5	5.3
Width	5	5.3
Height	.001	.001
Particle Timing & Motion		
Emit Start	0	19
Emit Stop	0	20
Display until	200	200
Life	200	200
Rate	700	1200
Spread off axis	15	15.3
Spread off plane	20	20.4
Type	Sphere	Shard
Size	0.2	0.15
Variation	85%	50%
Mapped distance	6000	6000
Spin at start frame	0	Time 30 / random

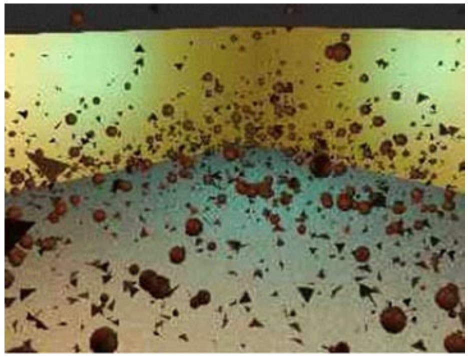
Helpers

Bounce	Bounce on	Bounce on
Reflector	Enclosure	Enclosure
Bounciness	50%	50%
Variation	50%	50%
Chaos	6	6
Blast	Blast on	Blast on
Start time	19	19
End time	20	20
Impact chaos	100%	100%
Range	6	10
Fall off	Linear	Linear
Shape	Spherical	Spherical
Center	1,1,0	1,1,0

Apart from the obvious differences, such as size of emitter and type of particle (we've selected 'shard' here, but it could again be anything else or a selected object), that would immediately set this second system apart in space from the first, we also want to introduce some differences in timing and motion. Since we don't need this dust to start kicking up, or even show in the scene until the blast starts, we've set the start and end times for emission to coincide with the it. And because the particles for TP1 are scheduled to move out as a consequence of the blast in their first frame of life, we can set spin for them as a constant. We couldn't do that for the particles in the first system because we didn't want them to sit there spinning on their axis when they're supposed to be at rest for the first 19 frames of the animation. There's certainly nothing carved in stone about the settings in the table above; most, if not all, can be tweaked to effect to differentiate the second clone particle system from its parent.

As noted above, we didn't want the idle pebbles in the first system spinning prior to the blast event, so now we'll fix it so they spin, but only after the blast has occurred. Advance the animation to frame 19, where the blast begins. Make the first particle system your active object and open the trueParticles property panels. Go to the Spin tab, and deselect the 'Make Constant' toggle at the base of the panel. Enter the spin values you wish to assign to the pebbles, say something along the lines of a random spin with a spin time of ten. Finally, record the appropriate key frames for that spin to extend it for the duration of the animation.

Preview this near final form of the animation, tweaking any settings to enhance the effect. Experiment with the effects of the different types of Blast and their values, and different ranges of values for Bounce. Changes in any of the main variables will introduce perceptible changes to the character of the very energetic, and chaotic scene you've created.



6.6.3 Tutorial 2: Path Follow

The aim here is to recreate with a trueParticles system something akin to a group of insects cavorting in and out of a beam from a volumetric spotlight, with the motion determined via attachment of the tP system to an eccentric path with Path Follow engaged. Toward the end of the animation, when they've tired of their ballet, the insects/particles will break from the path and scatter in various directions via the Blast helper.

The path supplied with this tutorial (object name = path.cob) was created separately, then saved in the TS path library for loading into the scene for use with tP. As much of the motion along the path is a product of settings within tP, only the general complexity of the path needed to be predetermined when it was created.

Paths are defined by the edges around the first face of an object that you select. Paths must be closed loops, but you can pick the percentage of the path that you want to use so it can act as an open path in actual use. If you load a closed path from the path library with nothing selected (hit the Object tool once to deselect the current selection) it will create a polygon object whose edges can be used as a path in trueParticles.

Load Tut2pathfollow.scn from the Scenes library. This scene is already set up with a ground plane, the path, and a volumetric spotlight aimed at approximate center scene (settings for the light are recorded at the end of this tutorial, and please note that rendering settings need to be set to raycast and high quality for the volumetric light to be effective). While a standard camera setup could be used for this scene, working via the perspective view and its adjustments suffice for this exercise.

The next step is to create a particle system with trueParticles by left-clicking the trueParticles icon. Right-click the trueParticles icon to bring up the main interface. Next, go to Helpers> Path tab panel and select ‘Use Path Follow’ by clicking the check box. Next activate the ‘Pick Path’ toggle and left-click on the path loaded in the scene. If your selection has been successful, the pick dialog will disappear, and in the Choose path group, “path : path” will appear.

Render the scene, if you wish, with the default settings, and note that it appears somewhat reminiscent of an abstract sculpture on stage in a smoky lounge at this point. As pleasant a prospect as that may be for another project, that’s not what we’re looking for right now.

While Path Follow functions perfectly well with any of the standard tP emitter modes, let’s select sphere. It will broaden the stream of particles a bit more than will the point emitter, though we’ll take care of most of the stream’s configuration later on. Go with the default radius of 1; variations from that can be worked with later.

First of all, let’s determine the length of the animation and particle timing. In the Motion panel, change the following settings from default:

Emit Start = 0
Emit Stop = 20
Display until 500
Life = 500
Rate = 2

This will give us a decent length animation to show the particles at play, and the Rate and Stop settings will provide a healthy group of particles, but won’t overwhelm the scene.

Next, let’s take care of the type of particle we’re going to set flying. For the sake of this tutorial, let’s select ‘sphere’ in the particle Type panel. As we move further along in this exercise, it’s easy to imagine all sorts of uses for the type of animation we’re blocking out here, with different particle types lending themselves to different effects. We started with the ‘insects in a spotlight’ metaphor, so let’s think how that might be made actual rather than metaphorical: those of you equipped with the hardware to handle it might go with “Selected Object” and send actual mesh insects flying along the path we’re setting up.

