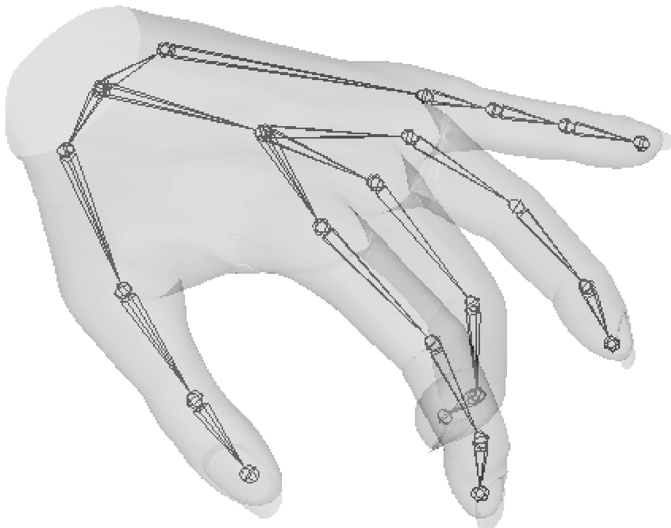


# Character Setup

Version 6

Cover image courtesy of The Art of Maya, Alias Systems.



ALIAS ■ 210 KING STREET EAST ■ TORONTO, CANADA M5A 1J7

# Table of Contents

## 6 Character Sets . . . . . . . . . . 513

# 1 Character Setup overview

## About

### Setting up the components of your scene for animation

Before animating the characters and objects in your scene, you set up your scene by *rigging* all your characters and by applying the appropriate constraints and deformers to all the objects you want to animate.

Rigging a character, also known as *character setup*, involves creating skeletons and IK handles for your characters, binding skins to the skeletons, and setting up deformers and constraints. You can also create deformers for your character and animate them to produce effects; for example, the jiggling belly (jiggle deformer), furrowing brow (wire deformer), and flexing biceps (lattice deformer) of a sumo wrestler model.

Non-character objects are also very important to bringing your scene to life. You can limit and control the transformations of objects by constraining them to characters or other models in your scene. You can also create deformers for objects to create complex deformation effects. For example, you can apply a squash deformer to the model of a ball and then parent constrain the ball to the hands of a character. With this setup, you can key the weights of the character's hands and the squash deformer's attributes to create an animation of the character bouncing the ball from hand to hand while the ball squashes on the ground and stretches as it rises back into the air.

In addition to setting up characters and objects for animation, you can set up dynamics for animation. You can constrain dynamic objects such as particle emitters, fields, and fluids to objects or characters in your scene. For more information, see the *Dynamics* and *Fluid Effects* guides.

The following sections provide a brief overview of the chapters in the *Character Setup* guide:


- ″Using skeletons″ on page 25
- ″Skinning your character″ on page 26
- ″Creating deformation effects″ on page 27
- ″Constraining objects″ on page 28


### Using skeletons

Skeletons are the underlining joint and bone hierarchies that let you animate your characters. Every skeleton has several parent joints and child joints, and one root joint. Parent joints are joints that have joints below them in the skeleton's hierarchy. For example, an elbow is the parent of the wrist and the child of the shoulder. The root joint is the first or top joint in a skeleton's hierarchy. For more information on setting up skeletons, see "Skeletons" on page 33.

You can use the following methods to animate a skeleton: forward kinematics (FK), inverse kinematics (IK), or IK/FK blending.

### Forward Kinematics

With forward kinematics, also known as *FK*, you transform and key joints directly, rather than using an IK handle to animate a skeleton.

Forward kinematics are useful for creating detailed arcing movements, but not very intuitive for goal-directed movements. For example, you can easily use FK to animate the rotation of an arm at the shoulder joint, but not the arm reaching for a glass. For more information on FK, see "Forward kinematics (FK)" on page 44.

### Inverse Kinematics

With inverse kinematics, also known as *IK*, you transform and key an IK handle to animate a skeleton. The IK handle is drawn as a straight line between the start and end joints of its IK chain. The effect the IK handle has on the joint chain depends on the type of IK solver used by the IK handle. See "IK solvers" on page 47.

Inverse kinematics are useful for goal-directed movements. For example, you can use IK to animate an arm reaching for a glass of water, but not for specific movements at individual joints. For more information on IK, see "Inverse kinematics (IK)" on page 45.

### Blending IK and FK

Instead of using only FK or IK to pose and animate a joint chain, you can use both FK and IK on the same joint chain. The Ik Blend attribute on the ikHandle lets you apply FK and IK animation to the same joints. Ik Blend specifies the amount of influence (weight) that FK or IK have over the animation of the joints.

Blending IK and FK is useful for posing complex characters that have a wide range of movements in their animations. For example, you can use IK to animate the directed motion of a character's arms, and you can use FK to animate the rotation of the shoulder, elbow, and wrist joints in the arm. For more information, see "IK/FK blending" on page 55.

## Skinning your character

Skinning is the process of binding a modeled surface to a skeleton. You can bind any model to its skeleton using skinning, or you can model over a pre-existing skeleton to create its skin. When a model is bound to a skeleton using skinning, it then follows or reacts to the transformations of the skeleton's joints and bones. For example, if you bind a model's arm to its underlying skeleton using skinning, rotating the elbow joints causes the skin at the elbow to crease and pucker.

There are three types of skinning in Maya: smooth skinning, rigid skinning, and indirect skinning.

### Smooth Skinning

With smooth skinning, you can create smooth, articulated deformation effects. Smooth skinning specifies that multiple joints and other influence objects can have varying influences on the same points (CVs, vertices, or lattice points) on a model. For more information on smooth skinning, see "Smooth skinning" on page 144.

### Rigid Skinning

With rigid skinning, you can create stiff, articulated deformation effects. Rigid skinning specifies that only individual joints can influence each CV, vertex, or lattice point on a model. For more information on rigid skinning, see "Rigid Skinning" on page 160.

### Indirect skinning

With indirect skinning, you can bind lattice or wrap deformers as skins to a skeleton. When a character is indirectly skinned, posing its skeleton causes the bound deformers to transform the model's skin. For more information on indirect skinning, see "Lattice deformers" on page 243 and "Wrap deformers" on page 279.

## Creating deformation effects

You can add deformation effects to your characters and objects to enhance their animations. Deformers are tools that let you transform or animate objects in ways that simple manipulation and keyframes cannot. Deformers have two main applications: to model surfaces or to add extra shape animations to an object.

### Deformers as modeling tools

You can use deformers as modeling tools. You can create a deformer, for example a sculpt deformer tool, tweak the model's shape with it, and then delete the history from your object when the deformer is no longer needed. When you delete the object's history, you delete the deformer and retain the object's deformed shape.

### Deformers as animation tools

You can use deformers as animation tools. You can create a deformer, tweak the target object with the deformer, and then key the deformer's attributes over time to produce an animation. For example, you can create a blend shape deformer for a model of a face. Then over time, you can manipulate and key the sliders for the deformer in the Blend Shape editor to create an animation.

## Constraining objects

With constraints, you can drive the position, orientation, and scale of one object with the transformation settings of another object. The object that is driven is called the *constrained* object, and the driver object is called the *target* object. The specific channels that are driven by a constraint depends on the type of constraint. For example, for an object constrained by a point constraint, only its X, Y, and/or Z translations are driven by its target objects. When a constraint relationship has more than one target object, *weights* are used to determine the amount of influence each object has on the constrained object. For more information on constraints, see ”Introducing Constraints” on page 425.

**Note** You can apply constraints and animation to the same objects. See ”Animation-Constraint blending” on page 444.

The following constraint types are available: point, orient, scale, aim, parent, geometry, tangent, and pole vector.

### Point constraints

Point constraints limit and control only the translation channels of the constrained object. Point constraints are useful when you want to constrain the position of one object to that of another without parenting. For example, you can use a point constraint to constrain the model of a crate to an animated train model and to the model of a crane that lifts the crate on and off the train. In this example, you can key the targets (the train and the crane) weights to determine which model at what time in the animation controls the translation of the crate. For more information on point constraints, see ”Point constraints” on page 428.

### Orient constraints

Orient constraints limit and control only the rotation channels of the constrained object. Orient constraints are useful when you want to constrain the orientation of one object to that of another. For example, you can use an orient constraint to constrain the blades of one windmill to those of another. In this example, when the target windmill’s blades turn around their axis, the constrained windmill’s blades rotate around their own local axis. For more information on orient constraints, see ”Orient constraints” on page 432.

### Parent constraints

Parent constraints cause the constrained object to inherit the transformations and global orientation of its target objects, mimicking a parent-child relationship. For example, you can constrain the model of a hat to the head and hands of a character with a parent constraint, so that when the head nods and rotates side to side, the hat follows the head's movements. And when the hand grabs the hat and lifts it off the head, the hat follows the hand. In this example, setting and keying the target weights lets you anchor in time the amount of influence the head and hands have on the hat. For more information on parent constraints, see "Parent constraints" on page 433.

### Scale constraints

Scale constraints limit and control the scaling channels of the constrained object. Scale constraints are useful when you want the size of one object to drive that of another object. For example, you can constrain the models of blades of grass to each other, so that when they appear to grow during their animation, the size of each blade of grass increases by the same amount. For more information on scale constraints, see "Scale constraints" on page 433.

### Aim constraints

Aim constraints limit and control the rotation channels and aim vector of the constrained object. The aim vector is an attribute on the aim constraint that forces the constrained object to always point at the target objects. Aim constraints are useful when you want the constrained object to always follow and point at the target objects. For example, you can constrain the eyes of a character to track the movements of another character in your scene. For more information on aim constraints, see "Aim constraints" on page 429.

### Geometry constraints

Geometry constraints constrain or bind the constrained object so that it follows the target curve or surface as it changes shape. Geometry constraints are useful when you want to attach one object to the surface of another without using more complex methods such as MEL or expressions. For example, you can bind a virus model to the surface of a cell model with a geometry constraint. For more information on geometry constraints, see "Geometry constraints" on page 434.

### Normal constraints

Normal constraints limit and control the orientation of the constrained object so that it aligns with the normal vectors of the target object's surface. Normal constraints are useful when you want an object to travel across a surface. Typically, you use normal constraints in conjunction with geometry constraints. For example, you can use a normal constraint and a geometry constraint to properly constrain a button on to a shirt. For more information on normal constraints, see "Normal constraints" on page 436.

### Tangent constraints

Tangent constraints limit and control the orientation of the constrained object so that the constrained object is forced to point in the direction of the tangent at its current location (point) on the curve. Typically, you use tangent constraints in conjunction with geometry constraints. For example, you can use a tangent and a geometry constraint to attach the model of a roller coaster car to roller coaster tracks. During the animation, the car follows the shape and tangents of the track. For more information on tangent constraints, see "Tangent constraints" on page 439.

### Pole Vector constraints

Pole Vector constraints cause the ends of pole vectors to move to and follow the position of an object, or the average position of several objects. The pole vector is a component of the IK rotate plane handle that determines where you get flipping when the IK handle crosses the pole vector. Pole Vector constraints are useful because they let you control flipping and the position of joints (for example, the elbow) in an IK joint chain. For more information on pole vector constraints, see "Pole Vector constraints" on page 443.

### Animation-constraint blends

You can apply animation and constraints to the same object. When you keyframe a constrained object or assign a constraint to a keyframed object, a pairBlend attribute is automatically added to the object. You can set and key the pairBlend attribute to animate the animation-constraint blend weight. The blend weight determines the amount of influence the animation and constraints have on the constrained object. For example, you can constrain a ball to the hands of two characters and key the hand weights. When the ball is thrown from one character to another, you can then keyframe the ball's flight through the air. The process of applying animation and constraints to the same object and then keying the blend weight is called animation-constraint blending. For more information on animation-constraint blending, see "Animation-Constraint blending" on page 444.

## Character sets

Character sets let you group attributes from multiple objects into a single node that you can then select and key. Character sets are useful when you want to animate multiple attributes from many different objects all at the same time. For example, you can use a character set to bring together in one node all the joint rotations of a character's legs. For more information on character sets, see "Character sets" on page 513.

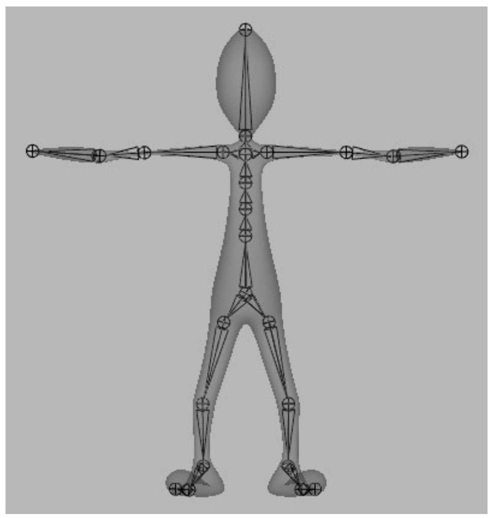
# 2 Skeletons

## About

### What are skeletons?

#### Skeletons

Skeletons are hierarchical, articulated structures that let you pose and animate bound models. A skeleton provides a deformable model with the same underlying structure as the human skeleton gives the human body.



Character created by Patrice Paradise and Olivier Georges

Just like in the human body, the location of joints and the number of joints you add to a skeleton determine how the skeleton's bound model or 'body' moves. When you bind a model to a skeleton, it is called *skinning*. For more information on the process of skinning, see the "Understanding skinning" on page 139.

### Skeleton components

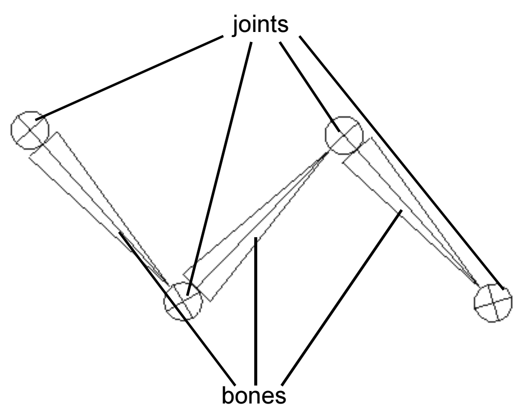
#### Skeleton components

A skeleton is composed of a series of joints and bones that form *joint chains.* You can pose and animate these joint chains using *IK handles* or by translating or rotating individual joints (FK). For more information on posing skeletons, see "Posing skeletons" on page 43.

# Joints and bones

Joints are the building blocks of skeletons and their points of articulation. Also, joints have no shape and therefore can not be rendered. Each joint can have one or more bone attached to it, and more than one child joint. Joints let you transform a skeleton when posing and animating a bound model.

Bones do not have nodes, and they do not have a physical or calculable presence in your scene. Bones are only visual cues that illustrate the relationships between joints.




## Related topics

- ”To create a joint or joint chain” on page 57
- ”Skeleton > Joint Tool” on page 103
- ”Set display size of all joints” on page 90
- ”Display bones as boxes” on page 91
- ”Display a joint’s local axis” on page 90
- ”Display all the local axes in a skeleton” on page 90

## Anatomical joint types

Various joint attributes determine how joints behave. By adjusting the attributes of a joint, you can limit how far a joint can rotate or restrict what planes it can rotate about. For example, to create a joint chain that moves and rotates like the human neck, you would set the degrees of freedom for the joints so that they rotate about only 2 axes and then you would limit each joints range of rotation. See ”Degrees of freedom” on page 41 and ”Limits” on page 41.

### Ball joint

A ball joint is a joint that can rotate about all three of its local axes. For example, the human shoulder is a ball joint.

### Universal joint

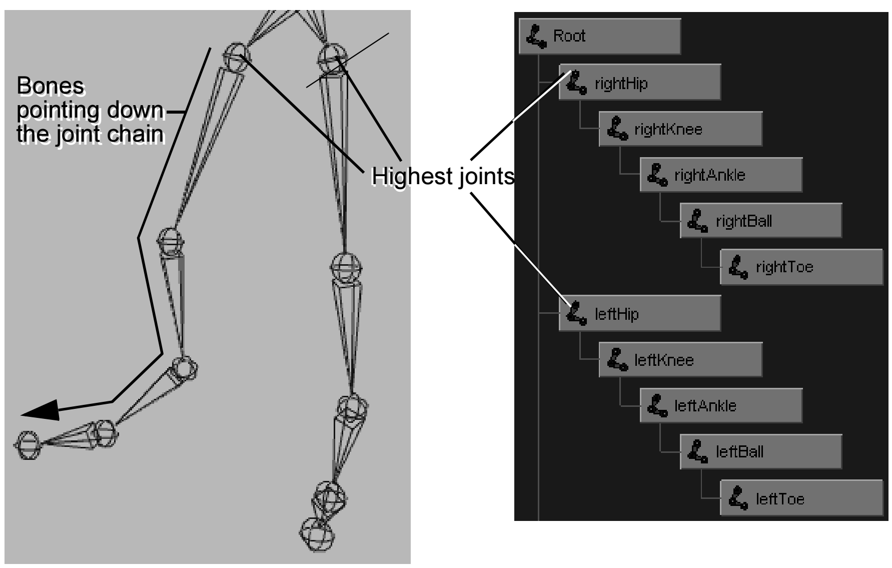
A universal joint is a joint that can rotate about only two of its local axes. The human wrist is a good example of a universal joint, though the wrist has limitations on the extent it can rotate.

### Hinge joint

A hinge joint is a joint that can rotate about only one of its local axes. For example, the human knee is a hinge joint.

## Joint chains

A joint chain is any group of joints and their bones connected in a series. Joints are connected linearly, and their paths are drawn on screen by the bones. A joint chain begins at the *highest joint* in the chain's hierarchy, and its bones are drawn pointing down the chain.

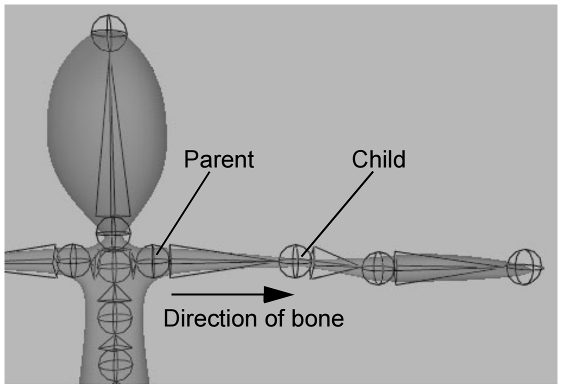



Scene view Hypergraph

When you create a joint chain, the position of each joint should approximate the anatomical positions of the joints of the character you are trying to create. For example, if you are creating joint chains for the skeleton of a human biped model, you should place the joints at points of articulation such as the elbows, knees, ankles and so on.

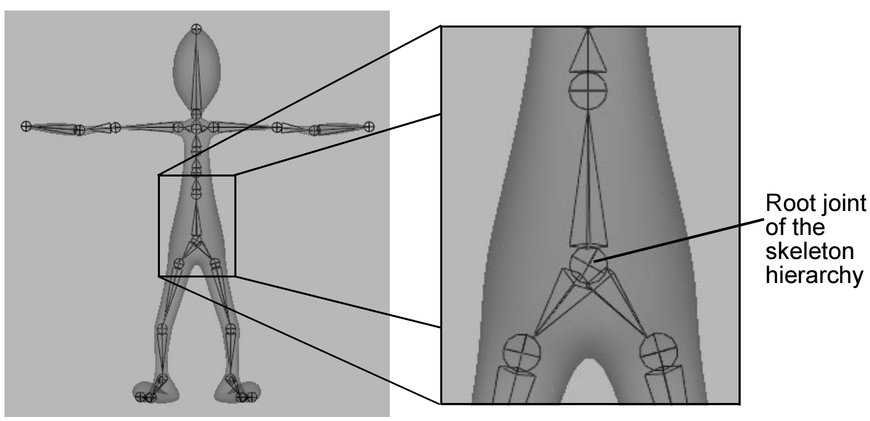
## Skeleton hierarchy

A *skeleton hierarchy* is composed of a series of joints and joints chains with hierarchical relationships. Each joint in a skeleton hierarchy is a *child* joint and a *parent* joint.




A parent joint is any joint higher in a skeleton's hierarchy than any of the other joints that are influenced by that joint's actions. Joints below a parent joint in the skeleton hierarchy are called child joints. Joints at the top of bones are always parents, and joints at the bottom of bones are always children. Parent joints drive the transformations of their respective child joints. Thus, when you translate or rotate a parent joint, you also translate or rotate all of its child joints.

The *root* joint is the highest joint in a skeleton's hierarchy. A skeleton can have only one root joint. You can move and orient the entire skeleton in world space by translating and rotating the root joint.




### Related topics

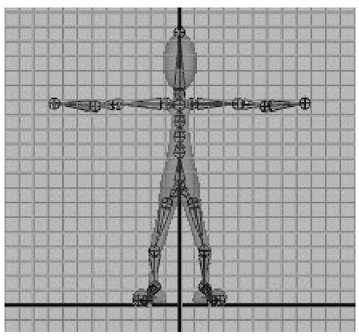
❖ "To reroot a skeleton" on page 65

## Building skeletons

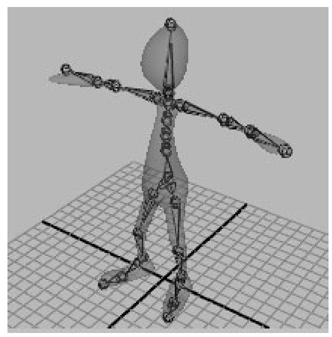
# Building skeletons

Building a skeleton is the process of placing and orienting joints to create a system with which you can pose a deformable object. Depending on the type of character you want to create, you need to place the joints where they approximate real joint positions. For example, if you are creating a skeleton for a human biped, you would place joints at all the major points of articulation in its model (shoulders, knees, elbows, spine and so on).

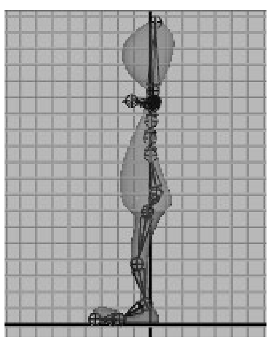
When constructing a skeleton, use multiple camera views to ensure that your skeleton fits its object appropriately in all three dimensions and use the display grid as a guide when judging the size and shape of the skeleton.



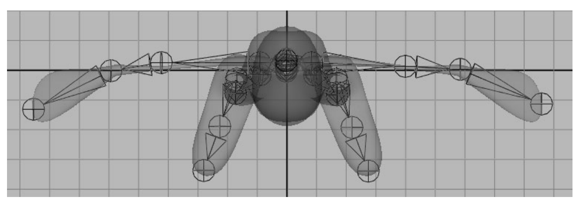
Front view



Perspective view



Side view



Top view

You can position and rescale the grid to suit your work and you can use snapping to align your joints with the grid. If you want to adjust the Maya snapping settings, go to Windows > Settings/Preferences > Snapping.

| Tip | You can use the Discrete Move setting from the Move tool as a virtual grid.<br>For more information on the Move Tool, see the Basics guide. |
|---|---|

Once you have created the skeleton, you can group or parent objects to joints and use the skeleton to control the objects movements. Skeleton hierarchies are the same as normal hierarchies, with the exception that they can have IK. Thus, skeletons and their joints can be viewed and navigated in the scene view, Outliner, and Hypergraph in the same way as any other hierarchy in Maya.

To complete the basic character set-up of your object, you need to apply a kinematics system to the joints of the skeleton and bind the object to its skeleton. See ”Posing skeletons” on page 43 and “Understanding skinning” in Chapter 3.

## Creating joint chains

When building a skeleton for your character, you first need to draw a joint chain using the Joint Tool. Once you create a joint chain, you can use the Insert Joint tool to continue the chain or add to it by creating new joint chains starting from any of the joint chain’s joints. In this way, you can create a complex structure of varying joint chains. These connected joint chains create a skeleton hierarchy.

You should insert all necessary joints before adding IK handles to your skeleton or skinning your character. Otherwise, if you insert joints into joint chains with IK handles, you will have to redo the IK handles. Also, inserting joints after skinning can lead to undesirable deformation effects.

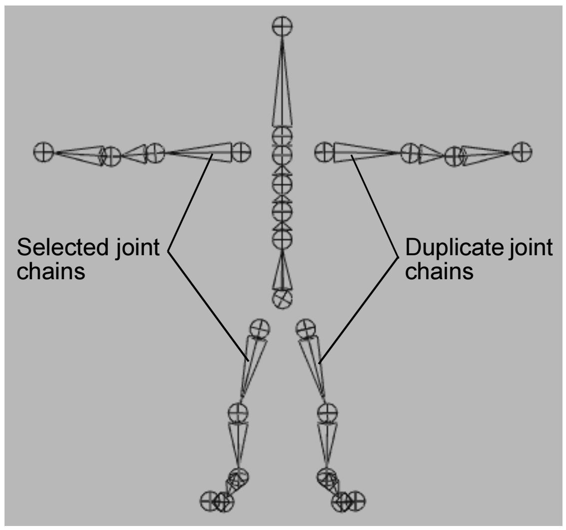
### Related topics

- ”To create a joint or joint chain” on page 57 and
- ”To place a joint in an existing joint chain” on page 58
- ”To remove a joint” on page 68
- ”Skeleton > Joint Tool” on page 103
- ”Skeleton > Insert Joint Tool” on page 109
- ”Skeleton > Remove Joint” on page 110

## Mirroring joint chains

You can duplicate or create *mirror copies* of joint chains using the Mirror Tool. A mirror copy is a copy that is symmetrical about a selected plane. In effect, the reflection of the original joint chain in the specified plane is turned into a real copy of the original with all the aspects of the joint chain mirrored accordingly. The origin of the plane is at the parent joint of the original joint chain. When mirroring joints, joint attributes and IK handles are mirrored as well as the joints and their bones. Other skeleton data such as constraints, connections, and expressions are not included in the mirrored copy.