Those of us with more modest setups could select round or square tiles, and simply map some .jpg bugs with transparency to the tiles, but we'll save that for another day. We don't want our particles to be terribly large, so lets set a size of .3 and a modest variation percentage, somewhere between 5 and 10%. And we want to give our particles some color, so select a shader or material of your choice and assign it to the tP system at frame 0.

Note: Remember if you're applying a texture map to pay attention to the mapping settings.

As noted above, we're going to want the particles to play along the length of the path, moving in and out of the light beam for almost the full duration of the animation, but not all of it. Let's go back to the Helpers > Path panel and take care of that now by making the following settings:

Percent of path used = 100%
Capture start = 0
Capture end = 485
Capture range = 1000 (or alternately 'Unlimited Capture Range')

This will make sure that all of the particles generated by the emitter are captured from the start and made to follow the full length of the path. We want to end the capture just before the conclusion of the 500 frame animation, at frame 485, when we'll send the particles off in other directions with the Blast helper.

Next, let's set the number of frames it takes for the flock of particles to travel the length of the path. Note that the setting here overrides any particle speed settings made in the Motion panel. Given the length and complexity of the path, one, or slightly more than one, complete circuit should be sufficient for the 500 frames, and we also don't want them moving unnaturally fast, so set Travel Time at 400, with a variation of about 10%.

Preview the animation now and see what we've got. Well it's not modern art anymore, but if they're insects now, they're fairly dyspeptic, and they're showing an annoying tendency to wander a bit off stage, away from the light.

To rectify that while we're still here in the Path panel, let's leave the path radius setting at zero (our bugs are already too close to, or stepping outside of, the perimeter of the scene we're setting up) but incorporate the following settings:

Stream Taper = somewhere between 50 and 60
Variation = 50%
Converge
Stream Swirl = 15
Variation = 50%
CW/CCW = both

Run another preview now. Finally, we've got some serious, serious insect action! But there's still room for some improvement in mixing up the action. Some of the insects look like they're getting a bit too intimate with each other at various points in the path, and even apart from that, a bit more chaos and frenetic action can't help but liven things up. So, go to the Bubble Motion panel and set:

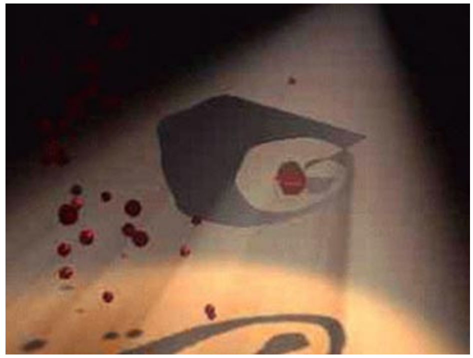
Displacement = 10
Variation = 25%

And if you wish to take it a step further, the settings in the Spin panel can be adjusted, though at this point the default settings may be just right given the other motion we've added to the mix.

So, now we've got our swirling, twirling batch of insects, but to avoid an abrupt end to the animation that leaves them in mid swirl, we need to break up the action in a realistic fashion.

Remember the fifteen frames we left free when setting path capture? Here's our opportunity to send the insects on to our neighbor's backyard. Go to Helpers>Blast and make the following settings:

Use Blast
Start = 485
End = 486
Strength = .3
Impact chaos = 100%
Fall off = none
Type = spherical
Blast center 0/0/0



That takes care of them and winds up the animation. Preview the whole animation again. It can obviously be stirred to personal taste by altering any of the above actions.

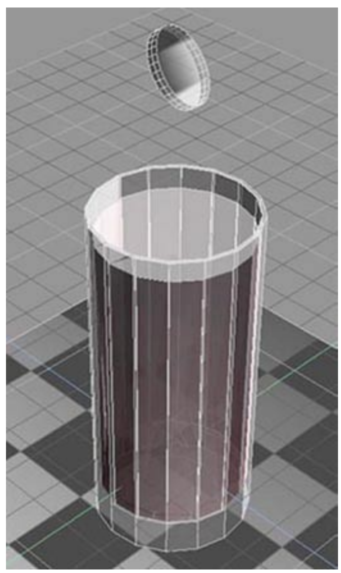
From here on, all that remains is to make the emitter and path.cob invisible and render your animation.

Summary of settings

Mode	sphere
Motion - Particle Timing	Emit start = 0
	Emit stop = 20
	Display until 500
	Life = 500
	Rate = 2
Motion - Icon radius	1
Type	Sphere
	Size = 0.3
	Variation = 10%
	Mapping = applicable if necessary
Spin	Axis = random
	Time = 30
Bubble Motion	Displacement = 10
	Variation = 25%
	Cycle Length = 32000
Helpers - Path	Selected object = path
	Percent of path used = 100%
	Capture start = 0
	Capture end = 485
	Capture range = 1000 (or unlimited)
Path Particle Movement	Travel time = 400
	Variation = 10%
	Stream taper = 50
	Variation = 50%

	Converge
	Stream swirl = 15
	Variation = 50
	Cw/ccw = Both
Helpers - Blast	Start = 485
	End = 486
	Strength = .3
	Impact chaos = 100%
	Fall off = none
	Type = spherical
	Blast center = 0/0/0
Volumetric Light	Linear falloff
	Intensity = 1.87
	Volumetric fog density .05
	Samples = 30

6.6.4 Tutorial 3: Seltzer Tablet and Bubbles in Glass of Liquid



Project: Create a 500-frame animation of a seltzer tablet dropping into a glass of liquid with consequent bubbles and turbulence as the tablet reacts with the liquid

Particle systems:

- **tP system 1:** Located at the base of the glass to create veil of small, less dense, and violent bubbles emitted into the liquid after the tablet action has started (due to overall slight carbonation of liquid by seltzer action). If the animation were continued beyond 500 frames, this would be the remaining bubble action in those additional frames.

- **tP system 2:** Located at base of glass and the main, most turbulent bubble effect, timed to coincide with arrival of tablet near bottom of glass, and to cease when tablet has been largely, then completely, dissolved in liquid.
- **tP system 3:** Animated to roughly coincide with tablet as it drops to create initial bubbles as the liquid begins to react with the tablet. Critical elements: adjustment of speed, life, and fade so that as tablet reaches bottom of glass, this effect is superseded by tP system 3, and bubbles from tP system 2 do not extend noticeably beyond a realistic height above liquid surface.

Setup:

Load Tut3seltzer.scn. The initial setup includes the necessary objects and the three particle systems, although truParts 2 and 3 are hidden. The necessary motion has already been keyframed into the seltzer tablet.

Selecting the first particle system, we find the emitter has been adjusted to the correct shape (cylinder) and size to coincide with the base of the glass (radius = 2.3 and height = 0.1), but all the other settings are default. This is the system which will create the thinner veil of small bubbles which begin to appear dispersed in the larger volume of the liquid after the main carbonation reaction begins - ambient carbonation, if you will. The first thing we need to do is adjust the emission period to match the motion of the seltzer tablet. Bubbles should not begin pouring out of the bottom of the glass until the tablet has reached the bottom, so adjust the time until you find the region where the tablet reaches the bottom of the glass. Remember too, that the bubbles produced by this particle system are a consequence of the carbonation spreading into the liquid as a whole, so it should start just a few frames after the tablet reaches ground zero. Somewhere in the neighborhood of frame 72 looks pretty good, so let's set some trueParticles controls:

Emit Start: 72
Emit Stop: 500
Display Until: 500
Life: 500
Life Variation: 0

Playing the animation, we see that emit start looks pretty good. We still need to tweak the effect, but first we need to adjust the particles to give us more accurate bubbles to work with. Tab over to the Type tab and select the Sphere particle type. Drop the Particle Size down to about 0.1. You should now have an array of little tiny bubbles. Switch back to the Motion tab, and have another look at the animation.

The particles are really flying along. Let's slow them down a little. Reduce Speed to anywhere between 3 and 4 (let's use 3.8), and you'll see the particles adjust to an appropriately sedate velocity.

Now we need to adjust the particle life so the particles die just as they reach the top of the glass. The exact value will depend on your speed and make take some experimentation. (I'll use a Life of 48.)

To improve realism, we should increase the volume of particles, and add a minor variation to the Speed. Be sparing with the variation, however; too much and the effect will look excessive. I'll use values of:

Rate: 2.2
Speed Variation: 2.5%

The one thing these bubbles are still missing is the most important of all: bubble motion. Change to the Bubble Motion tab. First we'll set Displacement, the side-to-side motion, to 0.1. We use a small value because these bubbles are just the fizz of carbonation; they shouldn't move in an agitated fashion and, unless we want to set up the glass as a reflector, we don't want the bubbles wandering outside the glass. Add a Variation of 50% (with such a small displacement, Variation needs to be large, or it will be too subtle to notice). Cycle Time and Phase, with their separate variation settings, may be adjusted as well to stir up the motion and decrease the uniformity.

Tip: We're going to perform a trick here that we'll also use with the other emitters in this scene: by leaving Cycle Time at an extremely high value, the particles will only perform a portion of the bubble movement, because they won't have time to complete even a single cycle during the animation. This will give the bubbles more of a drunken weave than a side-to-side shimmy.

The last adjustment to the first particle system is to work in some Spin for the bubbles. As they will be textured with a plain color and the glass reflectance shader to match the liquid, set for smooth shading, lack of rotation will be less noticeable than, say, if the particles were texture mapped, but we're endeavoring to create as natural an effect as possible here, so even if it's not readily apparent, we can at least know that our bubbles are doing what real bubbles would do in a similar situation. So try the following settings for Spin:

Time: 60

Variation: 33%
Phase: 180
Axis: random

Play the animation through to make sure everything is on track.

Now, select the second particle system. This is the one that really kicks up a fuss and corresponds to the main reaction between the seltzer tablet and the liquid. Again, we've chosen to use cylinder as the emitter type, and a height of 0.1. However, in this case, as the bubble action is going to be more localized in the glass (around the physical vicinity of the tablet when it has reached the bottom of the glass and goes through the movements that are already keyframed into it), it's set at a radius of 1.3.

In the Particle Type Tab, we again want to select sphere as the type, and because these will be the dominant, most vigorous bubbles in the whole scene, a maximum size setting of .3 is a good starting point. We can include a modest amount of variation, say something on the order of 5%, but we'll take care of most of the observable variation in size of the bubbles as they rise through employing the Grow For/Fade For functions. If you observe a seltzer tablet in liquid, the bubbles it generates at its surface start off rather small while they're still attached to the surface of the tablet, and increase in size as more CO2 fills their volume to the point that they break away from the tablet and rise in the liquid. Exact replication of this rather sophisticated action is not possible without a considerably greater degree of complexity than we wish to get involved in here, and besides, with the action moving fast and furiously as it will, this degree of exactitude would most likely be lost on all but the most particular of critics. So grow and fade can help create the illusion that this is happening. We don't want to go overboard on these settings, particularly the fade setting, so try the following as starting points, then tweak on your own time:

Grow For: 8.5
Variation: 5%
Fade for: 0.8
Variation .5%

The emissions should start coincident with the tablet's arrival at the bottom of the glass, and stop somewhere along the line after the tablet had been dissolved and the main chemical reaction has abated. So try these values in the Particle Motion Tab:

Emit Start: 67
Emit stop: 455
Display until 500
Life: 41
Rate: 3.1
Speed: 4.5
Variation: 5%

The life, rate and speed variables should be carefully observed and tweaked to make sure that the particles create the appropriate motion, but don't end up exceeding the top of the liquid by an unrealistic distance.