Mirroring is useful when creating the 'limbs' a character. For example, you can build a right arm and hand, and then create a mirrored copy of it for the left arm and hand.




Mirrored across YZ

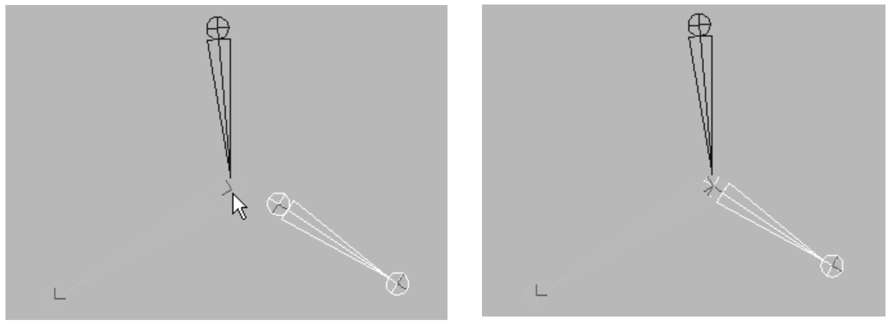
Mirroring affects all aspects of the creation of the left arm, including the joint limits. You do not have to reset the joint limits so that the left arm's joint limits are symmetrical to the right arm's joint limits—Maya does this for you.

### Related topics

❖ "To mirror a joint chain" on page 62
❖ "Skeleton > Mirror Joint" on page 111

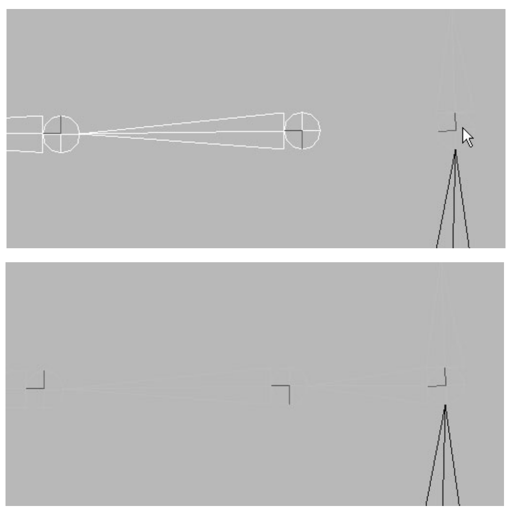
## Connecting joint chains and skeletons

In addition to creating and extending joint chains, you can create a complete skeleton for your character by connecting joint chains or skeletons. You can do this by combining a joint from a joint chain or the root of a skeleton with any joint of another joint chain or skeleton (except its root joint). The joint chain or skeleton that becomes part of the other joint chain or skeleton changes its position in the scene so that it connects directly with the other skeleton's joint.



Connect by combining joints (Connect mode)

You can also connect the joint of one joint chain or the root of one skeleton to any joint of another joint chain or skeleton by extending a bone from a joint on the joint chain or the root from the skeleton to a joint or the root joint of another joint chain or skeleton. The joint chain or skeleton that becomes part of the other joint chain or skeleton does not move to the other chain's or skeleton's position.



Connect by creating a bone (Parent mode)

## Related topics

- ❖ "To connect joint chains by combining joints" on page 59
- ❖ "To connect joint chains by connecting selected joints with a bone" on page 60
- ❖ "To disconnect joints" on page 67
- ❖ "Skeleton > Connect Joint" on page 110
- ❖ "Skeleton > Disconnect Joint" on page 110

# Setting up joints for posing and animation

## Limits

Joint limit information specifies the minimum and maximum translation, rotation, and scaling values for a joint. You can set a joint's limits if you want the joint to behave in a specific manner. For example, if you want to create an elbow joint for a human skeleton, you need to set its rotation limits so that it cannot rotate in X and Z, but can rotate only a specified amount in Y. Alternately, you can turn off transformation channels or use Set Driven keys to restrict the transformations of an object. See *Set Driven Keys* in the Animation guide.

**Note** It is only useful to set joint limits when you are planning to use FK to pose your joints. See "Forward kinematics (FK)" on page 44.

### Related topics

- "Skeleton > Joint Tool" on page 103
- "General skeleton and IK node attributes" on page 122
- "joint" on page 123

## Degrees of freedom

If you are planning to use an inverse kinematics (IK) to pose your joints, you may want to set the degrees of freedom for your joints. The degrees of freedom setting determines which local axes the joints can rotate around during IK posing and animation. You can set the degrees of freedom for a joint from the Joint section of the joint node attributes.

### Related topics

- "Limits" on page 41
- "joint" on page 123
- "Inverse kinematics (IK)" on page 45

## Damping and stiffness

You can set the damping and stiffness for joints that are influenced by IK. The damping and stiffness values let you specify how joints behave when posed by an IK handle.

Joint *damping* applies resistance to a joint as it approaches its joint limits. Instead of the joint abruptly stopping when it reaches its limits, you can use damping to slow it down smoothly. Depending on the strength and range you set, a joint with dampening will not reach its limit boundary unless forced. For most living creatures, when a joint rotates as far as it can, it tends to slow down or *dampen* before reaching its limit. For example, an elbow does not snap straight, but gradually slows down as the lower arm aligns with the upper arm. In animation terminology, this type of effect is called an *ease-in*.

Joint *stiffness* specifies a joint's resistance to rotation during inverse kinematics (IK) posing. Set the Stiffness value for a joint only if you want certain joints in a joint chain controlled by an IK handle to rotate less freely than others. For example, you can set the Stiffness of joints in the mid-back of a human character so that they rotate less freely than those in the lower back. The higher the stiffness value, the less the joint will bend.

You have to set the Stiffness value for all the joints in a joint chain in order for Stiffness to work properly because Stiffness calculations are relative between joints.

#### Related topics

❖ "Joint Rotation Limit Damping" on page 126
❖ "Stiffness" on page 123

### Local axes orientation

You can set the orientation of your joint's local axes to maintain symmetry in your character's skeleton. For example, if you create a joint chain for the right arm of a character, you can then mirror the joint chain to copy it for the left arm. See "Mirroring joint chains" on page 38. You should then align one of the joint's local axes with the bone so that it is pointing down the joint chain. This ensures that the joint chain will behave properly when you apply IK.

#### Related topics

❖ "To manually orient a joint's local axes" on page 63
❖ "To automatically orient a joint's local axes" on page 64
❖ "Skeleton > Orient Joint" on page 112

### Preferred angles

In a skeleton, each joint's preferred angles indicate the preferred rotations of the joint during inverse kinematic (IK) posing. When you build a skeleton, you should draw the joints so that they are oriented at the angles you would want them to move in during IK.

For example, if you are drawing a joint chain for the leg of a human character, you should draw the joints so that the bends in the joint chain properly represent the angle's of the limb in its rest position. The pose at which you set a skeleton's preferred angles is called the *rest pose.* Then, when you set the preferred angles for the leg's joint chain, the bend at the knee joint will be set as its preferred angle for IK posing.

### Related topics


- ❖ "To set a joint's preferred angles" on page 66
- ❖ "To make a joint assume its preferred angles" on page 66
- ❖ "Skeleton > Set Preferred Angle" on page 113
- ❖ "Skeleton > Assume Preferred Angle" on page 114


## Skeletons and skinning

Skinning is the process of binding a deformable object to a skeleton. When building a skeleton, you should take into consideration which skinning method you eventually want to use. Building a skeleton for an object that is to be smooth skinned requires a different construction strategy than building a skeleton for rigid skinning or an indirect skinning method such as lattice or wrap skinning.

For example, since smooth skinning enables gradual deformations that can be influenced by several joints, you can use joints to create deformation effects that resemble breathing or muscle action. Using joints for this would not work very well with rigid skinning. Instead, with rigid skinning you would use flexors to create similar deformation effects.

For information on how to skin your objects, see "Understanding skinning" in Chapter 3.

### **Posing skeletons**

## Posing skeletons

In Maya, you move and pose your character during animation using *kinematics*. Kinematics is a system with which you can specify motion for your skeleton. There are two types of kinematics in Maya: forward kinematics (FK) and inverse kinematics (IK). Each type of kinematics is best suited for specific types of motion.

With forward kinematics, you rotate and key the individual joints in the skeleton. Forward kinematics is ideal for creating detailed arc motions such as the rotation of a biped's shoulder joint.

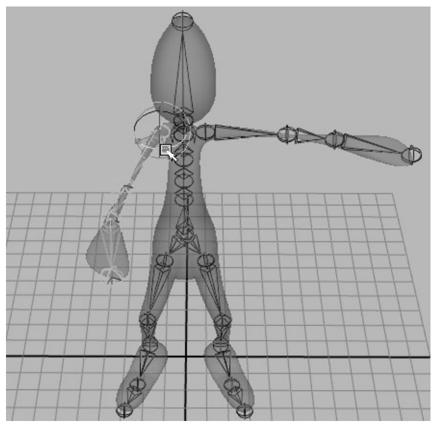
With inverse kinematics, you translate and key an IK handle to pose and animate the transformations and rotations of its IK chain's joints. All the joints between the start and end joints of the IK handle are driven by the IK handle. Inverse kinematics is ideal for creating goal-directed motions such as the motion of a biped's hand reaching for an object.

In addition to using solely FK or IK to pose and animate the joints of your skeleton, you can blend FK and IK animation on the same joints.

## Related topics


- ”Forward kinematics (FK)” on page 44
- ”Inverse kinematics (IK)” on page 45
- ”IK/FK blending” on page 55

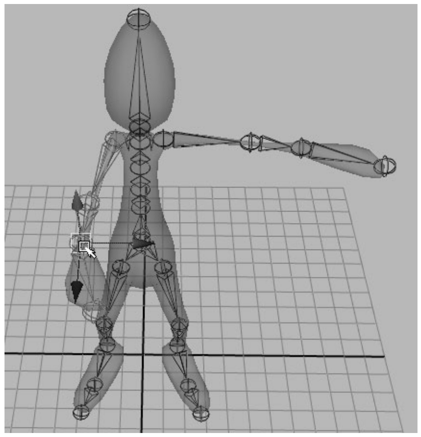

## Forward kinematics (FK)



With forward kinematics (FK), you rotate or move individual joints to pose and animate your joint chains. See ”Pose joints with FK” on page 68. Moving a joint affect’s that joint and any joints below it in the hierarchy. For example, if you want a joint chain to reach for a particular location in space, you have to rotate each joint individually so that the joint chain can reach the location. To do this, you rotate and translate the joint chain’s parent joint, then the next joint, and so on down the joint chain. When you animate a skeleton using forward kinematics, Maya interpolates the joint rotations starting with the root joint, then the root’s child joints, and so on down through the skeleton’s hierarchy.

Forward kinematics is ideal for creating non-directed motion (such as the rotations of a shoulder joint), and should not be used to animate a large, complicated skeleton. Also, forward kinematics is difficult to use for specifying goal-directed motion. For example, if you use FK to animate a skeleton, it is difficult to keep a foot still when the body moves forward. Also, when you think about moving your hand to some location in space, you don’t normally think about how you are going to rotate all the joints in your arm.

## Inverse kinematics (IK)



With inverse kinematics (IK), you move an *IK handle* to pose an entire joint chain. An IK handle is an object you can select and move that affects the joints it is assigned to. See "IK handles" on page 46. A joint chain that has an IK handle is called an *IK chain*. When you pose and animate a joint chain with an IK handle, the *IK solver* automatically rotates all the joints in the IK chain. The IK solver is what Maya uses to calculate the rotations of all the joints in the IK chain when you position the IK handle. See "IK solvers" on page 47.

Inverse kinematics is more intuitive for goal-directed motion than forward kinematics because you can focus on the goal you want a joint chain to reach without worrying about how each joint in the chain should rotate.

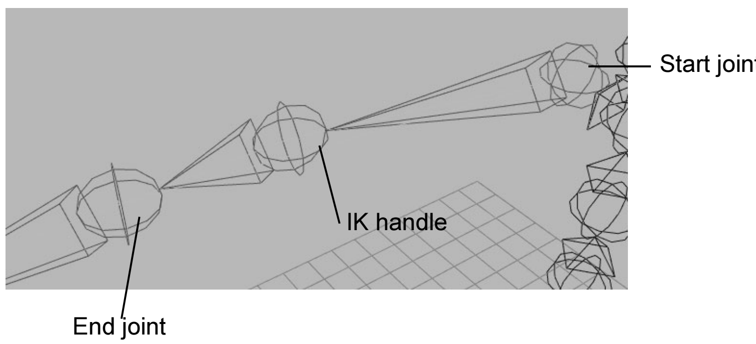
**Warning**

- You can not add to or remove joints from a skeleton if it is bound (skinned) to a model. If you do so, you will have to redo the IK handles or the skinning.
- If you want to pose your joint chains using IK handles, you should avoid creating joint chains that are drawn in straight lines. Rotating some of the joints in your joint chains slightly at various appropriate angles before inserting an IK handle makes it easier for you to pose the chains with IK.

# IK handles

In an IK chain, the joint where the IK handle begins is called the *start joint* and the joint where the IK handle ends is called the *end joint*. All the joints between the start and end joints are driven by the IK handle and its solver. In the scene view, the IK handle is drawn as a line from the start and end joints of the IK chain.

The end of the IK handle, which is located at the last joint of the IK chain by default, is called the *end effector*. When you move the IK handle, the IK solver uses the end effector's position and orientation in its calculations to rotate the joints in the IK chain accordingly. The end effector tries to follow the IK handle's position at all times. However, depending on the rotational limits and fully extended length of the IK chain, the end effector might not be able to reach IK handle. To move the IK handle to the end effector's position, see "Animate > IK/FK Keys > Move IK to FK" on page 102.




> **Note** The IK handle's end effector can only be viewed from the Hypergraph.

To create an IK handle, use the IK Handle Tool or the IK Spline Handle Tool. See "Skeleton > IK Handle Tool" on page 105 and "Skeleton > IK Spline Handle Tool" on page 107.

## General IK handle components

To show all the current IK handle's components, see "View all IK handle components" on page 70.

### Handle Wire

The handle wire is the line that runs through all the joints and bones in a joint chain controlled by an IK handle. The handle wire begins at the start joint's local axis and ends at the end joint's local axis.

To view an image of the handle wire, see ”Rotate Plane solver” on page 49.

### Handle vector

The handle vector is the line drawn from the start joint of the IK chain to the IK handle’s end joint (end effector). This is the axis used by the rotate plane.

To view an image of the handle vector, see ”Rotate Plane solver” on page 49.

## Related topics

- ”Specify the IK Handle Tool’s settings” on page 69
- ”Create IK handles” on page 69
- ”Pose joints with IK handles” on page 70
- ”Skeleton > IK Handle Tool” on page 105
- ”ikHandle” on page 127

# IK solvers

IK solvers are the mathematical algorithms behind the IK handles. IK solvers calculate the rotations of all the joints in a joint chain controlled by an IK handle. The effect an IK handle has on a joint chain depends on the type of IK solver used by the IK handle. By default, Maya loads the following IK solvers on start-up:

- Single Chain IK solver (ikSCsolver)

  See ”Single Chain solver” on page 48 and ”Use single chain IK” on page 72.
- Rotate Plane IK solver (ikRPsolver)

  See ”Rotate Plane solver” on page 49 and ”Use rotate plane IK” on page 73.
- Spline IK solver (ikSplineSolver)

  You can use the spline IK solver only in conjunction with a spline IK handle. See ”Spline IK solver” on page 52, ”Use spline IK” on page 75, and ”Skeleton > IK Spline Handle Tool” on page 107.

In addition to the default solvers, you can create the following solvers:

- 2 Bone IK solver (ik2Bsolver)

  Before you can use the 2 bone IK solver, you first need to manually load it in Maya. See ”2 Bone solver” on page 54 and ”Use 2 bone IK” on page 83.
- Multi-Chain IK solver (ikMCsolver)

To create the multi-chain IK solver, use the `createNode` MEL command. See ”Multi-chain solver” on page 55 and ”Use multi-chain IK” on page 74.

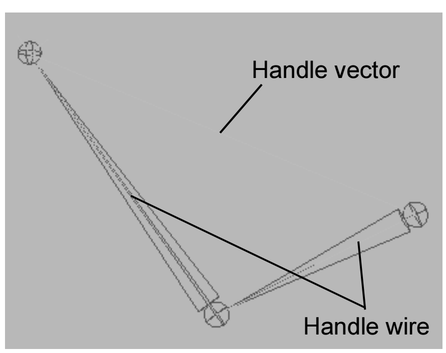
If you want to pose and animate joint chains that have between two and four joints, use single chain or rotate plane IK. If you want to pose and animate longer joint chains, use spline IK. If you want to pose and animate simple three joint IK chains for use in games, use 2 bone IK.

By default, each IK handle you create that uses the same type of IK solver, also shares the same IK solver node. For example, all IK handles that use single chain IK also connect to the same ikSCsolver node. Consequently, if you edit the attributes of the shared IK solver node, all the IK handles that connect to the node are affected. If you want to fine-tune the IK solvers for certain IK handles only, while not affecting other IK handles, you can create additional IK solvers for your IK handles using the `createNode` MEL command.

## IK solver calculations

When you move an IK handle, the solver performs the appropriate calculations to move and rotate all the joints in its IK chain accordingly. First, the solver looks at the position (Translate X, Y, and Z attributes) and orientation (Rotate X, Y, and Z attributes) of the IK handle. Next, the solver calculates how to move the position and orientation of the end effector as close to the IK handle’s position and orientation as possible. To do that, the solver calculates how to best rotate the joints in the IK handle’s joint chain. Finally, the solver then rotates all the joints in the joint chain so that the end effector reaches the IK handle’s position and orientation.

## Single Chain solver




A single chain IK handle uses the single chain solver to calculate the rotations of all the joints in the IK chain. Also, the overall orientation of the joint chain is calculated directly by the single chain solver.

### Difference between single chain and rotate plane IK handles

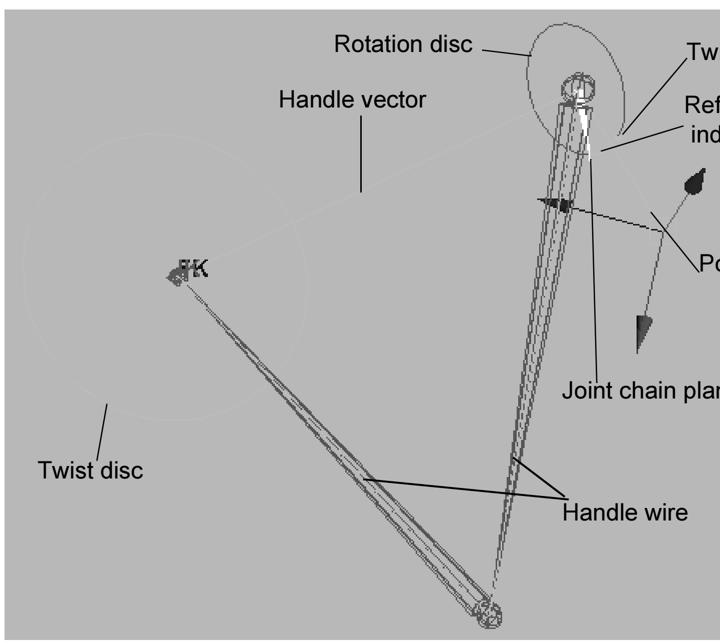
The difference between a single chain IK handle and a rotate plane IK handle is that the single chain IK handle's end effector tries to reach the position and the orientation of its IK handle, whereas the rotate plane IK handle's end effector only tries to reach the position of its IK handle. Since the rotate plane IK handle's end effector only tries to reach the position of its handle, the resulting joint rotations are more predictable. For the rotate plane IK handle, the orientation of its entire joint chain is controlled by the *twist disc* manipulator. For more information, see "Twist disc" on page 50 and "Rotate Plane solver" on page 49.

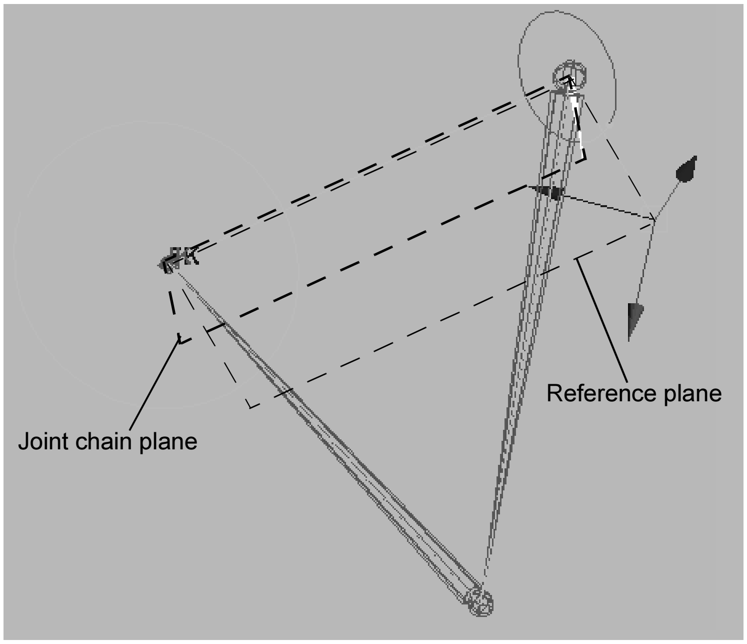
**Note** If your joint chain is suffering from flipping, use the rotate plane solver instead of the single chain solver. The rotate plane solver was introduced with the pole vector to control the flipping of IK chains that you sometimes get with the single chain solver.

### Single Chain IK handle components

See "General IK handle components" on page 46.

## Rotate Plane solver






A rotate plane IK handle uses the rotate plane solver to calculate the rotations of all the joints in its IK chain, but not the joint chain's overall orientation. Instead, the IK rotate plane handle gives you direct control over the joint chain's orientation via the pole vector and twist disc, rather than having the orientation calculated by the IK solver. The single chain solver and rotate plane solver differ in this respect. See ”Difference between single chain and rotate plane IK handles” on page 49.

The rotate plane solver is ideal for posing joint chains (such as arms and legs) that you want to stay in the same plane. For example, the shoulder, elbow, and wrist joints of an arm driven by a rotate plane IK handle all stay within the same plane as the elbow rotates. The plane itself can be rotated from the shoulder joint by the pole vector. See ”Pole vector” on page 51 and ”Twist disc” on page 50.

## Rotate Plane IK handle components

### Twist disc

The twist disc is a manipulator that you can use to twist or rotate the joint chain. The twist disc is located at the *end* joint of the IK chain.

Translating the pole vector often leaves the IK chain pointing in the wrong direction. You can use the twist disc to re-orient the plane after you move the pole vector to prevent flipping. To view an image of the twist disc, see ”Rotate Plane solver” on page 49.

### Joint chain plane

The joint chain plane is the plane that contains all the joints in the joint chain and poses through the axis. The joint chain plane rotates about the handle vector. When you manipulate the pole vector, you are rotating the joint chain plane about the handle vector. To view an image of the joint chain plane, see ”Rotate Plane solver” on page 49.

### Reference plane

For the joint chain plane to rotate and twist the joint chain, the plane must rotate relative to some other plane so that the degree of twist can be measured. The plane that the joint chain plane rotates relative to is the *reference plane*. To view an image of the reference plane, see ”Rotate Plane solver” on page 49.

### Pole vector

The pole vector is a manipulator that lets you change the orientation of the IK chain. The pole vector also lets you control *flipping*.

Since moving the pole vector changes the orientation of the reference plane, moving the pole vector can also change the orientation of the joint chain directly; just as manipulating the twist disc can change the orientation of the joint chain. This is because the joint chain’s degree of orientation—or *twist*—is defined as the difference in orientation between the reference plane and the joint chain plane. To view an image of the pole vector, see ”Rotate Plane solver” on page 49.

**Warning** When positioning your IK handle, if the handle vector and the pole vector cross each other or point in exact opposite directions, the joint chain can suddenly flip. You can prevent this flipping by moving the pole vector so that the handle vector does not cross it or point in the opposite direction from it.

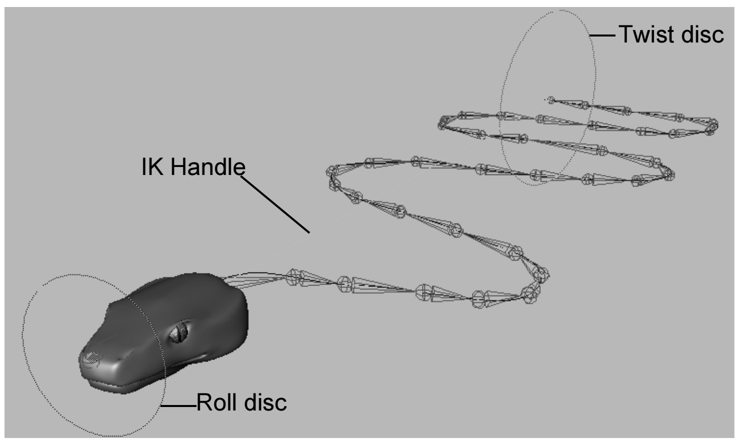
### Rotation disc

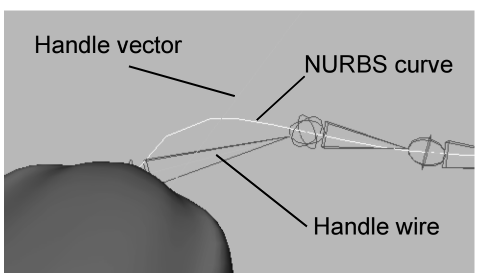
The rotation disc is an indicator that displays how much the IK chain has been rotated by the twist disc. The rotation disc is located at the start joint of the IK chain. To view an image of the rotation disc, see ”Rotate Plane solver” on page 49.