This system, too, benefits from the Bubble Motion and Spin functions. Because the emitter for this system is smaller in radius than that for the first system, more lateral motion is possible before the bubbles come close to breaking through the glass surface. This means that more displacement in the bubble motion is possible, and desirable, so try the following settings in the Bubble Motion Tab:

Displacement: 22
Variation: 10%
Phase: 180

The Spin settings are again somewhat arbitrary, but for starters, try:

Time: 30
Variation: 40%
Phase: 50
Variation: 40%
Axis: random

Apply materials matching those applied to the first system (i.e. identical to the liquid filling the glass) and preview your animation.

It looks pretty good at this point, and has not been too difficult to set up. However, something is lacking. If you look carefully at an actual instance of what we're trying to recreate here, you'll note that a seltzer tablet starts to emit small bubbles well before it reaches the bottom of the glass and reaches the most active phase of the reaction. They're smaller bubbles, and less vigorous then those born from the main reaction that eventually supersedes them, but they're certainly there. To take care of this, we've created the third particle system in this scene.

Select the third particle system, make it's emitter visible, and preview the animation path assigned to it. Notice that it tracks approximately along the same path, at roughly the same speed, and is about the same size, as the tablet. While we could choose to make the tablet itself an emitter, and set up a Custom Surface system, creating a separate cylinder emitter (radius: 1.3 and height: 0.2) is a lot easier. With the tablet acting as a Custom Surface emitter, you would have to take care of the Plane and Axis settings in order to direct the bubbles upward rather than outward; following this route, you start off with an emitter that generates bubbles chiefly in the direction we want: toward the liquid's surface.

Again, set the particle type to sphere, and set a size of 0.3, with a smidgen of variation, say on the order of 5%. Go to the Motion tab, and let's key in the following values as a starting point:

Emit start: 50
Emit stop: 85
Display until 500
Life: 50
Rate: 3
Speed 2.2

Now, take a moment to analyze the nature of the event we're setting up here. The tablet is dropped into the liquid and the emitter we've set up starts to emit bubbles in frame 50, at a point where the tablet has reached a reasonable depth below the surface of the liquid. Without some careful attention to the life and speed of the particles we've now created with this system, we can end up with one of two problems. If we set an appropriate constant life for the particles generated by this system so that the bubbles emitted at the start of its active track will coincide with the surface and not go beyond it, those emitted at the end of its movement will fall far short of reaching the surface. Or, conversely, if the life is set such that those bubbles emitted at the end of the emitter's movement reach the surface properly, those emitted at the beginning of the movement will end their lives far above the surface.

The perfect solution, or the solution for the meticulous, would be to toggle off "Make Constant" for these variables and key frame appropriate settings for them over the length of the emitter's path. But this is a tutorial, and not a two year degree plan, so we'll take a short cut that relies on the action of the second particle system to mask the gap in the lower third of the glass, and the grow and fade functions to make sure the bubbles do not exceed the surface unnaturally at any point along the emitter's path. Enter the following values in the Particle Type tab:

Grow for:20
Variation: .5%
Fade for: 27
Variation: 5%

Of course, some tweaking to taste will likely be necessary in these settings, but this should at least land you in the right neighborhood.

Again, some Spin and Bubble Motion are appropriate for this system, and the following settings in the appropriate Tab are good starting points:

Spin time = 30
Spine axis= random
Displacement: .5
Variation: 4%
Cycle: default
Variation: .5%
Phase: 30
Variation: 5%

Preview your animation, and vary any of the above settings to increase or decrease the turbulence and effectiveness of the actions.

6.6.5 trueParticles Command Reference

The Make Constant Button

This button, found at the bottom of the trueParticles dialog, controls whether changes to the trueParticles settings are animated over time, or constant over time. While it is ON, setting a value for any control sets the control to that value for the entire length of the simulation. If you had animation keys set for this control, they're gone now. While the button is OFF, setting a value for an animatable control edits the keyframe value for that frame only. Be careful with this control, as you can generate a lot of extra work for yourself by typing stuff in one mode while you think you're in the other. Controls that cannot be animated, ever, appear slightly recessed into the dialog box, while controls that will animate when Make Constant is off are not recessed.

6.6.6 The Mode Tabs

Mode

Emitter Mode Group

- **Point:** Particles are emitted from a single point in the emitter icon. Where they go from there depends on your settings in the Particle Motion groups.
- **Box/Sphere/Cylinder:** Particles are created at random locations within the volume of the emitter box/sphere/cylinder.
- **Custom Fill:** You pick an object with the Pick Object button, and particles are created at random locations within the volume of an object.
- **Custom Surface:** This works much like Custom Fill, except that particles are created across the surface of the picked object.
- **Fragmentation:** This is pretty complicated. You pick an object, like with the other custom modes. The particle system creates a particle copy of the object over the actual picked object, like a second skin. Then, it shatters the particle "skin" into little pieces. Please note that the Fragmentation option overrides any choices you make in the Particle Type group. The controls in the Fragmentation area are relevant for adjusting the thickness and number of chunks.

Custom Object Group (Custom Fill, Custom Surface, and Fragmentation modes only)

- **Pick Object Button:** When you're using Custom Fill, Custom Surface, or Fragmentation modes, use the Pick Object button to choose an object from the scene. This object will be the particle emitter.

Custom Surface Group (Custom Surface mode only)

- **Over Entire Surface/At all Vertices:** When emitting particles over a custom surface, these radio buttons are used to decide whether particles appear over the entire surface of the picked object or only from its vertices.

Fragmentation Group (Fragmentation mode only)

- **Thickness:** This controls how thick, in standard trueSpace units, each object fragment is. A value of zero creates two-dimensional faces with no thickness, like pieces of tinfoil. Higher values create thick shards like bits of broken pottery.
- **All Faces/Number of Chunks:** Here you set how the object is chunked up. All Faces creates one shard for each face in the picked object. Number of Chunks lets you enter a value in the Minimum line, and the program attempts to divide the object up into that many pieces. It may have to generate slightly more pieces than you enter, if the faces of the object are oddly shaped.

Motion

Particle Timing Group

- **Emit Start:** This is the first frame of the simulation in which particles are emitted.
- **Emit Stop:** This is the last frame of the simulation in which particles are emitted.
- **Display Until:** This is the last frame of the simulation in which particles are visible. No particles are displayed or rendered after this frame. Even if there are still "live" particles kicking around in the scene or if particles continue to be emitted, you won't see them.
- **Life:** Each particle remains in the scene for this many frames after its birth. Once its life is up, the particle dies (goes away), even if Display Until indicates that particles are still visible.
- **Variation:** This variation applies to Particle Life. See Definition of Terms for a discussion of Variations.
- **Rate:** Here you set the number of particles emitted per frame of the simulation. This many particles will be emitted for every frame between Emit Start and Emit Stop (inclusive).
- **Discrete Birth Intervals:** Appears as an icon with a solid or dashed line. While this option is disabled (this is the default), tP uses a process known as subframe sampling to "smear" the particle generation times across the whole frame, producing a smooth emission pattern over many frames instead of a series of lumps. When you choose discrete intervals (the dashed line), particles are emitted in chunks every frame.

There is no subframe sampling to correct for the chunking produced when the emitter is rotating. However, you can often conceal this limitation by adding a Speed Variation of 10% (or slightly more if the particles are moving very slowly).

Particle Motion Group

- **Speed:** Each particle travels this many units per frame.
- **Variation:** This Variation is applied to the particle Speed. See Definition of Terms for a discussion on Variations. Note that variation is per particle, not per frame; particles will be assigned a random speed that remains constant throughout the life of the particle.
- **Direction of Motion (Off Axis, Off Axis Spread, Off Plane, Off Plane Spread)**

These controls establish the direction of travel for the particles. This is difficult for most people to visualize, so I'm going to explain it through metaphor and example. Open up a fresh trueSpace scene, load a tP emitter, and advance to frame 20. You should still be working with the default point emitter. Notice the trapezoid-shaped plane that makes up part of the emitter, and the arrow pointing up through the center of it. Right now, the particles are traveling in a straight line along the direction of the arrow. You can use the **Off Axis** control to sweep the particles direction-of-travel (DOT) around the clock face. From your current view, positive values spin the hand counterclockwise, while negative values turn it clockwise. **Off Axis** is measured in degrees, and accepts values between +180 and -180. **Off Plane** (also measured in degrees) allows you to take that minute hand and rotate it out of the plane of the clock face. Set **Off Axis** to 90, and examine the particle stream with **Off Plane** values of 0, 45, 90, 135, and 180.

Both **Spread** controls operate in a similar manner. They fan out the particle spray in the plane of the **Off Axis** or **Off Plane** setting with which they're associated.

Uniqueness Group

- **Seed:** All of the random arrangements of particles resulting from variations are controlled by the uniqueness seed. Selecting a new seed value (any number, it doesn't matter) creates a new random arrangement of particles. To network render across many machines, you MUST set the uniqueness seed to the same value for each machine. Pressing the NEW button generates a new random seed.

Icon Group

- **Emitter Is Visible:** Appears as a small eye image. Unselect the eye to hide the emitter icon (the eye greys out when you click it).
- **Various Icon Size Controls:** These controls set the dimensions of the icon. Which controls you use depends on which Emitter Type you're using (see the description of the Emitter Type Group).
  Point:- Emitter Scale scales the icon in all three dimensions.
  Box: Length, Width, and Height adjust the three dimensions of the box.
  Sphere: Radius controls the radius.
  Cylinder: Radius defines the radius of the end caps on the cylinder, while Height controls the Height of the body of the emitter.
  Custom Fill/Custom Surface/Fragmentation: You get the generic point emitter, and the Emitter Scale control scales the icon. Be aware that in these modes particle are emitted from a picked object, and the emitter icon is really just a placeholder.

Type

Particle Type Group (not available in Fragmentation mode)

- **Shard:** Particles are single-sided equilateral triangles. You must enable double-sided rendering in order to see these particles correctly from all angles.
- **Star:** Particles are single-sided six-pointed stars. Can be used as simple snowflakes. You must enable double-sided rendering in order to see these particles correctly from all angles.
- **Box:** Particles are cubes. This calculates faster than using a picked object and choosing a cube.
- **Round Tile:** Each particle appears as a circle which always faces the viewer. Always. No matter how you rotate or spin these particles, they remain perfectly face-on to the camera. This makes them ideal for applying maps or other images. A number of interesting effects can be created in this fashion, including snow, smoke, steam, or distance-based flocking (put, say, an animation of a goose with a transparent background on each tile. Create a bunch of tiles. Instant low-res flock of geese with comparatively little render cost.).
- **Square Tile:** This works much like round tile, but the particles are square. **Note:** You'll need to choose Face Mapping in the Mapping Group to correctly map images onto round or square tiles.
- **Streak:** Streak creates triangles with long pointy tails. You can really milk this one for all it's worth by using motion blur and Aligned spin. Streaks of rain, flashes of laser fire or gunfire, stars flying by at warp speed - it's good for a lot of stuff.
- **Sphere:** Particles appear as spheres. This calculates faster than using a spherical picked object of corresponding detail level.
- **Metaparticles:** This uses metasurfacing to generate metaball particles. The controls in the

Metaparticles group are relevant to the blobbiness and coherency of the effect.