**Reference plane indicator** The reference plane indicator is the green dot on the rotation disc that moves to reflect the movements of the pole vector.

| | |
|---|---|
| Twist indicator | The twist indicator is the green arc between the reference plane indicator and the joint chain plane indicator on the rotation disc. The twist indicator displays the orientation of the joint chain relative to the reference plane. |
| Joint chain plane indicator | The joint chain plane indicator displays the orientation of the joint chain plane relative to the reference plane. The joint chain plane indicator appears in the rotation disc. |

## Spline IK solver




Spline IK handles let you pose a joint chain using a NURBS curve. When you manipulate the curve, the handle's spline IK solver rotates the joints in the chain accordingly. You can use spline IK to pose and animate long, sinuous joint chains such as those for a tail, a tentacle, a snake and so on.

### Spline IK handle components

#### Twist disc

See "Twist disc" on page 50.

#### Roll disc

The roll disc is a manipulator that lets you roll or rotate the entire spline Ik joint chain. The roll disc is always at the spline IK handle's *start* joint. The roll disc is similar to the rotation disc and pole vector. See "General IK handle components" on page 46.

### Advanced spline IK controls

With the spline IK twist control attributes, you can constrain the local rotation of the joints in a chain to a fixed worldspace *vector*. This vector is the orientation of the spline IK NURBS curve. For example, you can use the advanced spline IK twist controls to stabilize a snake character, the spine of a biped character, or the movements of a coil spring. Also, to fine-tune the twist along the chain, you can add additional twist to the joints with the Twist Value attributes.

By aligning the joints Up axes with a fixed worldspace vector before applying additional twist, you can achieve a more predictable and stable result than was previously possible.

See "Use the advanced Spline IK twist controls" on page 80 and "Add Advanced Twist Control Attributes" on page 132.

**Note** The Advanced Twist Control attributes are relevant only if your IK handle uses the ikSplineSolver.

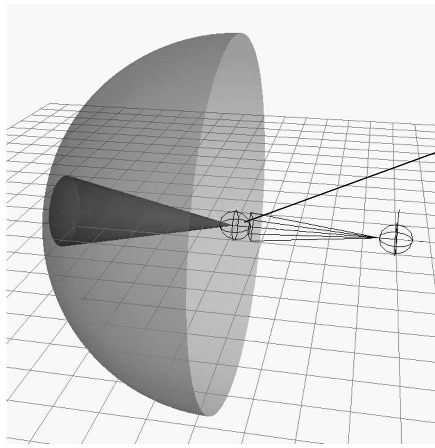
IK handles with spline IK let you pose your joint chains using NURBS curves. When you manipulate the spline IK curve, you are moving and rotating all the joints in the target IK chain. An easy way to manipulate the spline NURBS curve is to create a cluster deformer for each of the curves CVs. See Edit Curves > Selection > Cluster Curve in the *Modeling NURBS* book.

Spline IK handles are ideal for posing and animating long joint chains like those for tails, tentacles, necks, spines, and similar objects.

### Start joint flipping

The start joint of your spline IK joint chain can sometimes flip when you move or rotate the spline IK curve or its CVs, or when you slide the joint chain along its curve. The flipping is a normal result of the spline IK solver's calculations.

Joint flipping occurs when the orientation of a joint is more than 90 degrees from its rotation value of 0. A joint's rotation value is 0 when its Rotate X, Y, and Z values are 0, relative to its parent joint's rotation values. Flipping is most pronounced when a joint nears 180 degrees rotation.




You can prevent start joint flipping by moving each of the joints after the start joint to roughly their rest positions. See ”Prevent flipping in the spline IK chain” on page 82.

Unwanted start joint flipping can also occur when you animate a joint chain along its curve, for instance, when you slide a snake along a motion path.

If you have positioned joints appropriately and joint flipping still occurs, parent the start joint to another joint or to a transform node. See ”Auto Create Root Axis” on page 107 and ”Auto Parent Curve” on page 107.

## 2 Bone solver

The 2 bone IK solver is a subset of the *rotate plane* IK solver. Therefore, IK handles with the 2 bone IK solver solve the rotations of their joint chains in the same manner as a rotate plane IK handle. See ”Rotate Plane solver” on page 49 and ”Rotate Plane IK handle components” on page 50.

The two bone IK handle is meant for posing and animating short joint chains that consist of three joints (two bones). If you try to pose and animate a longer joint chain with the two bone IK handle, then the 2 bone solver will solve for the rotations of only the start and second to last joints and will ignore all other joints in the joint chain.

The 2 bone solver is ideal for posing joint chains (such as arms and legs) that you want to stay in the same plane. For example, the shoulder, elbow, and wrist joints of an arm driven by a rotate plane IK handle all stay within the same plane as the elbow rotates. The plane itself can be rotated from the shoulder joint by the pole vector. See ”Pole vector” on page 51.

The 2 bone solver is the fastest IK solver in Maya. This makes two bone IK handles ideal for setting up characters in a games development environment. Maya includes the source code for the two bone IK solver plug-in so that game developers can replicate the exact behavior of this feature in a games engine or modify the code to create their own custom IK solvers.

#### IK two bone solver plug-in source code

The source code for the two bone IK solver is available in the devkit's ik2Bsolver directory. The source code provides an example of how you can create your own IK solver plug-in. Also, by extracting the core algorithm, you can replicate the exact behavior of the 2 bone IK solver in a games engine. For more information, please read the README file in the ik2Bsolver directory.

### Multi-chain solver

The multi-chain solver can solve for multiple IK handles simultaneously. You can use the multi-chain IK handles to animate complex motions such as those of the tentacles of an octopus character.

An IK handle with the multi-chain solver manages the joints in its joint chain in the same manner as a *single chain* IK handle. See "Single Chain solver" on page 48.

**Note** The IK handle's Priority and Weight attributes apply only to the multi-chain IK solver. See *ikMCsolver* in Help > Node and Attribute Reference.

### Related topics

- "Use single chain IK" on page 72
- "Use rotate plane IK" on page 73
- "Use multi-chain IK" on page 74
- "Use spline IK" on page 75
- "Use 2 bone IK" on page 83

## IK/FK blending

You can pose and animate the joints of a joint chain using both Fk and IK. This is called animation *blending*. See "Blend IK and FK animation" on page 85. This blending between FK and IK animation is possible because of the IK handle's *Ik Blend* attribute. Ik Blend lets you switch between posing and animating with pure FK or pure IK, as well as control the blend between the two kinds of animation. See "Ik Blend" on page 130.

Posing and animating a joint chain with both FK and IK changes the way the joints and bones are displayed in your scene view. A joint chain with FK and IK is drawn using three default or user defined colors. The blue joint chain represents the skeleton with pure FK animation, the brown joint chain represents the skeleton with pure IK animation, and the magenta joint chain represents the resulting animation blend. Also, you can customize the colors of the IK/FK joint chains and the size of the IK/FK joints. See ”Set the display of IK/FK animated joint chains” on page 91.

In addition to blending IK and FK animation over multiple frames, a blend can occur over a single frame. If you want to animate an instantaneous switch between IK and FK, such a planted, stationary hand quickly rising into the air, you should blend IK and FK over a single frame. Whereas if you want to animate a motion that eases in or out of IK or FK, such as a character jumping up to grab a horizontal bar, you should blend IK and FK over several frames. Blending IK and FK over several frames is also useful when you want the prevent jerking in your animation.

## Related topics

❖ ”Blend IK and FK animation” on page 85
❖ ”IK Solver Attributes” on page 130

### Retargeting animation

## Animation retargeting

You can retarget animation data from the joints of one skeleton (source) to the joints of another skeleton (target). The *source* skeleton is the skeleton *from* which you want to retarget its animation data, and the *target* skeleton is the skeleton *to* which you want to retarget the source’s animation.

Retargeting is useful when you want to transfer animation between skeletons that have the same or different proportions.

**Note** When your skeletons have an IK, FK, and an IK/FK skeleton, then choose just one of the types of skeletons on which to perform the retarget. For example, *source* FK skeleton > *target* FK skeleton.

## Retargeting workflow

- Set a neutral pose for your source and target skeletons.

  See ”Set a skeleton’s neutral pose” on page 93.

- Label the joints of your source and target skeletons.
  See "Label my skeleton's joints" on page 94.
- Set the retargeting solver options.
  See "Skeleton > Retargeting > Toggle Selected Labels" on page 116.
- Perform the retarget.
  See "Retarget animation from one skeleton to another" on page 97.

# How do I? Set-up joint chains

## Create a joint or joint chain

Before drawing joints or a joint chain, you first need a geometry in which to place them. For instructions on creating models, see the Modeling Polygons, Modeling NURBS, or Modeling Subdivision Surfaces guides.

#### To set the Joint Tool settings

1. In the Animation menu set (press F2), select Skeleton > Joint Tool > ❐.
   The Joint Tool settings panel appears.
2. Adjust the Joint Tool settings as desired.
3. Do one of the following:
   - Click Reset Tool to reset to the default tool settings.
   - Click Close to close the tool settings panel.

#### To create a joint or joint chain

1. In the Animation menu set (press F2), select Skeleton > Joint Tool.
2. In a view, click on the location in your model where you want to create a joint.

| Tip | You can use the left mouse button to create joints and the middle mouse button to move the last placed joint. |
|---|---|

3. Click again in the model where you want to create the next joint in your joint chain.
   A bone appears between the first and second joints.
4. Continue clicking in the view until you create all the joints for your joint chain.

5 Press the Enter (IRIX, Linux, and Windows) or Return (Mac OS X) key to complete your joint chain.

## Related topics

- "Creating joint chains" on page 38
- "Skeleton > Joint Tool" on page 103

# Create a joint chain with an IK handle

**To create a joint chain with an IK handle**

1 In the Animation menu set (press F2), select Skeleton > Joint Tool > ❐.

The Joint Tool settings panel appears.

2 In the Joint Tool settings, turn on Create IK Handle.

3 Set the IK Handle Settings as desired.

4 Close the Joint Tool settings panel.

Now when you create a joint chain with the Joint Tool, an IK handle is automatically created with the joint chain.

## Related topics

- "Skeleton > Joint Tool" on page 103

# Insert joints

**To place a joint in an existing joint chain**

1 In the Animation menu set (press F2), select Skeleton > Insert Joint Tool.

**Note** The Insert Joint Tool does not have tool settings.

2 Press the left mouse button on the joint you want to be the parent of the new joint, and then drag it to where you want to place the new joint.

3 Repeat step 2 until you finish inserting all the joints you want.

4 Press Enter (IRIX, Linux, and Windows) or Return (Mac OS X).

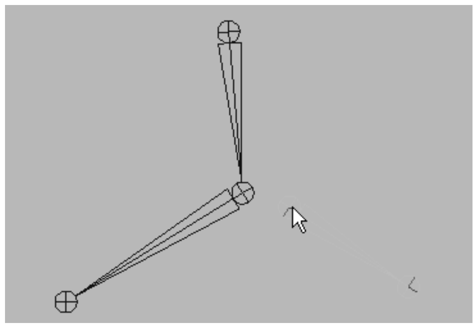
## Related topics

- See "Skeleton > Insert Joint Tool" on page 109

# Connect joints

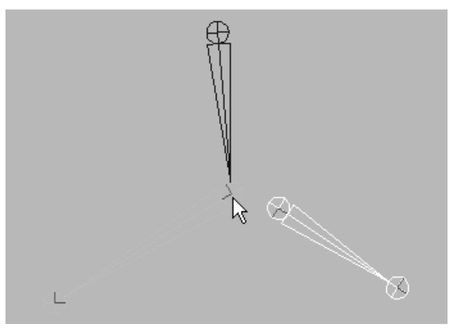
### To connect joint chains by combining joints

**1** Click the parent of the joint chain you want to connect to another joint chain.



**2** In another joint chain or skeleton, Shift + click on the joint that will replace the previously selected joint and become its joint chain's parent.

| | |
|---|---|
| **Note** | The joint that you select can not be a root joint. |



**3** In the Animation menu set (press F2), select Skeleton > Connect Joint > ❐.

The Connect Joint Options window appears.

**4** Turn on Connect Joint.

Connect Joint connects joint chains by combining the two selected joints; where the first joint selected is combined with and replaced by the second joint.

**5** Do one of the following:

- Click Connect to merge the selected joints.

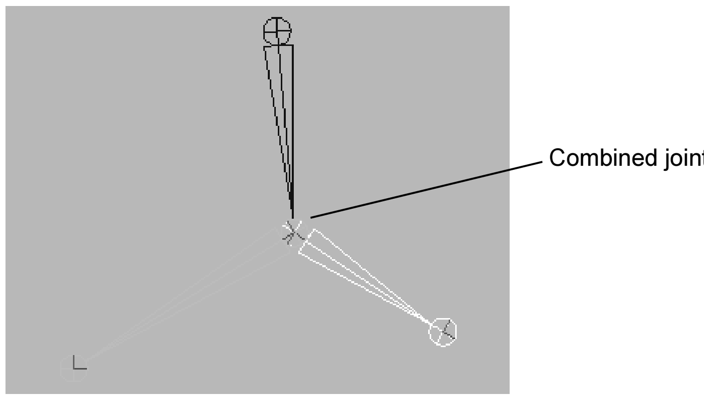
The Connect Joint Options window closes.

- Click Apply to merge the selected joints.

  The Connect Joint Options window remains open.

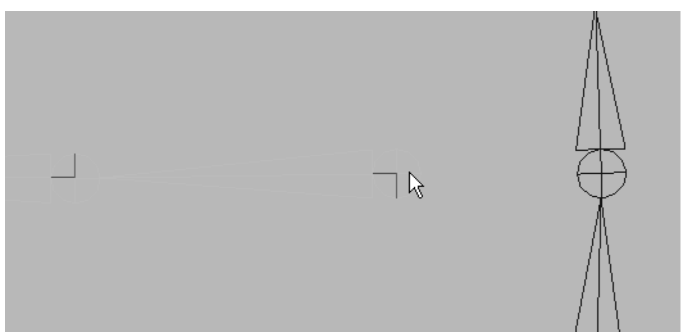
- Click Close to disregard the options window's settings.

  The Connect Joint Options window closes.



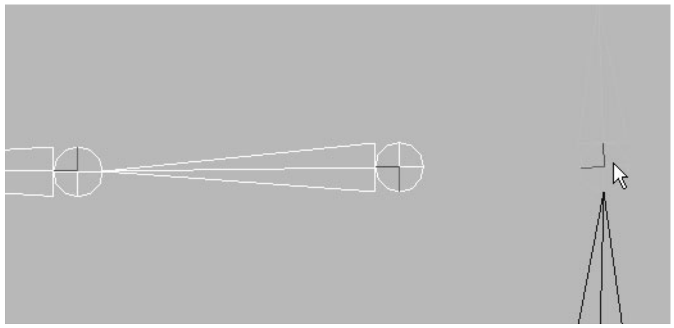

**To connect joint chains by connecting selected joints with a bone**

**1** Click the parent of the joint chain you want to connect to another joint chain.



**2** In another joint chain, Shift + click on the joint that will be the previously selected joint chain's parent.

| | |
|---|---|
| **Note** | The joint that you select can not be a root joint. |



**3** In the Animation menu set (press F2), select Skeleton > Connect Joint > ❒.

The Connect Joint Options window appears.

**4** Turn on Parent Joint.

Parent Joint connects joint chains by creating a new bone between two selected joints; where the first selected joint is the child joint and the second joint is its parent.

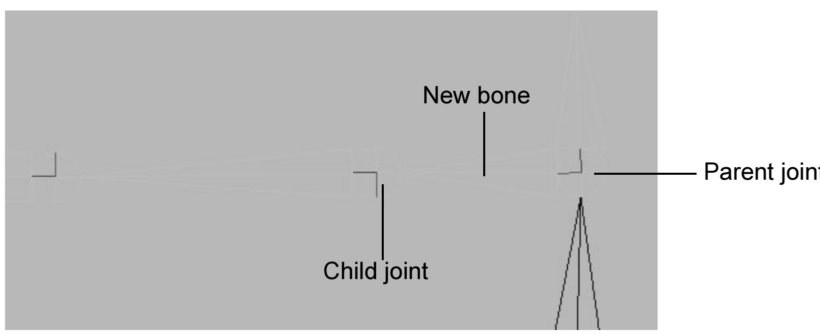
**5** Do one of the following:

- Click Connect to connect the selected joints with a bone.

  The Connect Joint Options window closes.
- Click Apply to connect the selected joints with a bone.

  The Connect Joint Options window remains open.
- Click Close to disregard the options window's settings.

  The Connect Joint Options window closes.




## Related topics

❖ "Connecting joint chains and skeletons" on page 39
❖ "Skeleton > Connect Joint" on page 110

## Move joints

### To move a joint

1 Select the Move Tool (hotkey: W) from the Tool Box.
2 Click on the joint you want to move.
   The current joint's transform manipulators appear.
3 You can now press drag the transform manipulators to translate the current joint along its X, Y, and Z axes.

## Move joints while using the Joint Tool

### To move the most recently created joint in the joint chain you are drawing

With the Joint Tool and most recently created joint active, do one of the following:

- Click the middle mouse button on the joint.
  The joint's transform manipulators appear.
  - Press the middle mouse button on the joint and drag it to its new position.
  - After you move the current joint, click in your scene view to continue creating joints for your joint chain.
- Press the Insert (IRIX, Linux, and Windows) or Home (Mac OS X) key.
  The joint's transform manipulators appear.
  - Drag the joint to its new position.
  - After you move the current joint, press the Insert (IRIX, Linux, and Windows) or Home (Mac OS X) key again and click in the scene view to continue creating joints for your joint chain.

## Mirror joint chains

### To mirror a joint chain

1 Select the parent joint of the joint chain you want to duplicate.
2 In the Animation menu set (press F2), select Skeleton > Mirror Joint > ❐.
   The Mirror Joint Options window appears.
3 In the Mirror Across options, select the plane across which you want the joint chain mirrored.

These options represent the plane whose origin is at the joint chain's parent joint.

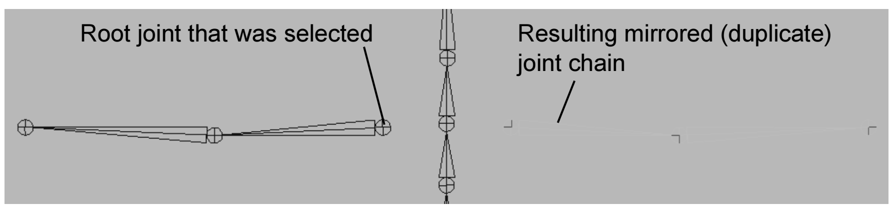
4. Set the Mirror Function.
5. If you want to rename the mirrored (duplicate) joints, enter their old names in the *Search For* box and their new names in the *Replace With* box.
6. Do one of the following:
   - Click Mirror to duplicate and flip the current joint chain in the specified plane.

     The Mirror Joint Options window closes.
   - Click Apply to duplicate and flip the current joint chain in the specified plane.

     The Mirror Joint Options window remains open.
   - Click Close to disregard the options window's settings.

     The Mirror Joint Options window closes.

The joint chain is mirrored across the selected plane whose origin is at the joint chain's parent joint.




Mirrored across YZ

## Related topics

- ❖ "Mirroring joint chains" on page 38
- ❖ "Skeleton > Mirror Joint" on page 111

# Orient a joint's local axes manually

### To manually orient a joint's local axes

1. Select the joint whose local rotation axes orientation you want to set.
2. Select Display > Component Display > Local Rotation Axes.

   The current joint's local rotation axes appear.
3. Click the Select by component type icon in the Status Line to turn on the component selection mode.
4. Click the Miscellaneous mask icon in the Status Line.

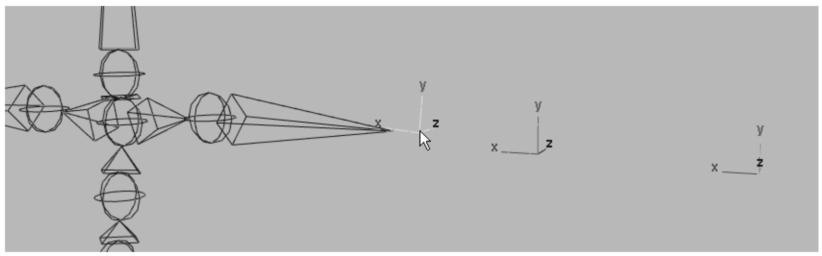
5 Press the right mouse button on the Miscellaneous mask icon.

The following menu appears:


6 Turn on Local Rotation Axes.

You can now select the local rotation axis of your joint.

7 Select the local axis.



8 Do one of the following:

- Select the Rotate Tool (hotkey: E) from the Tool Box and use the Rotate Tool to orient the local axis by hand.

  For information on using the Rotate Tool, see the Basic guide.
- If you want to enter precise orientation values for the local axis, you can use a MEL command like the following:

```
rotate -r -os 180 0 0;
```

  This command rotates the current joint's local axes 180 degrees about its X-axis.

## Related topics

❖ "Local axes orientation" on page 42

# Orient a joint's local axes automatically

### To automatically orient a joint's local axes

1 Select the joint whose local rotation axes orientation you want to set.

2 Select Display > Component Display > Local Rotation Axes.

The current joint's local rotation axes appear.

3 In the Animation menu set (press F2), select Skeleton > Orient Joint > ❐.

The Orient Joint Options window appears.

**4** Set the Orient Joint Options.

If you turn on Orient Child Joints in the options, you can view the local rotation axes of all the joints in the current joint's joint chain by performing steps 4, 5, and 6 from ”Orient a joint's local axes manually” on page 63.

**5** Do one of the following:

- Click Orient to reorient the current joint's local rotation axes using the current Orient Joint Options settings.

  The Orient Joint Options window closes.

- Click Apply to reorient the current joint's local rotation axes using the current Orient Joint Options settings.

  The Orient Joint Options window remains open.

- Click Close to disregard the options window's settings.

  The Orient Joint Options window closes.

## Related topics

❖ ”Local axes orientation” on page 42

❖ ”Skeleton > Orient Joint” on page 112

# Reroot skeletons

You can change a skeleton's hierarchy by changing which joint is its root joint. This process is called *rerooting*. You can reroot a skeleton early in its creation to test the skeleton's behavior with different roots.

**Note**

- Any IK handles that pass through the joint selected to be the new root joint will be deleted.
- Any animation on the skeleton's root joint will be affected by rerooting your skeleton

### To reroot a skeleton

**1** Select the joint in your skeleton you want as the new root.

If you select the child of the entire joint chain, the skeleton's hierarchy will reverse.

If you select a joint in the middle of the skeleton, two separate hierarchies below the root joint will be created.

**2** In the Animation menu set (press F2), select Skeleton > Reroot Skeleton.

## Related topics

- ”Skeleton > Reroot Skeleton” on page 110

# Set preferred angles

| | |
|---|---|
| **Note** | If you want an entire joint chain or skeleton to assume its preferred angles, select the parent joint of your joint chain or the root joint of your skeleton. |

**To set a joint’s preferred angles**

1 Select a joint.

2 In the Animation menu set (press F2), select Skeleton > Set Preferred Angle > ❐.

The Set Preferred Angle Options windows opens.

3 Set the options.

4 Do one of the following:

- Click Set to set the joint’s current local orientation as its preferred angles.

  The Set Preferred Angle Options window closes.

- Click Apply to set the joint’s current local orientation as its preferred angles.

  The Set Preferred Angle Options window remains open.

- Click Close to disregard the options window’s settings.

  The Set Preferred Angle Options window closes.

**To make a joint assume its preferred angles**

1 Select a joint.

2 In the Animation menu set (press F2), select Skeleton > Assume Preferred Angle > ❐.

The Assume Preferred Angle Options window appears.

3 Set the options.

4 Do one of the following:

- Click Assume to make the current joint assume its preferred angles.

  The Assume Preferred Angle Options window closes.

- Click Apply to make the current joint assume its preferred angles.

The Assume Preferred Angle Options window remains open.

- Click Close to disregard the options window's settings.

  The Assume Preferred Angle Options window closes.

| | |
|---|---|
| **Note** | You can also set a joint chain or skeleton's Preferred Angles by pressing the right mouse button on the chain or skeleton, and then selecting Assume Preferred Angle from the marking menu that appears. |

## Related topics

❖ "Preferred angles" on page 42
❖ "Skeleton > Set Preferred Angle" on page 113
❖ "Skeleton > Assume Preferred Angle" on page 114

# Disconnect joints

You can disconnect joints in a skeleton to separate joint chains. The joint that is disconnected becomes the root joint of a new skeleton.

| | |
|---|---|
| **Note** | • If you disconnect a joint in a joint chain that has an IK handle, the IK handle will be deleted.<br>• You can not disconnect a root joint. |

**To disconnect joints**

1. Select the joint you want to disconnect.
2. Open the Animation menu set (press F2).
3. Do one of the following:
   - Select Skeleton > Disconnect Joint.
   - Press the right mouse button on the joint.