- **Picked Object:** Particles appear as copies of an object you pick from the scene using the Pick Object button in the Custom Object group.

Particle Size Group

The Size control operates differently from usual with this particle type: it becomes a multiplier. For ex - ample, a Size of 1.0 produces same-size copies of the picked object, while a Size of 3.5 produces copies 350% the size of the original.

- **Size:** Here you set the size of each particle, in trueSpace units. Size is a multiplier when used in conjunction with the Fragmentation or Picked Object emission methods.
- **Size Variation:** Variation causes Size to vary +/ the given percent from particle to particle. See Definition of Terms for a discussion of Variations.
- **Grow For:** When using Grow For, particles are created at zero size and grow to full size over a number of frames. Set the number of frames of growth here. A value of zero indicates particles are created at full size.
- **Fade For:** When using Fade For, particles begin shrinking late in their lifespan, and reach zero size just as they die. Set the number of frames over which they shrink here. A value of zero indicates that particles maintain their full size right up to the time that they die.
- **Variations for Grow and Fade:** Applies a variation to the Grow and Fade times of individual particles. See Definition of Terms for a discussion of Variations. **Note:** When using lights as particles, Grow and Fade affect the intensity of the illumination instead of the size of the particles.

Metaparticles Group (Metaparticle type only)

- **Tension:** Tension controls how strongly a particle wants to blob with its neighbors and how big the blobs are.
- **Variation:** Variation causes Tension to vary +/ the given percent from particle to particle. See Definition of Terms for a discussion of Variations.
- **Grid:** Grid determines how fine the resolution of the particle system is. Low values produce smooth, slow calculating meshes while high values produce meshes that calculate quickly but may appear blocky and unrealistic.

Particle Size also influences the metaparticle effect, since bigger particles make bigger blobs. If you don't get anything at first, odds are your particles are too far apart. Increase particle size or adjust the stream to place the particles closer together until the particles begin to clump.

Custom Object Group (Picked Object type only)

Use the Pick button to choose a custom particle type.

Mapping Group

- **Time - Applies UV mapping to the particle system like so:** U is always 0.5. V is based on the age of each particle. Particles that have just been born have a V of 0.0. V increases as a particle ages, until it reaches 1.0 at an age of "Time" frames or greater. This can make particles change color as they get older, for example.

- **Distance - Applies UV mapping to the particle system like so:** U is always 0.5. V is based on the distance from the emitter for each particle. Particles that are right on top of the emitter have a V of 0.0. V increases with distance from the emitter, until it reaches 1.0 at a distance of "Distance" units or greater. This can make particles change color as they get further from the emitter, for example.
- **Face (Box, Tile, and Sphere types only) - Applies UV mapping to each face of each particle.**

Spin

Spin and Speed Group

- **Spin Time:** This is the number of frames a spinning particle takes to execute one complete revolution. A Spin Time of 0 indicates that the particle does not spin.
- **Variation:** Applies a variation to the Spin Time of each particle. See Definition of Terms for a discussion of Variations.
- **Phase:** Particles spin about their own axis, rotating through 360 degrees of a circle. "Phase" controls where in that 360 degrees of rotation the particles begin (when they're born).
- **Variation:** Applies a variation to the Phase of each particle. See Definition of Terms for a discussion of Variations.

Use Phase of 180 degrees and Variation of 100% to create spin with all particles at a random phase.

Spin Axis Group

- **Random:** Random makes the axis of rotation completely random for each particle.
- **Aligned:** Aligned causes particles to spin around the axis of their direction of a travel, just like a bullet or a thrown football. When you're using Aligned, you can use the Stretch control and a motion blur map to create particle motion blur.
- **User:** The User option lets you specify the axis of rotation manually. Enter values in the X-, Y-, and Z-axes. Now, imagine you've created a straight line in world space between 0, 0, 0 and the coordinates you've just entered. All the particles will be pointed in the direction that line points. Then they'll spin about that axis as though it were their direction of travel (see "Aligned" spin axis, above)

Example: X=0, Y=0, and Z=1 creates a line that points straight up. Enter these values for User-Defined Axis, and all the particles will point straight up (no matter which way they're traveling) and spin about that axis. This is basically like the spinning of a top.

- **Variation:** Applies a variation to the user-defined axis of each particle (those numbers you just entered). See Definition of Terms for a discussion of Variations.

Bubble Motion

Bubble Motion Group

Bubble Motion is motion perpendicular to the direction of travel. Essentially, it makes particles wiggle from side to side as they move.

- **Displacement:** This determines the "wiggle distance". Displacement is the distance a particle moves out to the side before it starts moving back to center again. Then, it moves out to the other side by the same amount, back to center again, and starts the process over. Visualize the swinging of a pendulum.
- **Variation:** As usual, this provides a per-particle modification to the Amplitude. See Definition of Terms for a discussion of Variations.
- **Period:** This is the Period of time (number of frames) required for each particle to complete one complete bubble movement (described in Amplitude, above).
- **Variation:** As usual, this provides a per-particle modification to the Period. See Definition of Terms for a discussion of Variations.
- **Phase:** Phase controls where in that 360 degrees of motion the particles begin (when they're born). Entering a nonzero value starts particles off partway through a bubble movement.
- **Variation:** Applies a variation to the Phase of each particle. See Definition of Terms for a discussion of Variations.