     A marking menu appears.

     - Select Actions > Unparent.

     The bone between the current joint and its parent joint is removed.

The joint disconnects from its skeleton and becomes the root of a new skeleton.

### Related topics

❖ "Skeleton > Disconnect Joint" on page 110

## Remove joints

| Note | |
|---|---|
| | • You can only remove one joint at a time. |
| | • You can not remove a root joint. |
| | • Do not remove joints that are bound to a surface (skinned). |

**To remove a joint**

1 Select the joint that you want to remove.
2 In the Animation menu set (press F2), select Skeleton > Remove Joint.

The joint is deleted from your scene and the bone of the joint above the removed joint now connects to the joint below the removed joint.

**To remove a parent joint and all its children**

1 Select the joint that you want to remove.
2 Press the right mouse button on the joint.
  A marking menu appears.
3 Select Actions > Unparent.
  The bone between the current joint and its parent joint is removed.
4 Press Delete.

The current joint and all its child joints are deleted from your scene.

### Related topics

❖ "Skeleton > Remove Joint" on page 110

# Pose with Forward or Inverse Kinematics

## Pose and animate with FK

## Pose joints with FK

**To pose and animate a joint with forward kinematics**

1 Select the joint you want to animate.
2 Select the Rotate Tool (hotkey: E) from the Tool Box.

The rotation transform manipulators appear.

**3** Advance the current time indicator along the Time Slider to the point in time where you would like to set a *key*.

**4** Rotate the joint to pose the current joint and all its child joints and then set a key (hotkey: S).

For more information on animating with FK, see the Keyframe animation chapter in the Animation guide.

### Set-up an IK handle

## Specify the IK Handle Tool's settings

#### To specify the IK Handle tool settings

**1** In the Animation menu set (press F2), select Skeleton > IK Handle Tool > ❐.

The IK Handle Tool settings panel appears.

**2** Do one of the following:

- Adjust the IK Handle Tool settings as desired.
- Click Reset Tool to reset to the default tool settings.

### Related topics

❖ "Skeleton > IK Handle Tool" on page 105

## Create IK handles

#### To create an IK handle

**1** In the Animation menu set (press F2), select Skeleton > IK Handle Tool.

Your mouse pointer changes to the tool's crosshair cursor.

If you want to set the IK Handle Tool's options, open the Tool Settings panel.

**2** Click the joint in your joint chain that you want as your IK handle's *start* joint.

**3** Click the joint in your joint chain that you want as your IK handle's *end* joint.

An IK handle that starts and ends at the joints you selected appears on the joint chain.

**4** For the new IK handle, select the IK solver to calculate the orientation of the joints in the IK handle's joint chain.

## Related topics


- ”IK handles” on page 46
- ”Create spline IK handles” on page 76
- ”Skeleton > IK Handle Tool” on page 105
- ”Create a single chain IK handle” on page 72
- ”Create a rotate plane IK handle” on page 73
- ”Create a multi-chain IK handle” on page 75
- ”Create a 2 bone IK handle” on page 83


# View all IK handle components

### To view all the components of the current IK handle

1. Select the IK handle.
2. Select the Show Manipulator Tool from the Tool Box (hotkey: T).

   All the current handle’s components appear in your scene view.

| Note | Only rotate plane and 2 bone IK handles have extra components. See ”Rotate Plane solver” on page 49 and ”2 Bone solver” on page 54. |
|---|---|

# Pose joints with IK handles

### To move the IK handle

1. Select the IK handle.
2. Select the Move Tool (hotkey: W) from the Tool Box.

   The translate manipulators appear.
3. In the current view, drag the IK handle to move it as desired.

   This moves the joint chain controlled by the current IK handle.

### To rotate the IK handle

1. Select the IK handle.
2. Select the Rotate Tool (hotkey: E) from the Tool Box.

   The rotation manipulators appear.
3. In the current view, press drag the IK handle to rotate it as desired.

   This rotates the joint chain controlled by the current IK handle.

## Related topics

- "Posing skeletons" on page 43
- "IK handles" on page 46

# Delete IK handles

### To delete an IK handle

1. Select the IK handle.
2. Do one of the following:
   - Select Edit > Delete.
   - Press Delete.

   The IK handle is deleted from your scene. However, the IK solver node remains in the scene so that it can be used by other IK handles in the scene, and by future IK handles.

# Set IK solver attributes

### To set IK solver attributes

1. Select the IK handle that has the solver you want to edit.
2. In the Attribute Editor, select the IK solver node's tab (for example, ikSCsolver*n*).
3. Set its attributes as desired.

   See "IK solver nodes" on page 136.

# Turn all IK solver nodes on or off

### To disable all IK solvers

Turn off Modify > Evaluate Nodes > IK Solvers. When off, IK Solvers does not have a checkmark.

### To enable all IK solvers

Turn on Modify > Evaluate Nodes > IK Solvers. When on, IK Solvers has a checkmark.

## Use single chain IK

# Create a single chain IK handle

### To create a single chain IK handle

1 Select the IK handle.
2 In the Attribute Editor, select the ikHandle node tab.
3 In the IK Solver Attributes section, select *ikSCsolver* from the IK Solver drop-down menu.

   The current IK handle is now driven by the single chain IK solver.

## Related topics

❖ "IK handles" on page 46
❖ "Create a joint chain with an IK handle" on page 58
❖ "Create IK handles" on page 69
❖ "IK solvers" on page 47
❖ "Single Chain solver" on page 48

# Pose joints with single chain IK handles

### To move a single chain IK handle

"To move the IK handle" on page 70.

### To rotate a single chain IK handle

"To rotate the IK handle" on page 70.

## Related topics

❖ "Posing skeletons" on page 43

# Create an ikSCsolver node in addition to the default

### To create additional single chain IK solvers

1 In the Script Editor or Command Line, type the following MEL command:

```
createNode ikSCsolver;
```

2 Press Enter in the number pad.

A new ikSCsolver*n* node appears in the dependency graph.

This solver can by used by any IK handle in your scene. See ”Create IK handles” on page 69.

### Use rotate plane IK

## Create a rotate plane IK handle

#### To create a rotate plane IK handle

1. Select the IK handle.
2. In the Attribute Editor, select the ikHandle node tab.
3. In the IK Solver Attributes section, select the *ikRPsolver* from the IK Solver drop-down menu.

   The current IK handle is now driven by the rotate plane IK solver.

### Related topics

❖ ”IK handles” on page 46
❖ ”Create a joint chain with an IK handle” on page 58
❖ ”Create IK handles” on page 69
❖ ”IK solvers” on page 47
❖ ”Rotate Plane solver” on page 49

## Pose joints with rotate plane IK handles

#### To move the rotate plane IK handle

See ”To move the IK handle” on page 70.

#### To manipulate the pole vector

1. Select the rotate plane IK handle.
2. Select the Show Manipulator Tool (hotkey: T) from the Tool Box.

   The IK handle’s pole vector, twist disc, and rotation disc appear.
3. In the current view, drag the pole vector to move it as desired.

   This rotates the joint chain controlled by the current IK handle.

#### To manipulate the twist disc

1. Select the rotate plane IK handle.
2. Select the Show Manipulator Tool (hotkey: T) from the Tool Box.

   The IK handle’s pole vector, twist disc, and rotation disc appear.

3 Select the twist disc.

   The twist disc turns yellow.

4 Drag the twist disc.

   This twists the joint chain controlled by the current IK handle.

## Related topics

- ”Posing skeletons” on page 43
- ”Pole vector” on page 51
- ”Rotate Plane solver” on page 49
- ”Twist disc” on page 50

# Create an ikRPsolver node in addition to the default

#### To create additional rotate plane IK solvers

1 In the Script Editor or Command Line, type the following MEL command:

```
createNode ikRPsolver;
```

2 Press Enter in the number pad.

   A new ikRPsolver*n* node appears in the dependency graph.

### Use multi-chain IK

# Create the multi-chain solver

#### To create the ikMCsolver node for your IK handle

By default, the multi-chain IK solver is not available.

1 In the Script Editor or Command Line, type the following MEL command:

```
createNode ikMCsolver;
```

2 Press Enter on the number pad.

   A new ikMCsolver*n* node appears in the dependency graph.

This solver is now available from the Joint Tool, the IK Handle Tool, and the Attribute Editor for any IK handle.

## Related topics

- ”IK solvers” on page 47
- ”Multi-chain solver” on page 55

## Create a multi-chain IK handle

**1** Create the multi-chain solver.

See "Create the multi-chain solver" on page 74.

**2** Select the IK handle.

**3** In the Attribute Editor, select the ikHandle node tab.

**4** In the IK Solver Attributes section, select the *ikMCsolver* from the IK Solver drop-down list.

The current IK handle is now driven by the multi-chain IK solver.

### Related topics

❖ "IK handles" on page 46
❖ "Create a joint chain with an IK handle" on page 58
❖ "Create IK handles" on page 69
❖ "IK solvers" on page 47
❖ "Multi-chain solver" on page 55

## Pose joints with multi-chain IK handles

#### To move a multi- chain IK handle

"To move the IK handle" on page 70.

#### To rotate a multi-chain IK handle

"To rotate the IK handle" on page 70.

### Related topics

❖ "Posing skeletons" on page 43

### Use spline IK

## Specify the IK Spline Handle Tool's settings

#### To specify the IK Spline Handle tool settings

**1** In the Animation menu set (press F2), select Skeleton > IK Spline Handle Tool > ❐.

The IK Spline Handle Tool settings panel appears.

**2** Do one of the following:

- Adjust the IK Spline Handle Tool settings as desired.

- Click Reset Tool to reset to the default tool settings.

## Related topics

❖ ”Skeleton > IK Spline Handle Tool” on page 107

# Create spline IK handles

A spline IK handle lets you control a joint chain with a NURBS curve. All joints driven by the spline IK handle stay on and follow the curve. The spline IK handle gives you additional attributes, such as *twist*, for additional control over the orientation of the joints in your joint chain. See ”ikHandle” on page 127.

### To create a spline IK handle and automatically create the spline IK NURBS curves

1. Select your joint chain.
2. In the Animation menu set (press F2), select Skeleton > IK Spline Handle Tool > ❐.

   The IK Spline Handle Tool settings panel appears.
3. Turn on Auto Create Curve.

   See ”Auto Create Curve” on page 108.
4. Select a Number of Spans for the spline IK curve.

   See ”Number of Spans” on page 109.
5. Click the *start* joint for the spline IK handle.
6. Click the *end* joint for the spline IK handle.

   The spline IK handle and a NURBS curve appears on the joint chain.

### To create a spline IK handle using a pre-existing NURBS curve

1. Draw a NURBS curve along the path of the joint chain.

   Create a simple curve with no sharp bends to ensure the joint chain moves smoothly when you animate the curve.

   If you create a curve with few CVs, your control of the curve’s shape and skeleton’s movement will be less precise, but you’ll be able to manipulate the curve and its joint chain more easily.

   For more information, search the online help for NURBS curves.
2. Select your joint chain.
3. In the Animation menu set (press F2), select Skeleton > IK Spline Handle Tool > ❐.

The IK Spline Handle Tool settings panel appears.

4 Turn off Auto Create Curve.

See "Auto Create Curve" on page 108.

5 Click the *start* joint for the spline IK handle.

6 Click the *end* joint for the spline IK handle.

7 Select the NURBS curve.

The spline IK handle appears on the joint chain. The joints in the chain rotate to follow the shape of the curve.

## Related topics

❖ "IK handles" on page 46
❖ "Skeleton > IK Handle Tool" on page 105
❖ "IK solvers" on page 47
❖ "Spline IK solver" on page 52

# Pose joints with spline IK curves

### To manipulate the CVs of the spline IK NURBS curve

1 Select the NURBS curve.

2 Turn on Display > NURBS Components > CVs.

3 Turn on Display > NURBS Components > Hulls.

4 Select a CV on the NURBS curve.

**Tip** For easier manipulation of the curve, cluster groups of the curve's CVs. For more information on how to create a cluster, see the Deformers chapter of the Character Setup guide.

5 Do one of the following:

- Select the Move Tool (hotkey: M) from the Tool Box.
- From the Modeling menu set, select Edit Curves > Curve Editing Tool.

6 Drag a CV.

This moves the spline IK joint chain.

### To create a cluster deformer for each CV on the spline IK NURBS curve

1 Select the spline IK curve.

2 In the Modelling menu set, select Edit Curves > Selection > Cluster Curve.

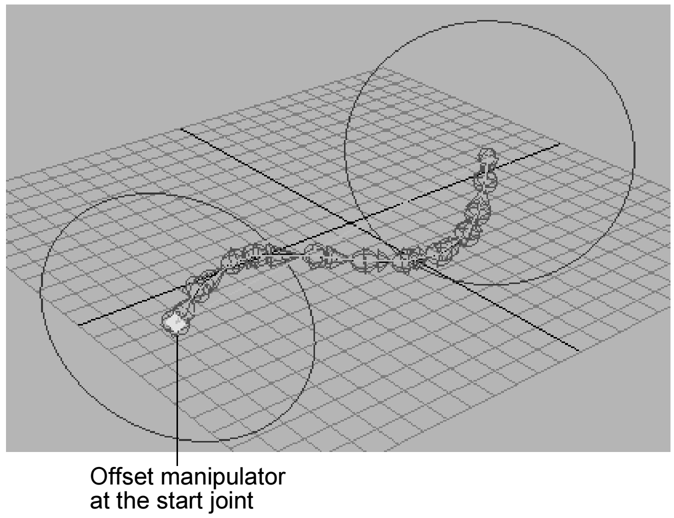
A cluster deformer is created for each of the spline IK curve’s CVs.

#### To twist and roll the spline IK joint chain

1 Select the spline IK handle.
2 Select Modify > Transformation Tools > Show Manipulator Tool.
 The twist and roll circular manipulators appear at the start joint and end joints of the spline IK chain.
3 Do the following:
 - To roll the entire joint chain, drag the circular manipulator at the start joint.
 - To twist the joint chain, drag the circular manipulator at the end joint.
 - You can also adjust twist and roll by selecting the spline IK handle and entering values for Roll and Twist in the Channel Box or Attribute Editor.

#### To slide the joint chain along the curve

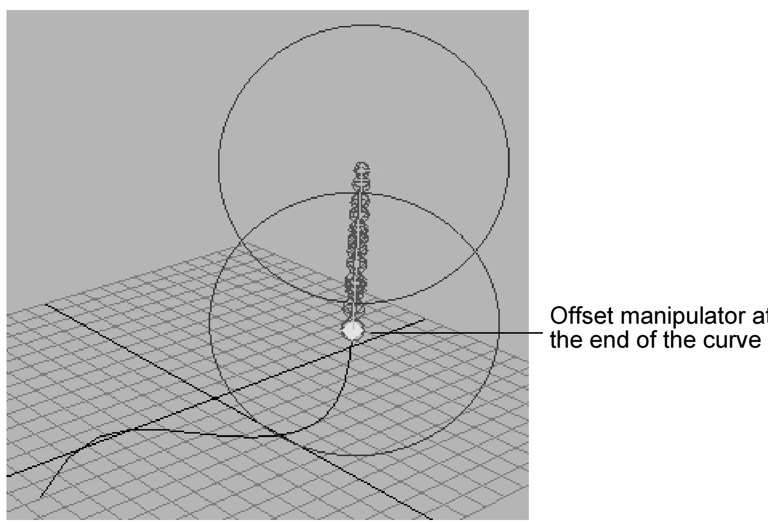
1 Select the spline IK handle.
2 In the Attribute Editor, select the ikHandle node’s tab.
3 In the IK Solver Attributes section, turn on Root On Curve.
 This constrains the start joint of the spline IK handle to a position on the curve. It also provides an offset manipulator to slide the start joint along the curve.
4 Select Modify > Transformation Tools > Show Manipulator Tool.
 The offset manipulator appears at the start joint.




**5** Do one of the following:

- Drag the manipulator to slide the joint chain along the curve.

  If you drag the start joint to the end of the curve, the child joints move off the end of the curve in a straight line.




- Enter values for Offset in the Attribute Editor to move the root of the spline IK chain along the curve.

| **Note** | |
|---|---|
| | • You cannot drag the manipulator past the ends of the curve.<br>• If you turn Root On Curve off, the Offset attribute is ignored. |

**To translate, rotate, and scale the spline IK curve**

1 Select the spline IK curve.
2 Do one of the following:
   - Select the Move Tool (hotkey: W) from the Tool Box
     The translation manipulators appear.
   - Select the Rotate Tool (hotkey: E) from the Tool Box
     The rotation manipulators appear.
   - Select the Scale Tool (hotkey: R) from the Tool Box
     The scaling manipulators appear.
3 Use the Move, Rotate, and Scale tools to translate, rotate, or scale the curve.

| Note | If you created the spline IK handle with Root on Curve off in the tool options, translating, rotating, and scaling the curve does not translate the start joint. |
|---|---|

## Related topics

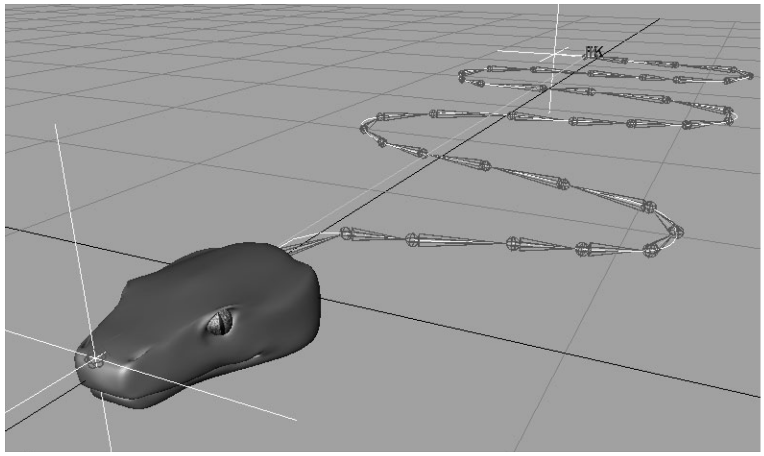
- ”IK handles” on page 46
- ”IK solvers” on page 47
- ”Spline IK solver” on page 52
- ”Twist disc” on page 50
- ”Posing skeletons” on page 43
- For information on animating with spline IK, see the Keyframe animation chapter in the Animation guide.

# Use the advanced Spline IK twist controls

## Example scenario

A snake character with Spline IK slithers across the forest floor. The snake is manipulated by translating and rotating the NURBS curve. The NURBS curve is driven by control objects (in this example, they are locators) at the beginning (head) and end (tail) of the curve.

1 Select the snake character’s Spline IK handle.



The Spline IK handle starts at the first joint and ends at the last joint of the snake’s skeleton, and its NURBS curve runs through all the joints and bones.

2. In the Attribute Editor, select the ikHandle node.
3. In the IK Solver Attributes, click the Add Advanced Twist Control button.

   The advanced Spline IK twist control attributes appear.
4. Set the World Up Type to Object Rotation Up (Start/End) and the Up Axis to Positive Y.

   This constrains the start and end joint Up vectors of the snakes joints to the Y axes of the locators used to drive the NURBS curve.

| | |
|---|---|
| **Note** | The Up Axis attribute specifies what local joint axis is aligned with the Up Object vectors specified in the Up Vector fields. |

5. Type the name of the head locator in the World Up Object field and the tail locator in the World Up Object 2 field.

   The positive Y orientation of the locators is used to calculate the Up vectors of the joints in the Snake’s skeleton.

| | |
|---|---|
| **Note** | You can specify a specific vector offset for the joints with the Up Vector and Up Vector 2 fields. |

6. Select and move the head locator.

This moves all the joints and bones in the snake's skeleton. Notice that the joints and bones also now respect the orientation of the manipulation objects.

## Prevent flipping in the spline IK chain

**To prevent flipping when a joint chain slides down its curve**

1. In the Animation menu set (press F2), select Skeleton > IK Spline Handle Tool > ❐.

   The IK Spline Handle Tool settings appears.
2. Turn *off* Root on Curve, Auto Parent Curve, Auto Create Curve, and Snap Curve to Root.
3. Turn *on* Auto Create Root Axis.
4. Click the start joint, the end joint, and then the NURBS curve.

   This creates the spline IK handle with a parent transform node above the start joint. In one of the following steps, you'll put the node on a motion path that prevents the start joint from flipping.
5. Select the parent transform node, then Shift-click the curve.

   You can select the parent transform node by dragging a selection box around the start joint.
6. Select Animate > Motion Paths > Attach to Motion Path > ❐.

   The Attach to Motion Path Options window appears.
7. Turn on Start/End.
8. For the Start Time and End Time, enter the frame range for the joint chain's motion.

   The parent transform node and its child joint chain move from the start of the curve to the end of the curve in the specified frame range.
9. Turn on Follow.

   If the curve has a 3D looping shape, you may also need to turn on Normal for the Up Direction to avoid unwanted flipping.
10. Leave all the other options at their default settings.
11. Click the Attach button.
12. Now when you play the animation, the parent transform node and joint chain move along the curve path. The movement should be free of unexpected flipping. However, flipping is unavoidable if the path is very complex.

**Note** You can roll and twist the joint chain with the IK handle's roll and twist manipulators for additional control. See "To twist and roll the spline IK joint chain" on page 78.

## Use 2 bone IK

### Create a 2 bone solver

**To load the 2 bone IK solver**

1. Select Window > Settings/Preferences > Plug-in Manager.

   The Plug-in Manager window appears.
2. Scroll through the list of plug-ins and turn on *loaded* for ik2Bsolver.mll.
3. Click Close.

   The Plug-in Manager window closes.
4. In the Script Editor or Command Line, type:

   ```
   ik2Bsolver;
   ```

   and then press Enter in the number pad.

The 2 bone IK solver is now available for all IK handles.

#### Related topics

❖ "IK solvers" on page 47
❖ "2 Bone solver" on page 54

### Create a 2 bone IK handle

**To assign a 2 bone solver to an IK handle**

1. Select your IK Handle.
2. In the Attribute Editor, select the ikHandle*n* tab.
3. In the IK Solver Attributes section, select *ik2Bsolver* from the IK Solver drop-down menu.

   The current IK handle will now use the 2 bone IK solver.

#### Related topics

❖ "IK handles" on page 46

# Pose joints with 2 bone IK handles

### To move the 2 bone IK handle

See ”To move the IK handle” on page 70.

### To manipulate the pole vector

1. Select the 2 bone IK handle.
2. Select the Show Manipulator Tool (hotkey: T) from the Tool Box.
   The IK handle’s pole vector, twist disc, and rotation disc appear.
3. In the current view, drag the pole vector to move it as desired.
   This rotates the joint chain controlled by the current IK handle.

### To manipulate the twist disc

1. Select the 2 bone IK handle.
2. Select the Show Manipulator Tool (hotkey: T) from the Tool Box.
   The IK handle’s pole vector, twist disc, and rotation disc appear.
3. Select the twist disc.
   The twist disc turns yellow.
4. Press drag the twist disc.
   This twists the joint chain controlled by the current IK handle.

## Related topics

- ”IK handles” on page 46
- ”IK solvers” on page 47
- ”2 Bone solver” on page 54
- ”Rotate Plane solver” on page 49
- ”Pole vector” on page 51
- ”Twist disc” on page 50
- ”Posing skeletons” on page 43

# Blend FK and IK

## Blend IK and FK animation

> Note If you want to rotate or key individual joints in a joint chain that already have IK animation, turn on Ik Fk Control. See "ikHandle" on page 127.

**To create an animation blend from FK to IK**

1. Select the IK handle of the Ik chain that you want to animate with FK.
2. In the IK Solver Attributes, drag the *Ik Blend* slider to 0.000.

   The animation mode is now set to pure FK.
3. Select Animate > IK/FK Keys > Set IK/FK Key.

   All the joints in the IK chain and the IK handle are keyed.

   You will now be keying the joints (FK)—not the handle (IK).
4. Deselect the IK handle.
5. Select a joint in the IK chain, drag the current time indicator along the Time Slider and rotate the joint.
6. Set a key.
7. Repeat steps 5 and 6 until you complete the FK portion of your animation.
8. Once you set the last FK key, deselect the current joint and select the IK handle of your joint chain.
9. Make sure that the Ik Blend slider is still at 0.000 and select Animate > IK/FK Keys > Set IK/FK Key.
10. Drag the current time indicator forward in time 1 frame.
11. Drag the Ik Blend slider to 1.000 and set a key.

    Since there is no period of animation between the last pure FK key and the first pure IK key, the FK animation switches to IK instantly (without a blend).