Use Phase of 180 degrees and Variation of 100% to create spin with all particles at a random phase.

Menu

In the Menu tab, you can load and save particle system configuration presets or preset libraries, and create copies of the particle system itself.

To load a preset - select a name in the Saved Presets window and hit Load.

To save your current configuration as a preset - Type a name in the Preset Name slot and hit Save. Picked objects and materials assigned to the particle system cannot be saved.

To delete an existing preset - Select the name of the preset in the Saved Presets window, then hit Delete.

To switch to a different preset library - Hit Load File, browse to the preset library you want to use, and hit Open.

To create a new preset library - Choose the New File option, type in a file name, then hit Save.

To copy the particle system - Hit the copy Particle System button. A new copy of the particle system is created on top of the old. The new copy will correctly reference picked objects, and will have the same materials as the original. Use the Object Move tool to move the new particle system into place.

Presets marked with an asterisk are designed to use one of the referenced-object emitter modes. You'll need to create an object and pick it as the emitter to see the full effect of the preset. Try various shapes and moving or static objects to see how they alter the effect.

6.6.7 The Helper Tools

Physics

Enabling this option makes four edit boxes available: Weight, Elasticity, Resistance, and Friction. These four parameters can be used for setting basic physical attributes of the particles.

- **Weight:** The weight of the particle in kilograms.
- **Elasticity:** Elasticity is the ability of a particle to preserve its energy during collision. Its value ranges from 0 to 200. A value of 0 means that all of the particle's energy is lost during collision, while a value of 100 means that the particle preserves all energy during collision. Values over 100 mean that the particle gains energy during collision.
- **Resistance:** Resistance describes the ability of a particle to react with its environment (atmosphere, local environment streams, wind, etc.). A value of 0 means that no reaction occurs; the particle does not "feel" the environment. The higher the value, the stronger the particle's interaction with its environment.
- **Friction:** Friction describes the efficiency of frictional forces that affect particles when they come in contact with other objects or the ground. The frictional forces induce deceleration in moving particles. Values can range from 0 to 100. A value of 0 represents no friction, while a value of 100 means maximum friction.

If "Use Global Physics" is enabled, all physical settings are equally applied to each emitted particle, and all particle movement is computed during physical simulation.

To see the interaction of particles and physics:

1. Enable "Use Global Physics" (on the Helpers>Physics subpanel) to give the particles physical properties.
2. Run a physical simulation. The particles will emit and interact with other objects in the scene.
3. Once the simulation is completed, press Play to see the result.

Other notes:

- When "Use Global Physics" is enabled for an emitter, the helper subpanels such as Wind, Gravity, Path, and Bounce cannot be used.
- When physics are applied, note that the MetaParticles will not respond when you run a simulation. They also cannot be switched with another type of particle after a simulation has been run. Use the "Bounce" helper with standard emitters to compensate and simulate the effect.
- Physics cannot be used with the Fragmentation emitter type.
- To improve the collision of particles with thin objects, place thicker objects in the same locations and make them invisible.

Until Cebas provides an updated version of PyroCluster be aware of the following:

1. Emitters, both standard and physical must be at 0,0,0 when you apply the PC effect. Once the effect is applied (by dragging the effect onto the emitter), you can move the emitter to any location you want, and the effects will still match the position of the particles.
2. For best results, run any physical simulation first before applying the PC effects.

Bounce

- **Use Bounce:** This turns on the Bounce Tool
- **Pick Reflector Object/Reflector:** Here you pick a reflector object from which particles will rebound. The Reflector: line indicates the name of your picked reflector.
- **Bounciness:** This is the amount of rebound on particles as they come off the reflector objects. Use a low percent for squishy collisions where most of the speed is lost, and a higher percent for more inelastic collisions. Values greater than 100% create a "flubber" effect in which the particles hit the object and gain speed.
- **Variation:** A Variation applied to Bounce. See Definition of Terms for a discussion of Variations.
- **Chaos:** This randomizes the direction in which particles rebound so that particles striking at the same speed and angle will actually bounce off in different directions. Greater Chaos values produce greater variability in the direction of bounce.

The bounce tool is a very capable helper, but it does have limits. If a given particle strikes the same face of a reflector object more than once per frame, it is possible that Bounce may fail to detect one of the collisions, and the particle may slip through the reflector object. Try to avoid setting up a scene in which particles impact the reflector object in this fashion.

Extremely complex objects can require a lot of calculation time. If you want to bounce particles off a very complex object, consider creating a simplified, nonrenderable version of the object over-top of the original object and using the simplified version as a reflector stand-in.

Blast

Blast creates an invisible force which propels particles away from the point of the explosion.

- **Blast Type Group**
  - **Use Blast:** Activate the Blast tool.
  - **Start Time:** This is the frame on which the explosion effect begins.
  - **End Time:** On this frame, the explosion effect stops.
  - **Strength:** Here, you set the strength of the blast. In Wind and Gravity, you set the strength of the effect and its direction using the same controls. Explosions work differently because of the different shape of the effect. Set the strength here with the one spinner, and you'll set direction elsewhere.
  - **Impact Chaos:** This randomizes the direction in which particles are propelled when struck by the blast. As with most Chaos parameters throughout this plugin, the goal here is to make the primary effect (the Blast) look "messy".
  - **Range:** This is the distance from the center point of the explosion at which the blast still has

full strength. Beyond this range, the blast gets weaker and eventually dies out altogether.