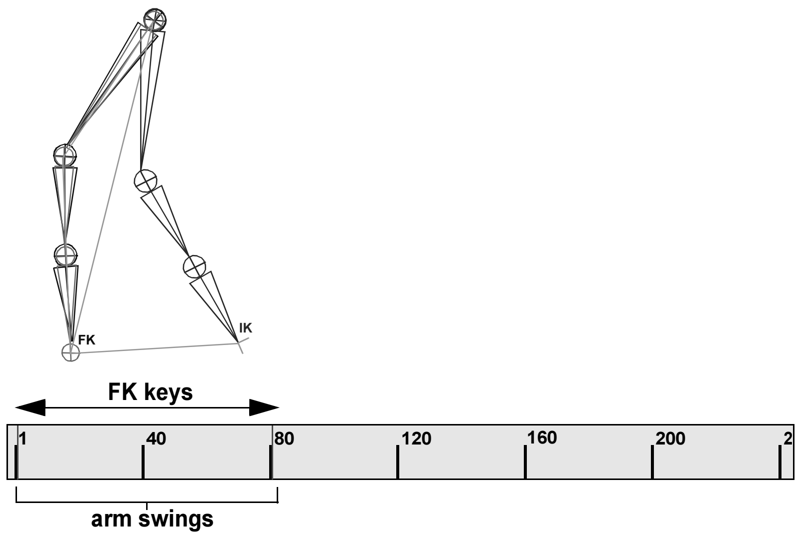
    The animation mode is now set to pure IK.

> Note When there is a period of animation between a pure FK key and a pure IK key, it is interpolated by the IK Solver. The interpolated animation then appears as Ik Blend values between 0.000 and 1.000.

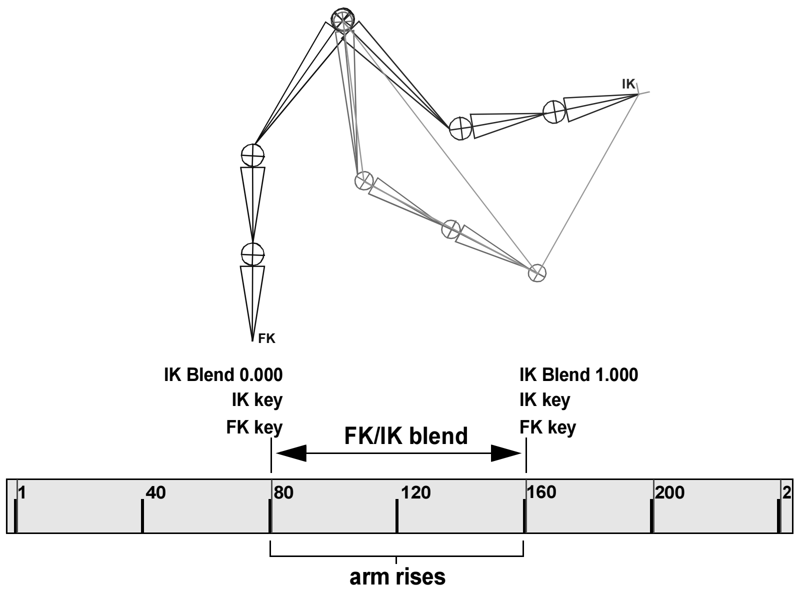
**12** Drag the current time indicator along the Time Slider, translate the IK handle and set keys as desired.

## Example of IK/FK blending

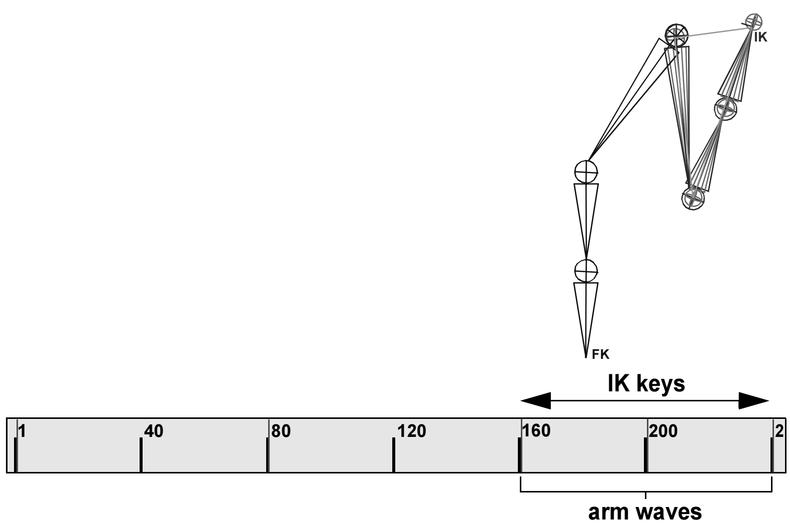
Blending the FK animation of a swinging arm with the IK animation of a waving arm.




A simple animation of a swinging arm during a walk cycle is created by selecting the root of an arm's joint chain (the shoulder joint), rotating it along the Z axis, and then setting keys for the joint. This produces a rudimentary FK animation.




This FK animation sequence is blended with the directed motion of the arm waving by setting keys on the IK handle and the shoulder joint at the beginning and end of the FK animation, moving the current time indicator to create a *blend region*, changing the Ik Blend value to 1.000 (pure IK), and once again setting keys on the shoulder joint and IK handle. In the area between the pure FK and IK animation—the blend region, the IK Solver interpolates the animation from 0.000 to 1.000.



Pure IK animation is reached once the Ik Blend value is 1.000. The IK handle is then translated and rotated while setting keys, to produce the animation of the arm waving.

When the animation of the waving arm is complete, the animation is set back (using another blend region and changing the Ik Blend value to 0.000) to pure FK. The resulting animation resembles a swinging arm (FK) that rises into a wave (IK) and then descends back into a swinging motion (return to FK).

#### To insert IK within FK animation (whether or not the FK is controlled by Set IK/FK Key)

1. Open the Animation menu set (press F2).
2. At the first frame of the time range where you want to insert IK, select the IK handle and set a key by selecting Animate > IK/FK Keys > Set IK/FK Key.
3. At the end frame of the time range where you want to insert IK, select the IK handle and set a key by selecting Animate > IK/FK Keys > Set IK/FK Key.

   We refer to the keys you set in the prior two steps as *bounding keys*, because they ensure that any keys you set between them will not change the animation outside their range.
4. Turn *on* Animate > IK/FK Keys > Enable IK Solver.
5. Manipulate the IK handle and use Set IK/FK Key as desired for the frames between the bounding keys.

#### To insert FK within IK animation controlled by Set IK/FK Key

1. Open the Animation menu set (press F2).
2. At the first frame of the time range where you want to insert FK, select the IK handle and set a key by selecting Animate > IK/FK Keys > Set IK/FK Key.
3. At the last frame of the time range where you want to insert FK, select the IK handle and set a key by selecting Animate > IK/FK Keys > Set IK/FK Key.

   We refer to the keys you set in the prior two steps as *bounding keys*. They ensure that any keys you set between them will not inadvertently change the animation outside of their range.
4. Turn *off* Animate > IK/FK Keys > Enable IK Solver.
5. Rotate the desired joints and use Set IK/FK Key as necessary for the frames between the bounding keys.

#### To insert FK within IK animation *not* controlled by Set IK/FK Key

1. Open the Animation menu set (press F2).
2. At the first frame of the animation, select the IK handle, turn *on* Solver Enable, and set a key for Solver Enable.
3. At the end frame of the time range where you want to insert FK, select the IK handle and set a key by selecting Animate > IK/FK Keys > Set IK/FK Key.
4. At the first frame of the time range where you want to insert FK, select the IK handle and set a key by selecting Animate > IK/FK Keys > Set IK/FK Key.

   We refer to the keys you set in the prior two steps as *bounding keys*. They ensure that any keys you set between them will not inadvertently change the animation outside of their range.
5. Turn *off* Animate > IK/FK Keys > Enable IK Solver.
6. Rotate the desired joints and use Set IK/FK Key as necessary for the frames between the bounding keys.

#### To eliminate unexpected joint flipping after turning on the IK Solver

After FK animation, a joint chain may flip to an undesired position when you turn on Animate > IK/FK Keys > Enable IK Solver. To undo the joint flipping and prevent it from occurring when you turn on Enable IK Solver, do the following:

1. Open the Animation menu set (press F2).
2. Undo the Enable IK Solver menu item to return the joint chain to the position it had before flipping.
3. Select the IK handle.
4. Select Skeleton > Set Preferred Angle.
5. Select Animate > IK/FK Keys > Enable IK Solver.
6. Select Animate > IK/FK Keys > Set IK/FK Key.

   Continue using IK as desired.

### Related topics

❖ "Forward kinematics (FK)" on page 44
❖ "Inverse kinematics (IK)" on page 45
❖ "IK/FK blending" on page 55
❖ "Animate > IK/FK Keys > Set IK/FK Key" on page 101
❖ "Ik Blend" on page 130

- ”Ik Fk Control” on page 130

# Adjust the skeleton display

## Display a joint’s local axis

### To display a joint’s local rotation axis

1. Select the joint.
2. Select Display > Component Display > Local Rotation Axes.

## Display all the local axes in a skeleton

### To display all local axes

1. Select a joint (a joint chain’s parent joint or a skeleton’s root joint).
2. Select Edit > Select Hierarchy.

   The entire skeleton or joint chain hierarchy is now selected.
3. Select Display > Component Display > Local Rotation Axes.

   All the local axes are displayed.

## Set display size of all joints

You can resize the display of a skeleton’s joints. Increasing the display size can make the joints and their bones easier to select. Decreasing the display size can make other objects such as flexors easier to select.

### To resize joints in the scene view

1. Select Display > Joint Size.

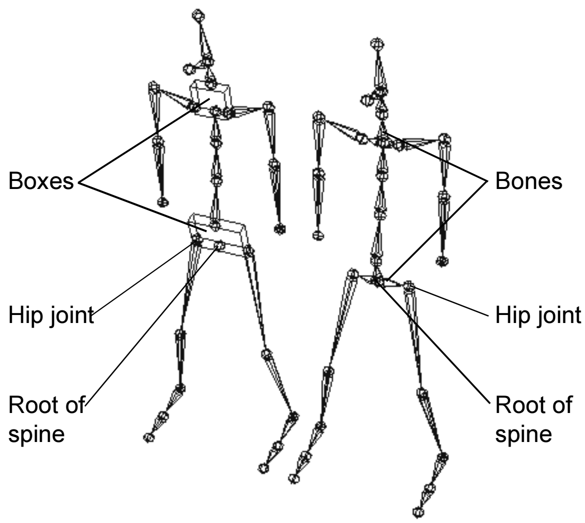
| | |
|---|---|
| **Note** | You can also set the size of joints from Windows > Settings/ Preferences > Preferences > Kinematics. |

2. Do one of the following:
   - Select from the percentages listed to resize the joints by 25%, 50%, 75%, or 100%.
   - Select Custom. Use the slider in the Joint Display Scale window that appears to select a resize percentage.

   The percentages are relative to the default setting, which is always 100% or 1.00.

## Display bones as boxes

Boxes are an alternate way of representing and visualizing joints. Bones are used to represent parent joints such as the pelvis or shoulders.




| | |
|---|---|
| **Note** | Only the bones of a joint with more than one child joint chain (see above figure) can display as a box rather than as bones. Multiple child-joint hierarchies typically occur at the upper back and root. |

#### To display bones as boxes

1. Select the joint.
2. In the Attribute Editor, select the jointn node tab.
3. In the Joint section, select Box from the Draw Style drop-down menu.

## Set the display of IK/FK animated joint chains

#### To set the display parameters for bones with FK or IK animation

1. Select Window > Settings/Preferences > Preferences from the Maya main menu bar.

   The Maya Preferences window appears.
2. Click Kinematics under Display to open the Kinematics Display Preferences.

   The Inverse Kinematics preferences appear.

3 Select a display option:
   - Select None if you do not want to display any of the IK/FK blend skeletons.
   - Select IK if you want to display only the IK and blend skeleton.
   - Select FK if you want to display only the FK and blend skeleton.
   - Select Both if you want to display all the IK/FK blend skeletons. Both is the default setting.

4 Do one of the following:
   - Click Save to save all changes to the IK/FK Blending Display. The Maya Preferences window remains open.
   - Click Cancel to disregard any changes to the IK/FK Blending Display. The Maya Preferences window closes.

#### To set the color of IK/FK joint chains

1 Select Window > Settings/Preferences > Colors from the Maya main menu bar.

   The Colors settings window appears.

2 Click the arrow beside IK/FK Blending to reveal the IK/FK joint color settings.

3 Click-drag the sliders beside FK Joints and IK/FK Blended Joints to set their display colors.

4 Do one of the following:
   - Click Save to save all changes to the Blend and FK Joint display colors. The Color settings window remains open.
   - Click Reset to Saved to cancel any changes made to the Blend and FK Joint display colors and revert to the color settings from your last file save. The Color settings window remains open.
   - Click Close to cancel any changes to the Blend and FK Joint display color. The Color settings window closes.
   - color. The Color settings window closes.

#### To set the joint size for bones with FK or IK animation

1 Select Window > Settings/Preferences > Preferences from the Maya main menu bar.

   The Maya Preferences window appears.

2 Click Kinematics under Display to open the Kinematics Display Preferences.

   The Inverse Kinematics preferences appear.

3 Set the joint size using the Ik/Fk Joint Size field or slider.

4 Do one of the following:
- Click Save to save all changes to the Ik/Fk Joint Size. The Maya Preferences window remains open.
- Click Cancel to disregard any changes to the Ik/Fk Joint Size. The Maya Preferences window closes.

# Retarget animation

## Retargeting workflow

See "Retargeting workflow" on page 56.

## Set a skeleton's neutral pose

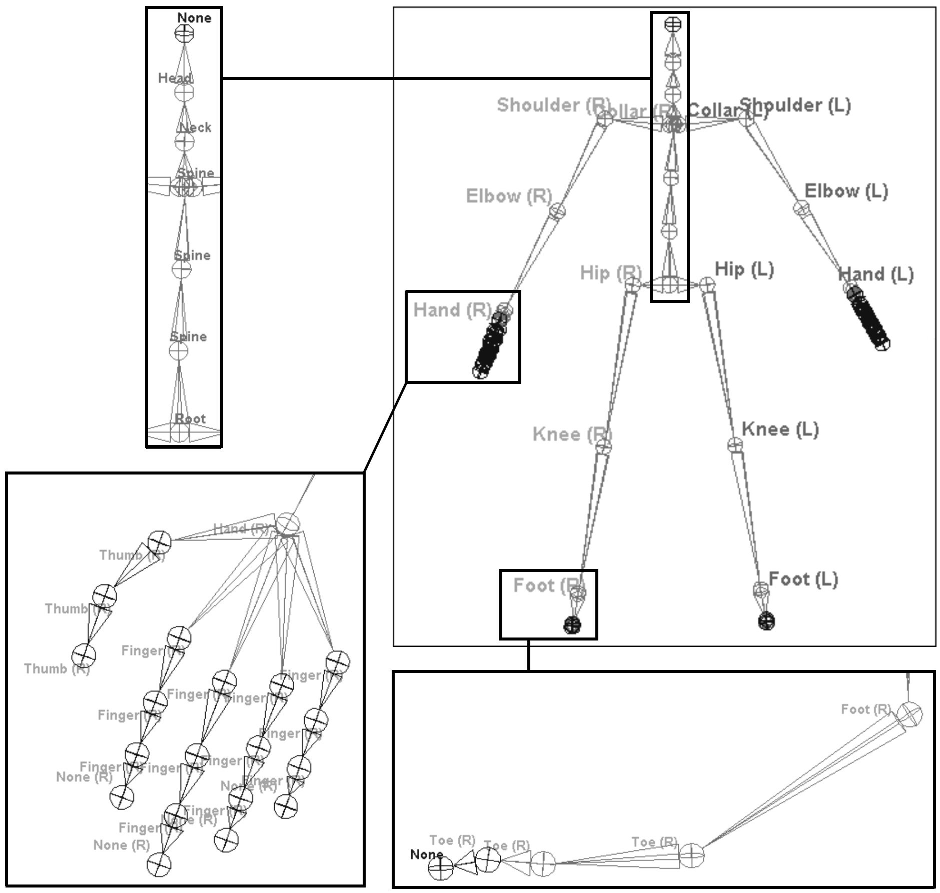
### To set a neutral pose

1 Position your character into the pose you want as its neutral pose.
2 Select the root of your skeleton.
3 Select Skeleton > Set Neutral Pose.

See "Skeleton > Retargeting > Set Neutral Pose" on page 115.

Your character's current pose is set as its neutral pose and the *retargetPose* tab appears on the root joint's node.

# Label my skeleton’s joints




There are three ways in which you can label the joints of your skeleton:

- Use the Skeleton > Retargeting > Joint Labelling menu to automatically label each joint in a selected joint chain. See ”To automatically label the joints of your skeleton’s limbs” on page 95.
- Use the Skeleton > Retargeting > Joint Labelling menu to label each individual joint in your skeleton. See ”To label your character’s joints using the Joint Labelling menu” on page 96.
- Select a joint and edit its label from the Attribute Editor. See ”To set or adjust individual joint labels from the Attribute Editor” on page 96.

**Note**
- If you want to retarget an arm or a leg, then you *must* label the three primary joints for those limbs on both the Source and the Target skeletons. For the arm, the three primary joints are: shoulder, elbow, and hand. For the leg, the three primary joints are: hip, knee, and foot.
- As you label your joints, each label appears beside their respective joints in the scene view. If you do not see the joint labels in the scene view, see ”Toggle the visibility of joint labels” on page 96.

### To automatically label the joints of your skeleton’s limbs

For the limbs of your source and target skeletons, do the following:

1. Select Skeleton > Retargeting > Show All Labels.

   Joints labels will now appear in the scene view. See ”Skeleton > Retargeting > Show All Labels” on page 116.
2. Select Skeleton > Retargeting > Joint Labelling.

   The Joint Labelling menu appears. See ”Skeleton > Retargeting > Joint Labelling” on page 116.
3. Tear-off the Joint Labelling menu.
4. In the scene view, select the parent or top joint of each limb (joint chain) in your skeleton and do the following:
   - Click a *side* label (Center, Left, Right) in the Joint Labelling menu.
   - Click a *type* label (Arm or Leg) in the Joint Labelling menu.

   Each joint, from the selected parent down its joint chain’s hierarchy, is automatically labeled according to their positions in their respective joint chain.

   For example, if you select the left shoulder joint of a skeleton and click *Left* and then *Arm* in the Joint Labelling menu, then that joint will be labelled as *Shoulder(L)*, the next joint in the chain will be automatically labeled as *Elbow(L)*, and the next joint will be labeled as *Hand(L)* and so on.

**Note** For arms, the forearm joint is labelled as an *elbow* joint. For legs, the ankle joint is labelled as a *foot* joint.

5 If you want to adjust or change the labels of your joints, see ”To label your character’s joints using the Joint Labelling menu” on page 96 or ”To set or adjust individual joint labels from the Attribute Editor” on page 96.

#### To label your character’s joints using the Joint Labelling menu

For your source and target skeletons, do the following:

1. Select Skeleton > Retargeting > Show All Labels.

   Joints labels will now appear in the scene view. See ”Skeleton > Retargeting > Show All Labels” on page 116.
2. Select Skeleton > Retargeting > Joint Labelling.

   The Joint Labelling menu appears. See ”Skeleton > Retargeting > Joint Labelling” on page 116.
3. Tear-off the Joint Labelling menu.
4. In the scene view, select a joint and do the following:
   - Click a *side* label (Center, Left, Right) in the Joint Labelling menu. Joint types such as Head or Spine do not require a type label.
   - Click a *type* label (Shoulder, Hip, Head, Spine and so on) in the Joint Labelling menu.

   The side and type labels appear beside the current joint. For example, if you selected *Right* as the side label and *Knee* as the type label, then the label *Knee(R)* will appear beside the joint in the scene view.
5. Repeat step 4 for the rest of the joints you want to label.

#### To set or adjust individual joint labels from the Attribute Editor

1. Select the joint in your source or target skeleton whose labelling you want to change and open the Attribute Editor.
2. In the Joint section of your *joint* node’s tab, expand Joint Mapping.

   The Side and Type drop-down menus appear.
3. Set or change the joint’s *Side* and *Type* labels as desired.

## Toggle the visibility of joint labels

#### To toggle the visibility of joint labels

There are three ways in which you can turn the visibility of joint labels in the scene view on and off:

- Use the Display > Component Display > Joint Labels or Skeleton > Retargeting > Toggle Selected Labels to toggle on and off the visibility of joint labels in the scene view for selected joints.
- Use Skeleton > Retargeting > Show All Labels and Skeleton > Retargeting > Hide All Labels to turn on or off the visibility of all joint labels in your scene view.
- Set the visibility of a single joint's label from the Attribute Editor. To toggle the display of a single joint's label in the scene view, select the joint, open the Attribute Editor, and turn Draw Label in the Joint Labelling section on or off.

## Remove the labels from my joints

### To unlabel a joint

1 Select the joint you want to unlabel.

2 Select Skeleton > Retargeting > Joint Labelling.

The Joint Labelling menu appears.

3 Select None.

The current joint's label is removed and is replaced with *None*.

## Retarget animation from one skeleton to another

| | |
|---|---|
| **Note** | When your skeletons have an IK, FK, and an IK/FK skeleton, then choose just one of the types of skeletons on which to perform the retarget. For example, *source* FK skeleton > *target* FK skeleton. |

### To retarget animation from one skeleton to another

1 Set the neutral poses for your *source* and *target* skeletons.

See "Set a skeleton's neutral pose" on page 93.

2 Make sure that all your source and target skeleton's joints are labelled properly.

See "Label my skeleton's joints" on page 94.

3 Select the root of the source character, and then Shift+click the root of the target character to include it in your selection.

The first root you select is considered the *source*, and the second root you select is considered the *target*.

4. Select Skeleton > Retargeting > Retarget Skeleton > ❐.

   The Retarget Skeleton Options window appears. See ”Skeleton > Retargeting > Retarget Skeleton” on page 116.
5. Set the Neutral Pose, Lower Body, Upper Body, and Time options as desired.
6. Click the Retarget button.

The animation on the source skeleton is retargeted to the target skeleton.

**What went wrong?**

# Character Setup

## Troubleshooting skeletons

## Possible animation retargeting problems

### Jittering motion in one or more limbs

#### Cause

This may occur when the difference between the orientation of the source and target skeleton’s limbs at their neutral poses is too great.

For example, if a source and target skeletons neutral pose resembled a marching pose, and the source skeleton had its *left leg forward* and its *right leg back* and the target skeleton had its *right leg forward* and its *left leg back*, then the retarget solver would not be able to compensate for the extreme differences between the source and target skeleton’s neutral poses.

#### Solution

1. Rotate the limbs of your source and target skeletons so that they have similar orientations.
2. Set a new neutral pose for both skeletons.

   See ”Set a skeleton’s neutral pose” on page 93.
3. Retarget the animation.

   See ”Retarget animation from one skeleton to another” on page 97.

### Arms and/or legs are crossed or at strange angles

#### Cause

This may occur when your source and target skeleton's are not facing in the same direction at their neutral poses. For example, at their neutral poses the source skeleton is facing down the x-axis and the target skeleton is facing down the y or z-axis.

#### Solution

1 Position your source and target skeletons so that they are both facing in the same direction.
2 Set a new neutral pose for both skeletons.
   See "Set a skeleton's neutral pose" on page 93.
3 Retarget the animation.
   See "Retarget animation from one skeleton to another" on page 97.

### A spine joint in the skeleton bends 90 degrees after the solve

#### Cause

This may occur when one or both the source and target skeletons are missing head or neck joints labels.

#### Solution

1 Make sure that there are head or neck labels on both your source and target skeletons. If there are none, then label the appropriate joints as heads or necks.
2 Set a new neutral pose for both skeletons.
   See "Set a skeleton's neutral pose" on page 93.
3 Retarget the animation.
   See "Retarget animation from one skeleton to another" on page 97.

## Retargeting error messages

For information of retargeting animation, see "Animation retargeting" on page 56 and "Retarget animation from one skeleton to another" on page 97.

## Error: Skipping arm retargeting since arms are not labelled

### Cause

Some of the joints in your *arm* joint chains are not labelled. All joints, including all those in the hands, must be labeled for a successful retarget.

See ”Label my skeleton’s joints” on page 94 and ”Skeleton > Retargeting > Joint Labelling” on page 116.

### Solution

Each *arm* joint chain must have joints labelled as a Shoulder, Elbow, and Hand.

## Error: Skipping leg retargeting since legs are not labelled

### Cause

Some of the joints in your *leg* joint chains are not labelled. All joints, including all those in the feet, must be labeled for a successful retarget.

### Solution

Each *leg* joint chain must have joints labelled as a Hip, Knee, and Foot.

## Error: Skipping legs and arms since labels are not found

### Cause

Some of the joints in your *arm* and *leg* joint chains are not labelled. All joints, including all those in the hands and feet, must be labeled for a successful retarget.

### Solution

Each *arm* and *leg* joint chain must have joints labelled as a a Shoulder, Elbow, and Hand or a Hip, Knee, and Foot.

## Error: Error finding matching corresponding joint in *n* chain

*n* represents a limb type such as *Left Leg* or *Center Head*.

#### Cause

One of the joint hierarchies (source or target) that you selected for the retarget has a different number of limbs than the other. The retarget solver can not solve between skeletons with different numbers of limbs (for example, single-headed source skeleton > two-headed target skeleton).

#### Solution

Make sure that the skeletons that you are performing the retarget on have the same number of limbs.

**Note** If your skeletons have an IK, FK, and/or an IK/FK skeleton, then choose just one of the types of skeletons on which to perform the retarget. For example, source FK skeleton > target FK skeleton.

# Reference Menus

## Animation menu set

### Animate >

### IK/FK Keys >

### Animate > IK/FK Keys > Set IK/FK Key

This menu item sets keys on all the current IK handle's keyable attributes and all the joints in its IK chain. When you use Set IK/FK Key, Maya performs additional operations to ensure a smooth transition between IK and FK.

You can use this menu item instead of Animate > Set Key when you want to key IK and FK animation on the same joint chain. We recommend that you use Set IK/FK keys when animating a joint chain with both forward and inverse kinematics.

Also, when you key attributes using Set IK/FK Key, the resulting animation curves display as solid or dotted lines. The animation curves appear as *solid* lines when the animation is fully driven by IK, and as *dotted* lines when the animation is fully driven by FK. For more information, see *"Graph Editor overview"* in the Keyframe Animation chapter of the *Character Setup* guide.

## Animate > IK/FK Keys > Enable IK Solver

This menu item turns the current IK handle's Solver Enable attribute on or off. The status of the Solver Enable attribute determines whether an IK handle's solver is active or inactive.

When Solver Enable is on, you can only pose your joint chain using the IK handle. When Solver Enable is off, you can only pose your joint chain by directly manipulating its individual joints (FK).

You turn Enable IK Solver on and off to switch between inverse and forward kinematics when posing a joint chain using both animation methods. See "Posing skeletons" on page 43.

**Tip** To display whether a selected IK handle's Solver Enable attribute is on or off, turn on Display > Heads Up Display > Animation Details.

## Animate > IK/FK Keys > Connect to IK/FK

This menu item links the selected object to the current IK handle by adding the read-only attribute *Ik Blend* to the object. See "Ik Blend" on page 130.

Once the object is connected, you can select the object and use Animate > IK/FK Keys > Set IK/FK Key to set keys on the IK handle and all the joints in its joint chain. See "Animate > IK/FK Keys > Set IK/FK Key" on page 101.

You can also constrain or parent the IK handle to the object if you want to use the object to manipulate the IK handle. Objects that control and drive the transformations of IK handles are called *control objects*.

**Note** If you turn on Display > Heads Up Display > Animation Details when the object is selected, the Ik Blend value for the IK handle appears in the scene view.

## Animate > IK/FK Keys > Move IK to FK

This menu item moves the current IK handle to the last joint in its joint chain.

In a typical FK/IK rig, the IK handle does not move with the skeleton during FK manipulation and thus appears out of sync. This can cause confusion when there are many IK handles. When animating with IK and FK, you can use Move IK to FK to synchronize the IK handle's position with the end effector's position.

For example, before switching from FK to IK in your animation, you need to move the IK handle to the FK bone's position so that the animation does not jump when the FK switches to IK.

| | |
|---|---|
| **Note** | If the IK handle is grouped or constrained in a control hierarchy, use Animate > IK/FK Keys > Connect IK to FK in conjunction with Move IK to FK to move the entire hierarchy. |

## Animate > Set Key

If you turn on Set IK/FK Keys in the options window for Animate > Set Key > ❐, you can subsequently use Animate Key > Set Key as an alternative to Animate > IK/FK Keys > Set IK/FK Key.

If you use Animate Key > Set Key for an object that's not an IK handle, a joint controlled by an IK handle, or an object connected to an IK handle, the Set IK/FK Keys option is ignored.

**Skeleton >**

## Skeleton > Joint Tool

### Related topics

- ❖ "Joints and bones" on page 34
- ❖ "Create a joint or joint chain" on page 57

### Skeleton > Joint Tool > ❐

#### Degrees of Freedom

Specifies which of the joint's local axes the joint can rotate about during inverse kinematics posing. Click X, Y, or Z. The default setting allows the joint to rotate about all three of its local axes during IK posing.

#### Auto Joint Orient

Sets the orientation of the local axes for all the joint's you create with the Joint Tool. The default selection is XYZ.

The none selection specifies that the joint's local axis have the orientation of the world axis.

The other selections specify that the joint's local axis be oriented so that the first axis points into the joint's bone, the third axis points sideways from the joint and its bone connecting the child joint, and the second axis points at right angles to the first axis and third axes. If the joint has more than one child joint, the first axis points into the bone that connects to the child joint that was created first. All three axes are aligned according to the right hand rule.

For example, for *XYZ* the positive X-axis points into the joint's bone and towards the joint's first child joint, the Z-axis points sideways from the joint and its bone connecting the child joint, and the Y-axis points at right angles to the X-axis and Z-axis.

> Note — If you want a joint to rotate about a particular axis, that axis must not be the axis that points into the joint's bone. For example, a joint can not rotate about its local X-axis if its orientation is set to XYZ.

See "Orient a joint's local axes manually" on page 63 and "Orient a joint's local axes automatically" on page 64.

### Scale Compensate

When this option is on, the joints you create are not scaled automatically when you scale joints above them in the skeleton's hierarchy. Default is on.

Scale Compensate can prevent undesirable shearing effects that can occur after you've skinned a character and then decide to scale a joint along one or two of its axes. Also, it can make it easier for you to change the length of individual bones.

### Auto Joint Limits

When this option is on, Maya automatically limits the extent a joint can rotate about its axes according to the angles at which you build the skeleton's joints. Also, the smaller inner angle of a joint rounded off to 180 degrees is set as the allowable range of rotation.

For example, when you create a knee joint, if you create the joint slightly bent back, the joint will not be able to swing the lower leg bone forward of the upper leg bone, nor will it be able to wobble from side to side. The joint will not be able to rotate in any other way except through the inner angle rounded off to 180 degrees.

> **Note**
> - Auto Joint Limits does not change the joint's Degrees of Freedom setting. See "Degrees of Freedom" on page 103.
> - This setting does not apply to a joint chain's start and end joints.

#### Create IK Handle

When this option is on, an IK handle is automatically created for any joint chain you draw and the IK Handle Options section of the Joint Tool settings is available.

The IK handle that is automatically created will run from the joint chain's start joint to its end joint.

> **Note** It is a more common practice to create a complete skeleton, and then add IK handles to the skeleton where needed.

#### IK Handle Options

Specifies the creation options for the IK handle that is automatically created when Create Ik Handle is on. These options are available only when Create IK Handle is on. For more information on these options, see "ikHandle" on page 127.

# Skeleton > IK Handle Tool

## Related topics

- "IK handles" on page 46
- "IK solvers" on page 47
- "Specify the IK Handle Tool's settings" on page 69
- "Create IK handles" on page 69

## Skeleton > IK Handle Tool > ❐

### IK Handle Settings

#### Current Solver

This drop-down menu specifies the type of IK solver your IK handle will have. The default selections are the ikRPsolver and ikSCsolver. See "IK solvers" on page 47.

#### Autopriority

When this option is on, Maya sets the IK handle’s Priority automatically upon creation. Maya assigns the IK handle’s Priority based on where the IK handle’s start joint is in the skeleton’s hierarchy. For example, if the IK handle starts at the root joint, the Priority is set to 1. If the IK handle starts at a joint just below the root, the Priority is set to 2, and so on. Autopriority is off by default. See ”Priority” on page 128.

#### Solver Enable

When this option is on, the IK solver (specified by Current Solver) will be active upon creation. Solver Enable is on by default so that you can pose the joint chain with the IK handle immediately. See ”IK solvers” on page 47.

#### Snap Enable

When this option is on, IK handle will snap back to the IK handle’s end joint’s position. Snap Enable is on by default.

#### Sticky

When this option is on, the IK handle will stick to its current position and orientation when you pose the skeleton using other IK handles or when translating, rotating, or scaling individual joints. Sticky is off by default.

#### Weight

Sets the weight value for the current IK handle.

The weight value, in conjunction with the current distance between the IK handle’s end effector and its goal, serve to prioritize the solutions of the current IK chain and those of its other IK handles that have the same Priority settings. See ”Priority” on page 128.

When the end effector of two or more IK handles with the same Priority cannot reach their goals simultaneously, the IK handle whose end effectors are furthest from their goals and whose weights are greatest are solved first.

#### POWeight

Specifies whether the current IK handle’s end effector favors reaching its goal’s orientation or position.

When this attribute is set to 1, the end effector tries to reach the IK handle’s position. When this attribute is set to 0, the end effector only tries to reach the IK handle’s orientation. A value of 0.5 specifies that the end effector equally favors reaching both the position and orientation as closely as possible. Default is 1.000.

# Skeleton > IK Spline Handle Tool

## Related topics

- ”IK handles” on page 46
- ”IK solvers” on page 47
- ”Specify the IK Spline Handle Tool’s settings” on page 75
- ”Create spline IK handles” on page 76

## Skeleton > IK Spline Handle Tool > ❐

### IK Spline Handle Settings

#### Root on Curve

When this option is on, the start joint of the IK spline handle is constrained the curve. You can drag an offset manipulator to slide the start joint (and its children) along the curve.

When this option is off, you can move the start joint away from the curve. The start joint is no longer constrained to the curve. Maya ignores the Offset attribute, and no offset manipulator exists at the start joint.

This option is also available from the ”IK Handle Attributes” on page 127.

#### Auto Create Root Axis

When this option is on, a parent transform node above the start joint in the scene hierarchy is created with the spline IK handle. You can avoid unexpected start joint flipping by moving and rotating this transform node rather than the start joint. See ”Prevent flipping in the spline IK chain” on page 82 for details.You can turn on this option only when Root on Curve is off.

Auto Create Root Axis is exclusive to the IK Spline Handle Tool settings.

#### Auto Parent Curve

If the start joint of the spline IK handle has a parent, this setting makes the spline IK NURBS curve a child of that parent. Therefore, the curve and joint chain moves with the transformations of the parent.

If you create a handle that starts at a joint in the chain lower than the root joint of your skeleton, turn this option on to ensure that the joint chain moves with the transformations of its parent joint.

Auto Parent Curve is exclusive to the IK Spline Handle Tool settings.

### Snap Curve to Root

When this option is on, the start of the curve snaps to the position of the start joint. The joints in the chain rotate to adapt to the shape of the curve.

Snap Curve to Root affects the spline IK handle only when Auto Create Curve is off in the tool settings and you already created your own curve.

Turn Snap Curve to Root off if you want to use the curve as a fixed path for your joint chain

Snap Curve to Root is exclusive to the IK Spline Handle Tool settings.

### Auto Create Curve

When this option is on, Maya creates a NURBS curve when you create the spline IK handle. The curve that is created follows the path of the joint chain.

If Auto Create Curve is on and Auto Simplify Curve is *off*, creating the spline Ik handle creates a complicated curve that passes through all the joints in the spline IK chain.

If Auto Create Curve is on and Auto Simplify Curve is *on*, creating the handle automatically creates a simplified curve that has a shape similar to the joint chain. The higher the Number of Spans, the closer the curve matches the joint chain. The curve has a curve degree of 3 (cubic).

If the spline IK joint chain is part of an existing skeleton, you should turn Auto Create Curve on. If you are using a curve as a path for sliding the joint chain, you should turn Auto Create Curve off.

When Auto Create Curve is off, you must create a NURBS curve for the joint chain.

Auto Create Curve is exclusive to the IK Spline Handle Tool settings.

### Auto Simplify Curve

When this option is on, Maya automatically creates a NURBS curve with the specified Number of Spans. The number of spans corresponds to the number of CVs in the curve. The curve has a curve degree of 3 (cubic).

If you create a curve with few CVs, your control of the curve's shape and skeleton's movement will be less precise, but you will be able to easily manipulate the curve and its joint chain. With fewer CVs, you will spend less time selecting and dragging CVs, and you are more likely to have a smooth curve.

Auto Simplify Curve is available only when Auto Create Curve is on.

Auto Simplify Curve is exclusive to the IK Spline Handle Tool settings.

### Number of Spans

Number of Spans specifies the number of CVs for the curve that Maya automatically creates when you create the spline IK handle.

The ratio between spans and CVs is as follows: number of CVs = the number of spans + 3.

| Number of Spans | Number of CVs |
|---|---|
| 1 | 4 |
| 2 | 5 |
| 3 | 6 |
| 4 | 7 |

Number of Spans is available only when Auto Create Curve is on.

Number of Spans is exclusive to the IK Spline Handle Tool settings.

### Root Twist Mode

When this option is on, manipulating the twist manipulator at the end joint will twist the start joint slightly with the other joints.

When this option is off, the start joint does not twist. If you want to turn the start joint, you have to use the roll manipulator. See ”Spline IK solver” on page 52.

Root Twist Mode is also available in the ikHandle node’s attributes. See ”Root Twist Mode” on page 132.

# Skeleton > Insert Joint Tool

Lets you add an additional joint to an existing joint chain. See ”Insert joints” on page 58.

## Related topics

❖ ”Joints and bones” on page 34
❖ ”Insert joints” on page 58

# Skeleton > Reroot Skeleton

Designates the current joint as the parent or *root* of its hierarchy. See ”Reroot skeletons” on page 65.

## Related topics

- ”Joints and bones” on page 34
- ”Joint chains” on page 35
- ”Skeleton hierarchy” on page 36
- ”Reroot skeletons” on page 65

# Skeleton > Remove Joint

Deletes the current joint and unites the remaining joints and bones into a single joint chain. See ”Remove joints” on page 68.

## Related topics

- ”Joints and bones” on page 34
- ”Remove joints” on page 68

# Skeleton > Disconnect Joint

Breaks the skeleton at the current joint, separating the skeleton into two joint chains. See ”Disconnect joints” on page 67.

## Related topics

- ”Joints and bones” on page 34
- ”Disconnect joints” on page 67

# Skeleton > Connect Joint

Connects the selected joints.

## Related topics

- ”Connecting joint chains and skeletons” on page 39
- ”Connect joints” on page 59

## Skeleton > Connect Joint > ❐

Lets you specify whether to create a new bone between the selected joints upon connection, or just combine the selected joints.

Mode

| | |
|---|---|
| Connect Joint | Connects the selected joints and moves the joint chain of the first joint to the position of the second joint. After connecting the joints, use Remove Joint to delete the extra joint. See ”To connect joint chains by combining joints” on page 59. |
| Parent Joint | Connects the selected joints by inserting a new bone between them. ”To connect joint chains by connecting selected joints with a bone” on page 60. |

## Skeleton > Mirror Joint

Duplicates the current parent’s joint chain across the specified plane. See ”Mirroring joint chains” on page 38 and ”Mirror joint chains” on page 62.

### Related topics

- ”Mirroring joint chains” on page 38
- ”Mirror joint chains” on page 62

### Skeleton > Mirror Joint > ❐

Lets you specify the plane to mirror the joints across, as well as the orientation of the resulting duplicate joints.

Mirror Across

Default is XY.

| | |
|---|---|
| XY | Mirrors the joint across the XY plane. |
| YZ | Mirrors the joint across the YZ plane. |
| XZ | Mirrors the joint across the XZ plane. |

Mirror Function

| | |
|---|---|
| Behavior | When this option is on, the mirrored joints have the opposite orientation as the original, and the local rotation axis of each joint points in the opposite direction of their counterparts.<br><br>This setting is handy for animating opposing movements in a pair of counterpart limbs. For example, if you select a pair of ankles and use the Rotate tool on both at the same time, you can rotate the feet to point symmetrically inward or outward with a single manipulation. |

| | |
|---|---|
| Orientation | When this option is on, the mirrored joints have the same orientation as the original joints.<br>With this setting, you can copy animation from one limb to another and get identical behavior. For example, if you want to animate a skier racing down a slope with legs turning in the same direction, use this setting. |

Replacement names for duplicated joints

| | |
|---|---|
| Search For | Lets you specify the name identifier of the joints in the mirrored joint chain that you want to replace.<br>For example, if you type “right_” in this field, all of the mirrored joints with the “right_” prefix in their name will change to what is specified in the Replace With field. |
| Replace With | Lets you specify the names with which to replace the mirrored joint name identifiers specified in the Search For field.<br>For example, if you type “left_” in this field, all the “right_” prefixes of the mirrored joint names will be replaced with “left_”. |

# Skeleton > Orient Joint

## Related topics

- ”Local axes orientation” on page 42
- ”Orient a joint’s local axes manually” on page 63
- ”Orient a joint’s local axes automatically” on page 64

## Skeleton > Orient Joint > ❐

### Orientation

Specifies the orientation of a joint’s local axis. The default selection is XYZ.

The none selection specifies that the joint’s local axis have the orientation of the world axis.

The other selections specify that the joint’s local axis be oriented so that the first axis points into the joint’s bone, the third axis points sideways from the joint and its bone connecting the child joint, and the second axis points at right angles to the first axis and third axes. If the joint has more than one child joint, the first axis points into the bone that connects to the child joint that was created first. All three axes are aligned according to the right hand rule.

For example, for *XYZ* the positive X-axis points into the joint's bone and towards the joint's first child joint, the Z-axis points sideways from the joint and its bone connecting the child joint, and the Y-axis points at right angles to the X-axis and Z-axis.

| Note | If you want a joint to rotate about a particular axis, that axis must not be the axis that points into the joint's bone. For example, a joint can not rotate about its local X-axis if its orientation is set to XYZ. |
|---|---|

#### Hierarchy

| | |
|---|---|
| Orient child joints | When this option is on, the settings for Orient Joint will affect all the joints below the current joint in the skeleton's hierarchy.<br>When this option is off, only the current joint is affected by the Orient Joint settings. |

#### Scale

| | |
|---|---|
| Reorient the local scale axes | When this option is on, the local scale axes of the current joint are also reoriented. |

## Skeleton > Set Preferred Angle

Sets the current skeleton's preferred angles. See "Preferred angles" on page 42 and "Set preferred angles" on page 66.

### Related topics

- "Preferred angles" on page 42
- "Set preferred angles" on page 66

### Skeleton > Set Preferred Angle > ❐

Lets you specify whether to set the preferred angle for the current joint or for all the joints from the current joint to its chain's end joint.

#### Options

| | |
|---|---|
| Selected Joint | Sets the current angle of the selected joint as its preferred angle. |

| | |
|---|---|
| Recursive | Sets the current angles of all the joints from the selected joint to the joint at the end of the chain as their preferred angles. |

# Skeleton > Assume Preferred Angle

Returns the current skeleton’s joints to the angles they possessed when their preferred angles were initially set. This function can also be accessed from a marking menu when you press the right mouse button on a joint chain or skeleton.

## Related topics

- ”Preferred angles” on page 42
- ”Set preferred angles” on page 66

## Skeleton > Assume Preferred Angle > ❐

Lets you specify whether to reset the current joint’s angle to its preferred angle or to reset all the joints from the current joint to its chain’s end joint to their preferred angles.

### Options

| | |
|---|---|
| Selected Joint | Resets the current angle of the selected joint to its preferred angle. |
| Recursive | Resets the current angles of all the joints from the selected joint to the joint at the end of the chain to their preferred angles. |

# Skeleton > Enable IK Handle Snap

Causes the IK handle to snap back to its end joint’s position. By default, Enable IK Handle Snap is on.

## Related topics

- ”IK handles” on page 46
- ”Snap Enable” on page 106

# Skeleton > Enable IKFK Control

When Enable IKFK Control is on, you can rotate the joints in an IK chain whether or not the IK handle has keys. Also, Enable IKFK Control disables *Enable IK Handle Snap*. See ”Skeleton > Enable IK Handle Snap” on page 114.

When Enable IKFK Control is off, you can not rotate the joints in an IK chain unless you disable its IK handle.

## Skeleton > Enable Selected IK Handles

Turns the editability of the selected IK handles on, and sets their Ik Blend values (in their IK Solver Attributes) to 1.000. When Ik Blend is 1.000, the IK animation is in full control of the skeleton.

### Related topics

❖ "IK handles" on page 46
❖ "Skeleton > Disable Selected IK Handles" on page 115

## Skeleton > Disable Selected IK Handles

Turns the editability of the selected Ik handles off, and sets their IK Blend values (in their IK Solver Attributes) to 0.000. When Ik Blend is 0.000, the FK animation is in full control of the skeleton.

### Related topics

❖ "IK handles" on page 46
❖ "Skeleton > Enable Selected IK Handles" on page 115

### **Retargeting >**

## Skeleton > Retargeting > Set Neutral Pose

Sets the current skeleton's pose as its neutral pose. The neutral pose is the common pose you want both the source and target skeletons to be in when retargeting animation data.

The neutral pose maintains the distinct characteristics of your source and target skeletons. This lets you differentiate your skeletons when retargeting animation. For example, if you are retargeting animation from an android to a hunchbacked character, the neutral pose prevents the android character's stiff back from overriding the hunchback's slouch.

See "Set a skeleton's neutral pose" on page 93.

## Skeleton > Retargeting > Go to Neutral Pose

Makes the current skeleton assume its predefined neutral pose. If the skeleton you want to perform this operation on does not have a neutral pose, then an error message appears.

**Note** Skeletons that have neutral poses have a *retargetPose* tab on their root joint's node in the Attribute Editor.

## Skeleton > Retargeting > Joint Labelling

The Joint Labelling menu lets you assign standard, consistent labels to the joints of your source and target skeletons.

Assigning consistent labels to your source and target skeletons is required to properly retarget animation from one skeleton to another.

### Related topics

- "Label my skeleton's joints" on page 94
- "Retargeting workflow" on page 93

## Skeleton > Retargeting > Toggle Selected Labels

Shows or hides the joint labels in the scene view for the selected joints.

## Skeleton > Retargeting > Show All Labels

Shows all joint labels in the scene view.

## Skeleton > Retargeting > Hide All Labels

Hides all joint labels in the scene view.

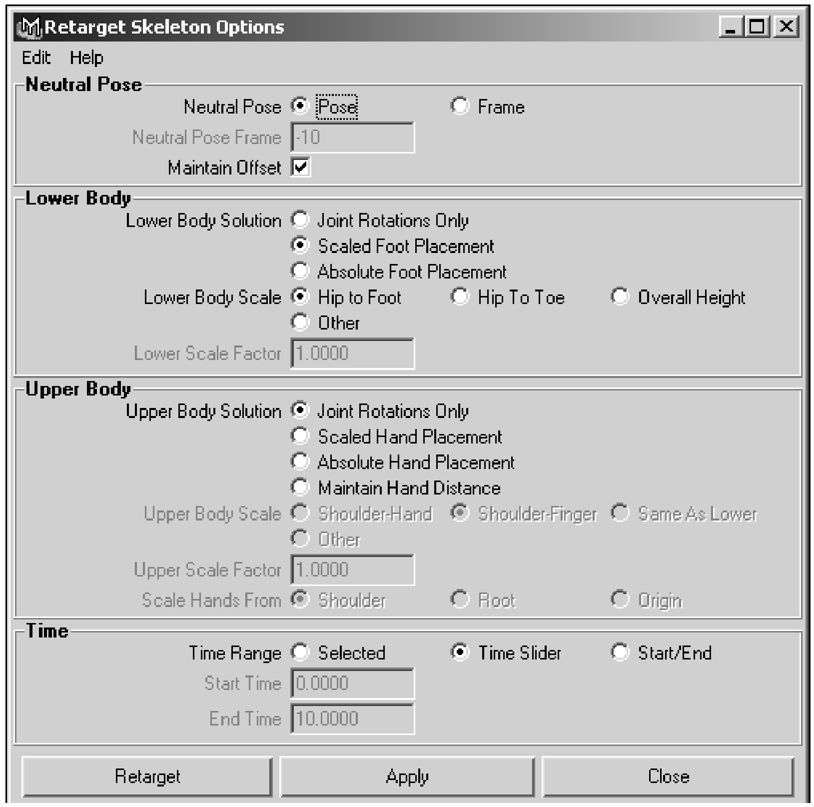
## Skeleton > Retargeting > Retarget Skeleton

Retargets the animation on one skeleton (source) to that of another (target) skeleton.

### Skeleton > Retargeting > Retarget Skeleton > ❐

The Retarget Options window lets you set the *retargeting* solver's options.

The retargeting solver is what looks at the animation on the *source* skeleton's joints and retargets them to the corresponding, labelled joints of the *target* skeleton.




## Neutral Pose

Specifies whether a defined neutral pose (the pose set with Skeleton > Set Neutral Pose) will be used to relate the source and target skeletons during the retarget or whether it will be a pose at a specified frame. See "Set a skeleton's neutral pose" on page 93 and "Skeleton > Retargeting > Set Neutral Pose" on page 115.

### Neutral Pose

| | |
|---|---|
| Pose | Sets the neutral pose as the pose you set with Skeleton > Set Neutral Pose. If you have already set a neutral pose for each skeleton using Set Neutral Pose, use this option. Pose is on by default. |

| | |
|---|---|
| Frame | Sets the poses of the source and target skeletons at the specified frame as their neutral poses. Use this option if you have not previously set the neutral poses for your skeletons using Skeleton > Set Neutral Pose. |

#### Neutral Pose Frame

Specifies the frame to use to set the neutral poses of the source and target skeletons.

#### Maintain Offset

Maintains the source and target skeleton's global positions so that the target skeleton does not move to the position of the source skeleton when you retarget.