- **Falloff Type**
  - **None:** This indicates an explosion with unlimited range. The explosion has full strength out to the furthest reaches of the scene. Choosing this Falloff Type disables Range.
  - **Linear:** Linear explosions have full effect out to a certain distance (set with the Range parameter), and then gradually fade down to nothing. Linear falloff means that the explosion loses force gradually with increasing distance beyond its range until it dies.
  - **Exponential:** This is a limited-range explosion type, similar to the Linear Falloff selection. Exponential falloff means that the explosion keeps most of its force for a good distance beyond the range, and then dies out suddenly.
- **Blast Shape:** Here you set the shape of the blast as spherical, cylindrical, or planar. Choosing the Spherical blast shape disables the Blast Orientation controls, as a spherical explosion does not have a meaningful front or back.

- **Blast Location Group**
  - **Blast Center (X, Y, Z):** Here you set the world coordinates for the location of the center of the blast.
  - **Blast Orientation (X, Y, Z):** For Cylindrical Blasts, you're setting the direction of the top of the cylinder. For Planar blasts, you're setting the direction in which the plane is facing.

Path Follow

Path Follow's method of particle capture can best be understood using a concept called the Capture Bubble. The capture bubble is an imaginary sphere that exists from Capture Start to Capture End. It is centered on the first vertex of your chosen path, and its radius is either infinite (if you select Unlimited Capture Range) or established by Capture Range. All particles that are inside the capture bubble, or that move into it, are captured by Path Follow.

For each captured particle, Path Follow creates an imaginary copy of the path, with the first vertex of the path positioned where the particle is. It then proceeds to push the particle along the path according to your settings.

If your particles are traveling in a beam (as often happens in the Point Source mode), then they'll proceed along in a grand conga line, one after the other. They have to, because they're all using the same imaginary path, and the path has no thickness, only length. You can disperse such a particle beam by increasing the Path Radius. This turns the spaghetti string of the path into more of a sausage shape.

- **Choose Path Group**
  - **Use Pathfollow:** Check this box to enable the pathfollow motion controller.
  - **Pick Path:** Choose a spline to use as the particle path.
  - **% of Path Used:** Particles don't have to follow the entire length of the path. By setting this percentage, you can have them traverse only a portion of your picked path.
- **Capture Particles Group**
  - **Unlimited Capture Range:** Here we're establishing how Path Follow grabs particles when its capture time begins. Imagine there's a bubble centered around the first vertex in the picked spline path. This bubble appears when the capture time begins, and all particles

that enter the bubble are "grabbed" and pushed along the path. If you check the Unlimited Capture Range check box, the bubble is infinitely large; it captures every particle emitted by this tP emitter, no matter where it is in the scene. If you clear this box, Capture Range becomes the radius of the bubble.
    - **Capture Range:** This is the radius of the capture bubble if you don't use Unlimited Capture Range.
    - **Capture Start:** This is the frame at which capture begins.
    - **Capture End:** At this frame, Path Follow stops controlling the particle, and it just continues on in whatever direction it's going at the time.
- **Move Particles Group**
    - **Travel Time:** This sets the number of frames that each particle takes to traverse the path.
    - **Variation:** A Variation applied to Travel Time. See Definition of Terms for a discussion of Variations.
    - **Constant Speed:** By default, particles traverse each segment of the path in the same period of time. This means that particles will speed up when crossing long segments, and slow down when crossing short ones. To maintain a constant speed, check this box.
    - **Path Radius:** By default the spline path is a line. Particles line up in a big conga line as they follow the path, because the path is infinitely narrow - it has only length, with no thickness. Increase Path Radius and that line gains thickness, becoming a sausage shape. Path Radius is the radius of the sausage.
    - **Stream Taper:** Taper is designed to take particles that are dispersed at the time of capture and gradually pull them together as they traverse the path. You reduce the spread by as much as 99%.
    - **Variation:** A Variation applied to Stream Taper. See Definition of Terms for a discussion of Variations.
    - **Converge/Diverge/Both:** This applies to Stream Taper. The basic function of Taper occurs when you make particles converge. Select Diverge to reverse the effect and cause the particles to spread out as they follow the path. Select Both to have some particles converge while others diverge.
    - **Stream Swirl:** Swirl causes particles to spin in a ring about the path as they travel along it. Can't visualize it? Think back to high school science class and astronomy. The earth moves around the sun in a ring. The ring is the path. The earth is where a particle would be if you didn't use Stream Swirl. Got that? Now, as the earth follows its path, the moon spins around the earth. At the same time, the moon moves with the earth in its path around the sun. The moon is the particle's actual position once Stream Swirl has been applied. The end result is that the particle spirals along the path in a corkscrew fashion.
    - **Variation:** A Variation applied to Stream Taper. See Definition of Terms for a discussion of Variations.
    - **CW/CCW/Both:** Here you decide whether Stream Swirl causes particles to swirl in a Clockwise or Counterclockwise direction. The Both option has predictable results.

Gravity

- **Gravity Group**
- ◦ **Use Gravity:** Activate the Gravity tool.

- **X, Y, Z:** Set a fall speed value for each of the three axes. Gravity pulls particles in that direction at that speed. This is a constant force, which means that the particles will accelerate in this direction over time.

The Gravity helper is a continuous accelerator, which means that particles will accelerate without limit over time. There is no terminal velocity.

Wind

- **Wind Group**
  - **Use Wind:** Activate the Wind tool.
  - **X, Y, Z:** Set a wind speed value for each of the three axes. Wind pulls particles in that direction at that speed. This is a constant force, which means that the particles will accelerate in this direction over time.

The Wind helper is a limited accelerator, which means that particles will be accelerated up to the speed of the wind propulsion, but no further.

6.7 Physical Simulation

Physical Simulation in trueSpace allows you to simulate the realistic behavior of objects according to the laws of physics. You can create real-time realistic movement of objects in 3D space, taking into account the presence of gravity and the forces originating from the interactions between objects (collisions). You can directly interact with those objects during the simulation as well.