### Lower Body

#### Lower Body Solution

Lets you choose how the lower body animation of the source skeleton is retargeted to the lower part of the target skeleton.

| | |
|---|---|
| Joint Rotation Only | When on, only joint rotations are retargeted. Use this option if you are mapping animation between rigs that have the same bone lengths. For example, if you are retargeting animation from a low resolution rig to a high resolution rig of the same skeleton. |
| Scaled Foot Placement | When on, all animation data is retargeted. Use this option if you are retargeting motion from one skeleton to a completely different skeleton. For example, retargeting animation from a giant to a dwarf.<br><br>This option takes into account the position and orientation of the source's feet, and solves the retarget by translating/orienting the target's feet like the source's feet. This ensures that the target skeleton's foot placement looks natural and that its feet move in relation to its proportions-not those of the source skeleton. This option is on by default. |
| Absolute Foot Placement | When on, ensures that the target skeleton's feet land at the same points as those of the source. Use this option when you need the target skeleton's feet to |

land at very specific locations. For example, you would use this if you are retargeting a fight sequence or dance routine.

### Lower Body Scale

Specifies how the animation on the hip, knee, ankle, foot, and toe joints of the source skeleton is retargeted to the corresponding leg joints of the target skeleton.

For example, if you are retargeting animation from a giant to a dwarf, you should select the Hip to Foot option because it best handles the retargeting of animation between skeletons whose legs have different bone lengths.

Only available when Scaled Foot Placement in on.

| | |
|---|---|
| Hip to Foot | Calculates the lower body scale for target using the source's hip to foot scale. |
| Hip to Toe | Calculates the lower body scale for the target using the source's hip to toe scale. |
| Overall Height | Calculates the lower body scale for the target using the overall height of the source. |
| Other | Calculates the lower body scale for the target using the current Lower Scale Factor value. |
| Lower Scale Factor | Lets you specify a custom scale factor for the target's lower-body animation. Only available when Other in on. |

## Upper Body

### Upper Body Solution

Lets you choose how the upper body animation of the source skeleton is retargeted to the upper part of the target skeleton.

| | |
|---|---|
| Joint Rotation Only | When on, only joint rotations are retargeted. Use this option if you are mapping animation between rigs that have the same bone lengths. For example, if you are retargeting animation from a low resolution rig to a high resolution rig of the same skeleton. |
| Scaled Hand Placement | When on, all animation data is retargeted. Use this option if you are retargeting motion from one skeleton to a completely different skeleton. For example, retargeting animation from a giant to a dwarf. |

This option takes into account the position and orientation of the source's hands, and solves the retarget by orienting and positioning the target's hands like the source's hands. This ensures that the target skeleton's hand placement looks natural and that its hands move in relation to its proportions-not those of the source skeleton. This option is on by default.

Absolute Hand Placement
: When on, ensures that the target skeleton's hands move to the exact same points as those of the source. Use this option when you need the target skeleton's hands to move to very specific positions. For example, you would use this if you are retargeting a fight sequence or dance routine.

Maintain Hand Distance
: Maintains the distance between the hands and the rest of the body for the target skeleton. Use this option when the source and target skeletons are carrying objects. For example, basketball players holding balls, waiters holding trays and so on.

### Upper Body Scale

Specifies how the animation on the shoulder, elbow, wrist, hand, and finger joints of the source skeleton is retargeted to the corresponding arm joints of the target skeleton.

Only available when Scaled Hand Placement in on.

Shoulder-Hand
: Calculates the upper body scale for target using the source's shoulder to hand scale.

Shoulder-Wrist
: Calculates the upper body scale for the target using the source's shoulder to wrist scale.

Calculates the upper body scale for the target using the lower body scale of the source.

Other
: Calculates the upper body scale for the target using the current Upper Scale Factor value.

Upper Scale Factor
: Lets you specify a custom scale factor for the target's upper-body animation. Only available when Other in on.

| | |
|---|---|
| Scale Hand From | Specifies what pivot is used in the calculations for scaling the target skeleton's hands in respect to the source skeleton's hands. Shoulder is selected by default.<br>Only available when Scale Hand Placement is on. |

### Time

#### Time Range

**Warning** Any animation on the target skeleton within the specified time range will be over written by the source skeleton's animation data.

| | |
|---|---|
| Selected Time Range | Sets the retargeting time range to the range of time selected (using Shift-drag) in the time slider. |
| Time Slider | Sets the retargeting time range to the time sliders current range. |
| Start/End | Sets the retargeting time range to the specified Start Time and End Time. |

## Skeleton > Retargeting > Rename Joints From Labels

Changes the current joint's name to its joint label.

For example, if a joint is named *joint2* and its label is *Shoulder(L)*, then the joint's name would change to LeftShoulder.

## Skeleton > Retargeting > Label Based On Joint Names

Changes the current joint's label to its joint name.

For example, if a joint is labeled as *Shoulder(L)* and its joint name is *Right_Hip*, then the joint's label would change to Hip(R).

# Nodes

## Skeleton and IK nodes

## General skeleton and IK node attributes

### Transform Attributes

See *General Attributes* in the Basics guide.

### Pivots

Specifies whether to display the IK handle's rotate pivot and scale pivot. The Local Space and World Space sections specify the pivot positions in local space (relative to the IK handle) and world space.

### Limit Information

Specifies the transform limits for the current joint or IK handle.

The Min and Max checkboxes turn on the Min or Max limits and the < and > icon buttons change their corresponding Min or Max limit values to the Current value.

#### Translate

| | |
|---|---|
| Trans Limit X | Specifies translation limits on the joint's local X-axis. |
| Trans Limit Y | Specifies translation limits on the joint's local Y-axis. |
| Trans Limit Z | Specifies translation limits on the joint's local Z-axis. |

#### Rotate

| | |
|---|---|
| Rot Limit X | Specifies rotation limits about the joint's local X-axis. |
| Rot Limit Y | Specifies rotation limits about the joint's local Y-axis. |
| Rot Limit Z | Specifies rotation limits about the joint's local Z-axis. |

#### Scale

| | |
|---|---|
| Scale Limit X | Specifies scaling limits along the joint's local X-axis. |
| Scale Limit Y | Specifies scaling limits along the joint's local Y-axis. |
| Scale Limit Z | Specifies scaling limits along the joint's local Z-axis. |

### Display

See *Display* in the Basics guide.

## Node Behavior

See *"Node behavior"* and *Set a node's update state* in the Basics guide.

# joint

For more information, see Help > Node and Attribute Reference.

## Transform Attributes

See *Transform attributes* in the Basics guide.

## Joint

### Draw Style

Specifies how the current joint is drawn in the scene view. These attributes are only applicable to the highest joint in a joint chain. For example, the highest joint in an entire skeleton is the root.

| | |
|---|---|
| Bone | When this attribute is on, the current joint's connections to neighboring joints are drawn as bones in the scene view. This is the default. |
| Box | When this attribute is on, the current joint's connections to neighboring joints are drawn as a box in the scene view. |

### Degrees of Freedom

Specifies which of its local axes the joint can rotate about during inverse kinematics (IK) posing and animation. X, Y, and Z are on by default.

### Stiffness

Specifies the current joint's resistance to rotation during posing. You set stiffness for each axis (X, Y, Z) separately. For example, a wrist joint moves more freely bending toward the forearm than it does from side to side. So, you can set the Stiffness value in the plane perpendicular to the forearm (most often the Y-axis) to reduce its mobility in that plane.

**Note**

- For Stiffness to work properly, the joint chain needs to have more than two bones.
- IK solver calculations for Stiffness can require a little more time than they usually require, so use stiffness only when its effect really needed.
- Joint stiffness with IK is difficult to control. The current joint stiffness algorithm works like a set of springs might work on the joints. While it is easy to say how it works, it is very difficult to control.

The effect of the X, Y and Z Stiffness values is relative to the values assigned to other joints in the joint chain. For example, in a joint chain with two joints, if joint1 has a Stiffness of 1.0 and joint2 has 2.0, joint2 will be twice as stiff as joint1. With stiffness set to 0, no stiffness is specified. In general, this is the recommended setting for all of a skeleton's joints.

Since the Stiffness values for joints are relative to the values for all the other joints in the joint chain, when you set the Stiffness for at least one of the joints, you should also set the Stiffness values for the other joints in the chain so that they do not have the default (0). For example, you might set the Stiffness values for all the joints in the chain to 1, and then set the Stiffness values for the very stiff joints to 2 (twice as stiff as the rest), or 3 (three times as stiff), and so forth. If some of the joints in the chain still have the default setting of 0, the joints may lock up during IK posing.

### Preferred Angle

Specifies how an inverse kinematics joint prefers to rotate during IK posing.

The IK solver often can rotate a joint in a number of different ways in order for the last joint to reach the goal.

Depending on how you want your character to move, some rotations are more appropriate than others. You need to identify the preferred angles for your character's actions. Two types of IK solvers, the single chain IK solver and the rotate plane IK solver, will then give those angles priority over other possible angles during joint rotation. The angles you give priority to are the *preferred angles*.

### Joint Orient

Specifies the orientation of the joint's local rotation axis.

### Segment Scale Compensate

When this attribute is on, the joint compensates for the scaling of its parent joint, and so is not affected.

When this attribute is off, the current joint's scaling (the Scale X, Y, and Z attribute values) is affected by the scaling of its parent joint.

Default is on.

## Joint Labelling

Specifies the label settings for the current joint.

### Side

Specifies a side label for the current joint.

| | |
|---|---|
| Center | Applies the *Center* side label to the selected joint. |
| Left | Applies the *Left* side label to the selected joint. For a symmetrical skeleton, all the joints on the left side of the skeleton should have the Left side label. |
| Right | Applies the *Right* side label to the selected joint. For a symmetrical skeleton, all the joints on the right side of the skeleton should have the Right side label |
| None | Applies the *None* side label to the selected joint. When a joint is labeled None, then it is not included in the retarget. |

### Type

Specifies a type label for the current joint.

| | |
|---|---|
| None | Sets the type label of the current joint to *None*. When a joint is labeled None, then it is not included in the retarget. |
| Root | Sets the type label of the current joint to *Root*. |
| Foot | Sets the type label of the current joint to *Foot*. |
| Hip | Sets the type label of the current joint to *Hip*. |
| Knee | Sets the type label of the current joint to *Knee*. |
| Toe | Sets the type label of the current joint to *Toe*. |
| Spine | Sets the type label of the current joint to *Spine*. |
| Head | Sets the type label of the current joint to *Head*. |
| Neck | Sets the type label of the current joint to *Neck*. |
| Collar | Sets the type label of the current joint to *Collar*. |

| | |
|---|---|
| Shoulder | Sets the type label of the current joint to *Shoulder*. |
| Elbow | Sets the type label of the current joint to *Elbow*. |
| Hand | Sets the type label of the current joint to *Hand*. |
| Finger | Sets the type label of the current joint to *Finger*. |
| Thumb | Sets the type label of the current joint to *Thumb*. |
| PropA | Sets the type label of the current joint to *PropA*. |
| PropB | Sets the type label of the current joint to *PropB*. |
| PropC | Sets the type label of the current joint to *PropC*. |
| Other | Sets the type label specified in the Other Type field to the selected joint. |

OtherType

Lets you define a custom label type for the current joint. For example, using the otherType field you can label a dog skeleton's tail joint as *Tail(None)*. Only available when Type is set to *Other*.

**Warning** The retarget solver only recognizes default label types and may not know how to properly retarget animation between joints that have custom label types.

Whereas if you labeled a joint with a default label type such as *Foot(Left)*, the retarget solver would automatically know the proper behavior for a *foot* (for example, it stays planted on the ground until the knee joint above it in the hierarchy moves or rotates) and would retarget the animation accordingly.

Draw Label

When on, the current joint's label is visible in the scene view.

## Joint Rotation Limit Damping

Sets the damping range for the current joint. You can set joint damping when you want a joint to slow down when it approaches its defined limits. You need at least three joints for damping to work properly.

Min Damp Range

Specifies the angles (relative to the minimum joint limit angles) at which resistance begins to occur.

#### Max Damp Range

Specifies the angles (relative to the maximum joint limit angles) at which resistance begins to occur.

#### Min Damp Strength and Max Damp Strength

Specifies the amount of increasing resistance within the Min Damp Range and the Max Damp Range. Values can range from 0, which takes the joint all the way to its limit with no resistance, to 100, which halts the joint at the outer edge of the damp range. A value of 50 would specify a gradually increasing resistance as the joint rotates past the Min Damp Range angle.

# ikHandle

For more information, see Help > Node and Attribute Reference.

## Transform Attributes

See *General Attributes* in the Basics guide.

## Skeleton Info

#### Start Joint

Displays the name of the start joint IK handle's joint chain. You can click the > icon button to open the Attribute Editor for the start joint.

#### End Effector

Displays the name of the IK handle's end effector. You can click on the > icon button to open the Attribute Editor for the end effector.

## IK Handle Attributes

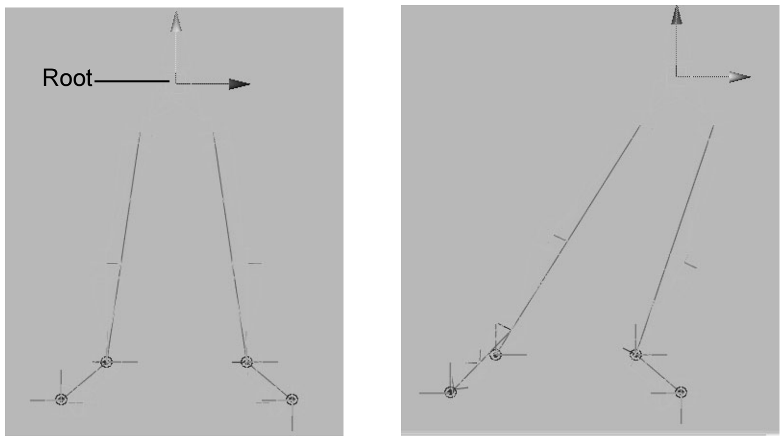
#### Snap Enable

When this attribute is on, the current IK handle will snap back to the position of its end joint. Default is on.

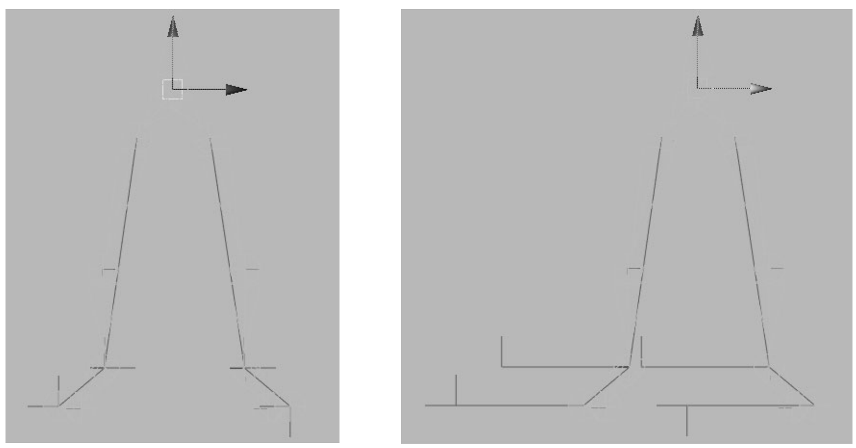
#### Stickiness

When this attribute is on and the current IK handle has no keyframes, the current IK handle will stick to its current position when you pose the skeleton with another IK handle or by translating, rotating, or scaling the joints directly. When this attribute is on and their are keyframes on the current handle, Stickiness is ignored. Default is off.

You can use sticky when posing a joint chain with an IK handle to prevent unwanted joint chain movement. For example, you can turn on Stickiness for the IK handles of a model's legs if you want the feet to stay firmly planted on the floor while you move and pose the hips of the model.




Translating the root along the X-axis with Stickiness *on*

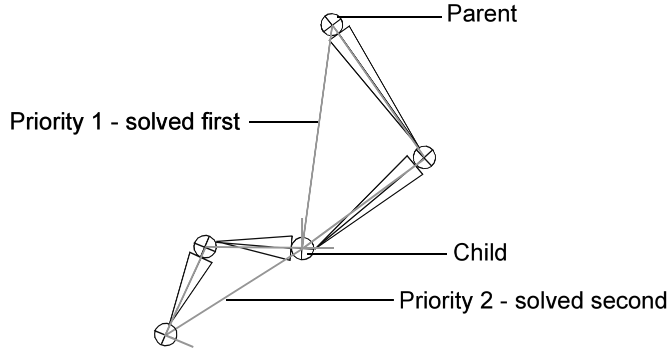


Translating the root along the X-axis with Stickiness *off*

### Priority

Specifies the priority of the current IK handle. Priority is useful only when a joint chain has more than one IK handle. The purpose of Priority is to ensure that the IK handles of a joint chain solve in the correct order so that they properly produce the desired animation.

When you set Priority for the IK handles of a joint chain, Maya calculates the priority for each handle based on its position in the hierarchy. The IK handle with a Priority of 1 has first priority, and will rotate the joints first. An IK handle with a Priority of 2 has second priority, and will rotate the joints next, and so on. Default is 1.




### Weight

Sets the weight value for the current IK handle.

The weight value, in conjunction with the current distance between the IK handle's end effector and its goal, serve to prioritize the solutions of the current IK chain and those of its other IK handles that have the same Priority settings. See "Priority" on page 128.

When the end effector of two or more IK handles with the same Priority cannot reach their goals simultaneously, the IK handle whose end effectors are furthest from their goals and whose weights are greatest are solved first.

Not available for ikSCsolver, ikRPsolver, or ik2Bsolver handles.

### POWeight

Specifies whether the current IK handle's end effector favors reaching its goal's orientation or position.

When this attribute is set to 1, the end effector tries to reach the IK handle's position. When this attribute is set to 0, the end effector only tries to reach the IK handle's orientation. A value of 0.5 specifies that the end effector equally favors reaching both the position and orientation as closely as possible. Default is 1.000.

Not available for the ik2Bsolver or ikRPsolver handles.

## IK Solver Attributes

| Note | |
|---|---|
| Note | • The Ik Fk Control attribute turns off global and local *IK Handle Snap.* |

### Ik Blend

An Ik Blend value of 0.000 sets the animation mode to pure FK, and an Ik Blend value of 1.000 sets the animation mode to pure IK. When the Ik Blend value is a number between 0.000 and 1.000, then the animation on the current skeleton is blended IK and FK. See ”IK/FK blending” on page 55.

> Note In addition to blending IK and FK animation over multiple frames, a blend can occur over a single frame. Blending over a single frame instantly switches IK to FK or FK to IK.

### Ik Fk Control

Lets you manipulate and key the joints of a joint chain that has an animated IK handle. In addition to being located in the local IK Handle Attributes, Ik Fk Control is also present in the global skeleton settings (Skeleton > Enable Ik Fk Control).

### IK Solver

Specifies the type of solver for the IK handle. See ”IK solvers” on page 47. The default options are the following:

| | |
|---|---|
| ikSCsolver | Select the ikSCsolver for an IK single chain handle. |
| ikRPsolver | Select the ikRPSolver for an IK rotate plane handle. |

### Pole Vector

Specifies the position of the pole vector’s end point.

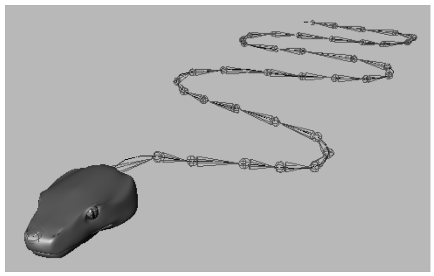
Pole vector is only applicable to IK handle’s that use the rotate plane solver. See ”Rotate Plane solver” on page 49.

> Note Twist Type, Root On Curve, and Root Twist Mode are available when you select Skeletons > IK Spline Handle Tool > ❐. See ”Skeleton > IK Spline Handle Tool” on page 107.

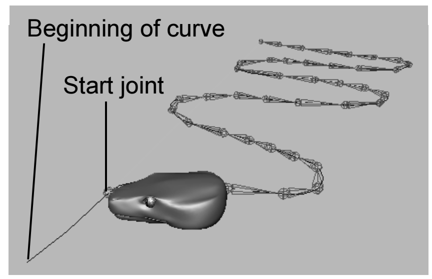
### Offset

This attribute is only available for spline IK handles and when Root On Curve is on.

Sets the amount the root of the IK chain's start joint is offset from the beginning of the spline IK curve.



Offset 0.000

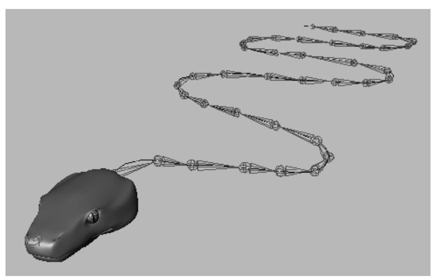



Offset 5.000

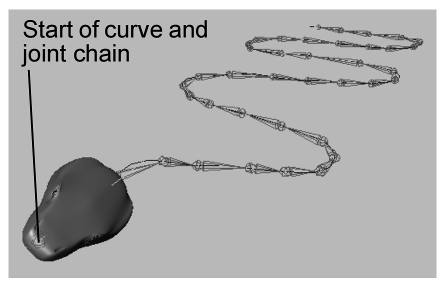
### Roll

This attribute is only available for spline IK handles.

Rolls the entire joint chain from the start joint of the spline IK handle.



Roll 0.000




Roll 50.000

### Twist

Not available for single chain IK handles.

Twists the joint chain from the end joint by the specified amount.

You can also control the rotation of the joint chain by manipulating the twist disc. See "Pose joints with spline IK curves" on page 77.

### Twist Type

This attribute is only available for spline IK handles.

This option specifies how the twist occurs in the joint chain:

| | |
|---|---|
| Linear | Twists all parts evenly. |

| | |
|---|---|
| Ease In | Twists more at the end than the start. |
| Ease Out | Twists more at the start than the end. |
| Ease In Out | Twists more at the middle than at either end. |

See "Skeleton > IK Spline Handle Tool" on page 107.

#### Root On Curve

This attribute is only available for spline IK handles.

See "Skeleton > IK Spline Handle Tool" on page 107.

> Warning If Root On Curve is off in the spline IK handle's IK Solver Attributes, the solver ignores any motion you previously keyed with Offset. Set keys with Root On Curve off or on, not a mixture of both.

#### Root Twist Mode

This attribute is only available for spline IK handles.

See "Skeleton > IK Spline Handle Tool" on page 107.

### Add Advanced Twist Control Attributes

This button reveals additional attributes for the current Spline IK handle.

#### Enable Twist Controls

When this attribute is on, Maya applies an up vector constraint to the local frame of each joint in the current Spline IK handle's joint chain.

#### World Up Type

Specifies the *type* of up vector constraint that is applied to the joints in the Spline IK handle's joint chain.

| | |
|---|---|
| Scene Up | Sets the scene's up direction as the Up vector for all joints. |
| Object Up | Sets the Up vector of each joint as the direction from the joint to the center of a specified object.<br>Type the object's name in the World Up Object field. |
| Object Up (Start/End) | Sets the Up vector of the first and last joints as the direction from the joints to the center of the specified start and end objects. |

| | |
|---|---|
| | Type the names of the start and end joint up objects in the World Up Object and World Up Object 2 fields. The results from the start and end of the chain are interpolated along the chain to provide up vectors for the middle joints. |
| Object Rotation Up | Sets a particular vector in the local space of the specified object as the Up vector for all joints.<br><br>Type the local vectors in the Up Vector fields and the object name in the World Up Object field. The default Up Vector value is (0,1,0). This means that the object's positive Y axis is used as the Up Vector. |
| Object Rotation Up (Start/End) | Sets a particular vector in the local space of the specified objects as the Up vector for the start and end joints in the current chain.<br><br>Type the local vectors and object name for the start joint in the Up Vector and World Up Object fields, and type the local vectors and object name for the end joint in the Up Vector 2 and World Up Object 2 fields. The results from the start and end of the chain are interpolated along the chain to provide up vectors for the middle joints. |
| Vector | Sets a particular vector in worldspace as the up vector for all joints.<br><br>Type the worldspace up vectors in the Up Vector fields. |
| Vector (Start/End) | Sets specific vectors in worldspace as the up vectors for the start and end joints.<br><br>Type the start joint worldspace up vectors in the Up Vector fields and the end joint worldspace up vectors in the Up Vector 2 fields. The results from the start and end of the chain are interpolated along the chain to provide up vectors for the middle joints. |
| Relative | Sets the Spline IK to behave as in previous versions of Maya (pre-Maya 5.0.1). The up vector constraint is not applied to the joints in the chain. |

### Up Axis

Specifies which local joint axis is aligned with the worldspace up vector. Positive Y axis is the default.

If you select any of the *Closest* axes, either the positive or negative axis is used at each joint, depending on which is currently closer to the worldspace up vector. The Closest option lets you preserve the orientation of joint chains whose axes flip every few joints.

**Note** We recommend that you not use *Closest* for animation, because it is prone to cause flipping in the joint chain.

### Up Vector

When the World Up Type is Object Rotation Up or Object Rotation Up (Start/End), the values in these fields specify vectors in the local space of the specified up objects.

When the World Up Type is Vector or Vector (Start/End), the values in these fields specify worldspace world vectors.

The Up Vector fields are for the start joints, and the Up Vector 2 fields are for the end joints.

### World Up Object

When the World Up Type is Object Up, Object (Start/End), Object Rotation Up, or Object Rotation Up (Start/End), the name in this field specifies the object to use for computing the up vector. The name in this field connects the worldspace matrix of the transform with the IK handle.

The World Up Object field is for the start joint, and the World Up Object 2 field is for the end joint.

### Twist Value Type

Sets how the additional, user-defined twist is distributed along the joint chain.

| | |
|---|---|
| Total | Applies an additional amount of twist to the end joint in the chain. Twist is measured in degrees.<br>Twist values for the remaining joints are interpolated from 0 at the start joint. The type of interpolation used is determined by the Twist Type attribute. |
| Start/End | Applies an additional amount of twist to the start and end joints in the chain. Twist is measured in degrees.<br>Twist values for the remaining joints are interpolated using the interpolation type specified by the Twist Type attribute. |

Ramp — Connects a texture to the Twist Ramp attribute to provide explicit twist values for the entire chain. Twist is measured in degrees.

Each joint is assigned a UV texture coordinate of (0,j/N), where *j* is the joint index in the chain and *N* is the number of joints in the chain. The alpha value returned by the texture is multiplied by the value of the Twist Ramp Multiplier attribute to produce the twist for each joint.

Start/End Twist

Specifies the twist values for the start and end joints of the chain. These fields are available only when Twist Value Type is set to Start/End.

Twist Ramp

Specifies twist along the chain by connecting to a texture. This field is available only when Twist Value Type is set to Texture.

Twist Ramp Multiplier

Since textures generally return values in the [0, 1] range, this multiplier ensures that the twist values are mapped using an appropriate range. This field is available only when Twist Value Type is set to Texture.

**Note** The Twist Value attributes work well with the Roll and Root Twist Mode attributes, and do not interfere with or supersede their functionality.

## effector

For information on the attributes for the effector*n* node, see the "General skeleton and IK node attributes" on page 122 and Help > Node and Attribute Reference.

## bindPose

Appears when a smooth skin or rigid skin is bound to the selected joint. For information on the attributes for the bindPose*n* node, see *Node Behavior* and *Set a node's update state* in the Basics guide.

## skinCluster

Appears when a smooth skin is bound to the selected joint. See ”skinCluster” on page 225.

## jointCluster

Appears when a rigid skin is bound to the selected joint. See ”jointCluster” on page 226.

## IK solver nodes

For more information, see Help > Node and Attribute Reference.

### Related topics

❖ ”IK solvers” on page 47
❖ ”Use single chain IK” on page 72
❖ ”Use rotate plane IK” on page 73
❖ ”Use multi-chain IK” on page 74
❖ ”Use spline IK” on page 75
❖ ”Use 2 bone IK” on page 83

### IK Solver Attributes

These attributes are common to all IK solvers.

#### Max Iterations

Specifies the maximum number of iterations the IK solver will take in calculating how the end effector reaches the IK handle. If the Tolerance cannot be meet after Max Iterations, the IK solver will stop. Default is 2147483647. A large value such as the default value means that the IK solver will typically stop when the Tolerance is met.

#### Tolerance

Specifies the precision sought by the IK solver in calculating how the end effector reaches the IK handle. Once the IK solver meets the Tolerance, the IK solver stops calculating. Default is 0.000.

### ikSystem

This node only appears when an IK solver node is selected.

The IK system node organizes all the default IK solvers in Maya. The IK system controls whether all the IK handles using IK solvers in the system snap to their end effectors, whether the IK solvers are active, and the order in which Maya evaluates the solvers.

For more information, see Help > Node and Attribute Reference.

ikSystem

The ikSystem window lists the IK solvers in the IK system. By default, the available IK solvers are the ikRPsolver (for IK rotate plane handles), the ikSCsolver (for IK single chain handles), and the ikSplineSolver (for IK spline handles). The order in which the IK solvers are listed informs you of the order in which Maya evaluates the solvers.

Global Snap

Specifies whether all the IK handles using any of the IK system's solvers will snap back to their end joint's positions. Turning Global Snap off has the effect turning off each IK handle's Snap Enable attribute. Default is on.

Global Solve

Specifies whether all the IK handles using any of the IK system's solvers are active. If off, you can only use forward kinematics (FK) posing to pose the joint chains controlled by the IK handles. Default is on.

# Tips

## Tips for working with spline IK handles

- To ensure that the joint chain moves smoothly when you animate the spline IK curve, the joints in your joint chain should be close to each other (short bones).
- Create a simple curve with no sharp bends to help make the joint chain move smoothly when you animate the curve. Use a small number of CVs.
- When you add an IK spline handle to the skeleton of most creatures—including fish and snakes moving along a motion path—parent each IK spline start joint to a transform node or parent joint that's not controlled by an IK spline handle. This makes the joint chain move with the transformations of the parent while avoiding unexpected joint flipping. See "Prevent flipping in the spline IK chain" on page 82.
- If you're working on a character with a root joint that rotates very little, for example, a swaying tree, you do not need to parent the start joint to a transform node or joint. The start joint can serve as the character's root joint.

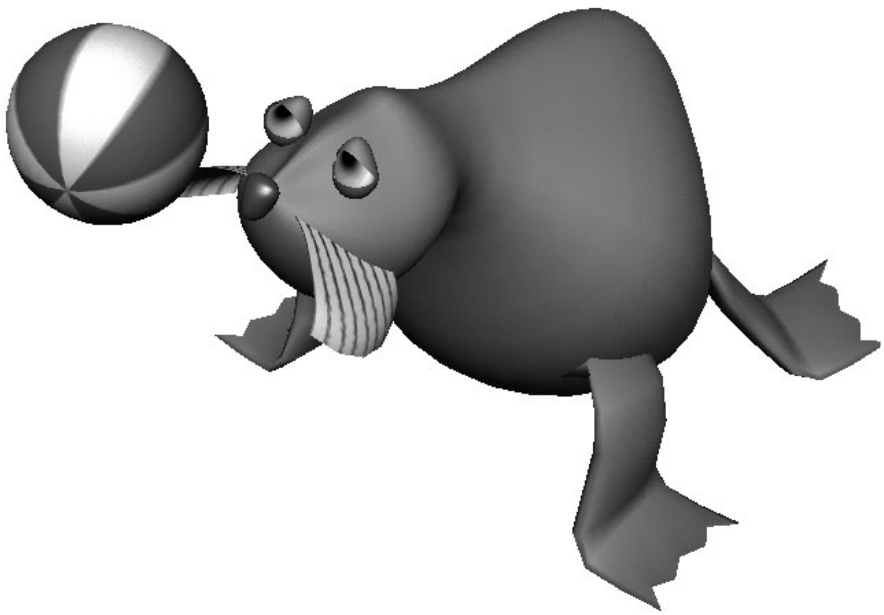
- For a character such as a fish or snake moving along a motion path, if you create a handle that starts at a skeleton's root, turn on Auto Create Root Axis when you create the IK spline handle. This prevents unexpected joint flipping as you animate the automatically created parent transform node along a motion path. Also turn off Auto Parent Curve.
- If you create a handle that starts at a joint other than the skeleton's root, turn on Auto Parent Curve and turn off Auto Create Root Axis so the handle's curve and start joint move with the transformations of the parent joint.

  When you manipulate a tail or neck parented to a spine, avoid moving the first CV of the curve for the tail or neck. Move the second CV minimally, preferably only along an imaginary line extending straight out from the end of the spine. Manipulate the other CVs freely. This technique ensures that the skin flows naturally where the spine meets the tail or neck.
- Do not parent the curve to the start joint. This creates a dependency graph loop that causes the start joint to chase the curve as the curve moves. To detect such loops, use the MEL command *cycleCheck -all.* For more information, see Help > MEL Command Reference and the MEL guide.
- Do not parent the curve to a transform node that uses that same curve as a motion path. In other words, don't turn on Auto Create Root Axis and Auto Parent Curve if you want to put the transform node on that curve. This creates a dependency graph loop.
- If you change an IK spline curve to a soft body, you can add dynamic forces to change the curve's motion. For example, you can connect turbulence to the curve to create random, erratic motion. For more information, see the Dynamics guide.

# 3 Skinning

## About

Skinning is setting up a model's deformable objects so that they can be deformed by skeletons.



**Understanding skinning**

### Understanding skinning

Skinning is the process of setting up a character's model so that it can be deformed by a skeleton. For more information on skeletons, see "Skeletons" on page 33. You skin a model by binding a skeleton to the model. You can bind a model to a skeleton by a variety of skinning methods, including smooth skinning and rigid skinning. Smooth skinning and rigid skinning are direct skinning methods. You can also use indirect skinning methods, which combine the use of lattice or wrap deformers with either smooth or rigid skinning.

## Deformable objects and skin objects

During skinning, you bind a model's deformable objects to a skeleton. After skinning, the model is called the character's skin, and the deformable objects are called skin objects.

A deformable object is any object whose structure is defined by NURBS control vertices (CVs), polygonal vertices, or lattice points. NURBS curves, NURBS surfaces, polygonal surfaces (meshes), and lattice deformers are all deformable objects. A character's model can consist of one deformable object (for example, a large polygonal surface) or of groups of deformable objects (for example, groups of NURBS surfaces).

Typically, a character's model consists of deformable objects (typically NURBS surfaces, polygonal surfaces, or both) organized in hierarchical groups. The organization reflects the structure of the character's appearance, and should be based on how the character will be animated. For example, a NURBS model could consist of groups of NURBS surfaces that make up the character's feet, legs, torso, arms, hands, neck, and head.

## Direct skinning methods

The direct skinning methods include smooth and rigid skinning.

### Smooth skinning

Smooth skinning provides smooth, articulated deformation effects by enabling several joints to influence the same deformable object points (CVS, vertices, or lattice points). For more information, see "Smooth skinning" on page 144.

### Rigid skinning

Rigid skinning provides articulated deformation effects by enabling joints to influence sets of deformable object points. For more information, see "Rigid Skinning" on page 160.

## Indirect skinning methods

Indirect skinning methods include lattice skinning and wrap skinning.

### Lattice skinning

In lattice skinning, you skin the influence lattices of lattice deformers. These influence lattices in turn influence other deformable objects (for example, NURBS surfaces or polygonal meshes). An advantage of lattice skinning is that you can easily make adjustments to the deformation by tweaking influence lattice points.

For more information about lattice deformers, see ”Lattice deformers” on page 243.

### Wrap skinning

In wrap skinning, you skin the wrap influence objects of wrap deformers. These wrap influence objects in turn influence other deformable objects. An advantage of wrap skinning is that you can skin low-res objects and use them as the character’s low-res model, and then later introduce the high-res model and deform its objects with the low-res objects.

For more information about wrap deformers, see ”Wrap deformers” on page 279.

## Bind pose

When you bind skin, Maya creates a bind pose node (default name: bindPose*n*) for each skeleton. For each skeleton, the bind pose node keeps track of the joints’ transformation attributes (translation, rotation, and scale) when skinning takes place. The bind pose node also keeps track of the transformation attributes of any influence objects. The bind pose node facilitate putting the skeleton back into the bind pose at any time after binding skin.

The use of constraints, expressions, or IK handles with keys set can restrict going to the bind pose. If you are using constraints, expressions, or IK handles and you want to go back to the bind pose, you will need to disable the nodes that carry out these features. They can restrict going to bind pose because they can lock the attributes they affect, preventing them from being set to the bind pose values.

## Double transformation effects

A double transformation effect is where skin object points are subjected to the action of a joint more than once, resulting in extreme, undesirable shape changes. Double transformation effects can occur if the skin object points are also being affected by a deformer that is in turn affected by the joint’s actions. For example, if after skinning you create a cluster deformer to further control certain skin object points and then parent the cluster deformer handle to the joint that also affects the skin object points, when you move the joint the points will be affected twice over by the joint’s action. One way to remedy this is to organize the affected points into a set that is only affected by the cluster deformer, which remains parented to the joint.

**Tip** If you bind an object as skin, and bind a shape that is in the construction history of that object, you will get double transformations.

For example, revolve a curve and bind the curve as well as the revolved surface. You will get double transformations since the surface is deformed by the skeleton, plus the revolve is computing a new revolve based on the curve's deformation.

To counter this, delete or detach the construction history on the shape before binding.

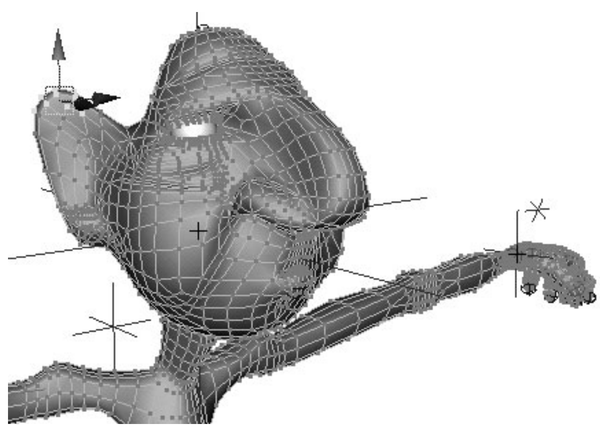
### Editing skin point set memberships

## Editing skin point set memberships

Skinning organizes deformable object points (CVs, vertices, or lattice points) into skin point sets. You can edit these sets in the same ways that you can edit deformer sets. For more information, see "With Maya's deformers, you can change the shape of objects." on page 231 and "Editing deformer set membership" in Chapter 4.

### Point tweaking skinned objects

## Point tweaking skinned objects



Point tweaking skin objects is moving or setting keys on the individual skin points of a skinned object. When you tweak the points of a skinned object, Maya automatically prevents the unexpected effects that can occur when you manipulate the object. Maya does so by applying the tweaks to the object before applying any deformations to the object.

When you bind skin, Maya creates tweak nodes as well as nodes for the skinned objects. In the dependency graph, Maya places the tweak nodes as inputs for the smooth skin objects so that any point tweaking is carried out before the evaluation of the smooth skin nodes or rigid skin nodes. This placement means that, by default, a skinned object's deformation order includes point tweaking first.

If you prefer, you can change the deformation order so that point tweaking does not occur first.

Also, if you do some point tweaking and then want to check how the skinned object deforms without the tweaking, you can disable the tweak node.

| | |
|---|---|
| Note | You can do point tweaking on objects after skinning, but you should avoid changing the number of the object's points (for example, CVs, vertices, or lattice points). Changing the number of points can lead to unexpected deformation effects. |

## Editing node behavior to improve performance

You don't need to know about node behavior in order to do skinning effectively. If you are new to skinning, you can skip this section. However, familiarity with node behavior can provide you with more control over the performance of skinned objects.

For each object in your scene, if there has been any change to its node or any of the nodes in its history (its input or output nodes), Maya will evaluate the nodes and update the display. A skinned object has more nodes in its history than an object unaffected by skinning or deformers. If you have many skinned objects in your scene, you could improve the display performance by editing the node behavior attributes of the skinned nodes.

### Workflow summary

## Workflow summary

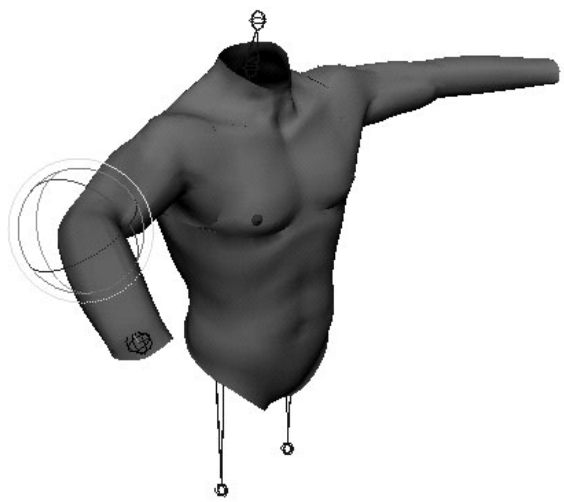
Once you've created a model and skeleton for your character, you're ready to skin the model so that the skeleton's actions can deform it. For skinning, you can either use smooth skinning or rigid skinning. For more information on smooth skinning, see "Smooth skinning" on page 144. For more information on rigid skinning, see "Rigid Skinning" on page 160. You can also skin the influence lattices of lattice deformers or the wrap influence objects of wrap deformers either by smooth or rigid skinning.

Skinning can be an iterative process in which you might have to edit and refine the skeleton and the model's deformable objects to get the right skin deformation effects.

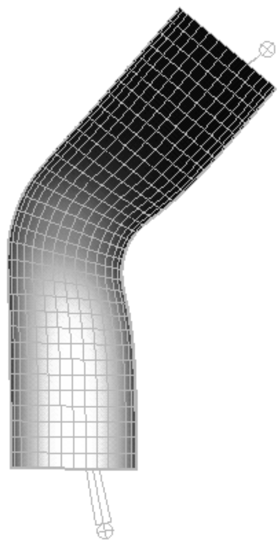
### Smooth skinning

## Smooth skinning

Smooth skinning provides smooth, articulated deformation effects by enabling several joints to influence the same deformable object points. If you'd like to explore some examples now, see "Examples" on page 177.



# Understanding smooth skinning



With smooth skinning, more than one joint can influence each CV to provide smooth bending effects. You do not need to use flexors or lattice deformers.

Smooth skinning makes smooth, articulated deformation effects available immediately after you bind skin. The smoothing effects around joints are automatically set up when you bind skin. Maya provides smooth deformation effects by allowing several nearby joints to have varying influences on the same skin points (NURBS CVs, polygonal vertices, or lattice points). By default, their influence varies with distance, but you can edit or paint the skin point weighting on a joint-by-joint basis.

Unlike rigid skinning, with smooth skinning you don't have to use deformers, flexors, or edit skin point set memberships to get smooth deformation effects. The smoothing effects around joints are automatically set up when you bind. The effect of each joint on a smooth skin point depends on the joint's proximity to the point.

## Smooth skin objects and points

During smooth skinning, you bind a model's deformable objects to a skeleton. After smooth skinning, the deformable objects are called smooth skin objects (or skin objects, or skin). The points (NURBS CVs, polygonal vertices, or lattice points) of the deformable objects are then referred to as smooth skin points, or skin points.

The skin points can automatically avoid the improper influence of joints that are in close proximity but are far in terms of the skeleton's hierarchy. For example, you can avoid having to worry about hand skin points being influenced by a nearby thigh joint.

## Smooth skin point weights

During smooth skinning, for each smooth skin point (for example, each CV of each NURBS surface), Maya assigns a smooth skin point weight for each joint that controls the influence of that joint on each point.

If you want to change the results of smooth skinning to create unique skeletal deformation effects, you can edit or paint the weights of smooth skinning at the point level (the CV, vertex, or lattice point level). Additionally, to add further deformation effects to smooth skin, you can use Maya's deformers and smooth skin influence objects.

Joints closer to a smooth skin point will have a greater influence than joints far from the skin point. The joint closest to a smooth skin point will have the greatest influence.

Which joints have the next greatest influence can depend on whether you want Maya to consider the skeleton's hierarchy during binding or to ignore the skeleton's hierarchy during binding.

### Weighting based on skeleton hierarchy

If you tell Maya to consider the skeleton's hierarchy, the joint that will have the next greatest influence will be a relative (parent or child) of the closest joint. For example, if your character's arms are hanging down so that the forearm bones are near the hip bones, you can make sure that the skin points for the arms do not come under the influence of the hip bones. This is because the hip bones are not near the forearm bones in the skeleton's hierarchy even though the distance between them is small.

When you bind skin, you tell Maya to consider the skeleton's hierarchy by setting the Bind Method option to Closest Joint. See "Skin > Bind Skin > Smooth Bind" on page 211.

### Weighting based only on joint proximity

If you tell Maya to ignore the skeleton's hierarchy (set Bind Method to Closest Distance), the next joint that will have the next greatest influence on a smooth skin point's weight is always the next closest joint to the point. Depending on the structure of the skeleton and the placement of the model, this joint could be much higher or lower in the skeleton's hierarchy than the closest joint. For example, if your character's arms are hanging down so that the forearm bones are near the hip bones, the hip bones could have the second greatest influence over the skin points for the arms. This could lead an inappropriate influence on the weights of the arm's skin points.

When you bind skin, you tell Maya to ignore the skeleton's hierarchy by setting the Bind Method option to Closest Distance. See "Skin > Bind Skin > Smooth Bind" on page 211.

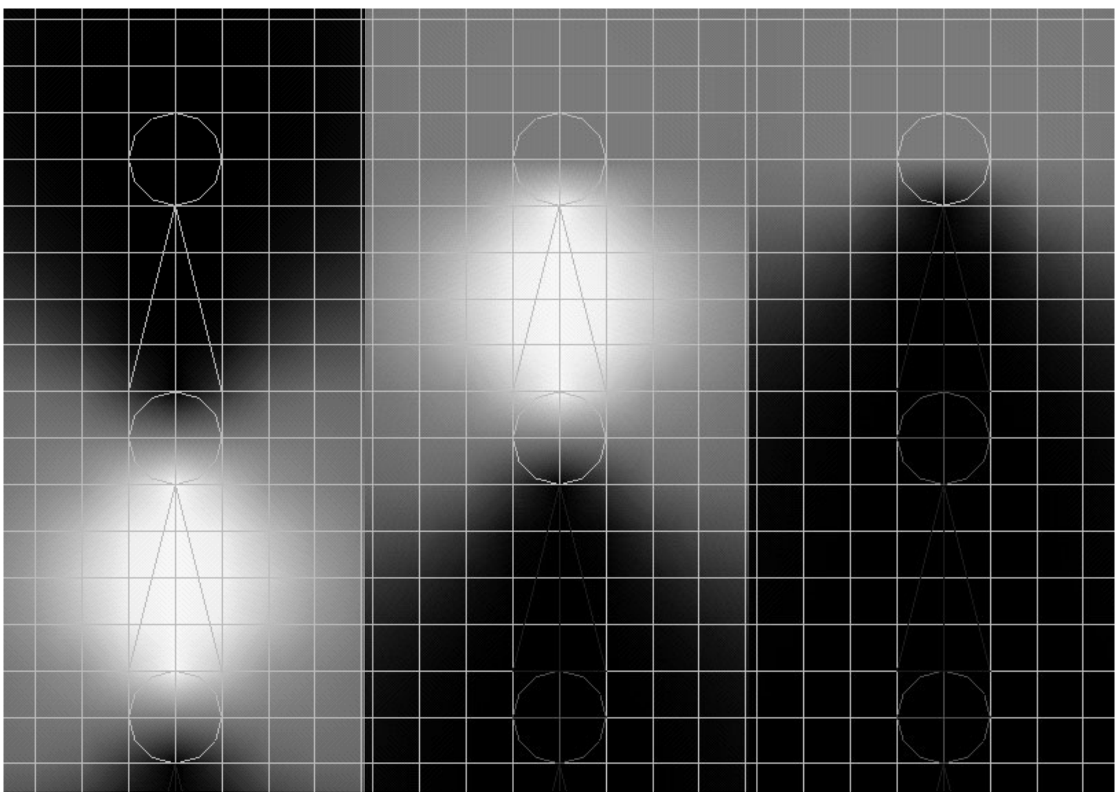
## Weighting can be influenced by varying number of joints

You can control how many of the skeleton's joints can influence a smooth skin point's weight. A typical value for a character would be a maximum of four or five joints that can influence a given smooth skin point.

When you bind skin, you tell Maya how many joints can influence a smooth skin point by specifying the Max Influences option. See "Skin > Bind Skin > Smooth Bind" on page 211.

## Weighting varies based on joint distance

The influence of each joint on a smooth skin point's weight varies with the distance between the skin point and the joint.



Note that if the joint has a bone, the influence of the joint extends along the entire bone, from the center of the joint to the end of the bone. The joint's influence can extend to all the points near the entire length of the bone. If the joint is an end joint (has no bone), then the joint's influence just extends forward from the center of the joint.

When you bind skin, you tell Maya how the weighting varies based on joint distance by specifying the Dropoff Rate option. See "Skin > Bind Skin > Smooth Bind" on page 211.

## Smooth skin point sets

A set of smooth skin points is created for each deformable object. The set contains all the points (NURBS CVs, polygonal vertices, or lattice points) that can be influenced by the skeleton.

For more information on sets and partitions, refer to the *Basics* guide.

## Smooth skin influence objects

With smooth skinning, you can use NURBS or polygonal objects as smooth skin influence objects to further shape and control the deformation of smooth skinned objects. Such influence objects provide deformation effects in a manner similar to the wrap deformer's wrap influence objects. You can use the influence objects to restrict deformation (for example, undesirable shoulder deformations) as well as to create deformations (for example, bulging muscles).

For more information on using smooth skin influence objects, see "Using smooth skin influence objects" on page 156.

## Checking the binding

Exercise the skeleton to check the smooth skin deformation effects. Rotate the skeleton's joints to view the smooth skin's behavior. As you exercise the skeleton, at times you might want to go back to the bind pose. For more information on going to the bind pose, see "Going to the bind pose" on page 149.

As you check the binding, you may find that you want to adjust the smooth skin's behavior.

## Editing smooth skin

### Adjusting smooth skin behavior

If you don't like the smooth skin's behavior, you can detach the skin, edit the skeleton or the deformable objects, set new binding options, and then bind again. Skinning can be an iterative process of checking the binding, detaching, editing the skeleton, and then binding again. For more information on detaching skin, see "Detach smooth skin" on page 191.

However, if you just want to edit the smooth bind options (for example, Max Influences and Dropoff Rate) without detaching and binding again, you can do so. For more information, see ”Editing maximum influences” on page 150, and ”Edit joint smooth skin attributes” on page 186.

To change the smooth skinning deformation effects, you can edit the skin point weights with the Component Editor or the Paint Skin Weights Tool. As you check the binding, you can use the Paint Skin Weights Tool to view joint influences and change them by painting. This tool provides an intuitive way to modify the deformation effects. For more information, see ”Painting smooth skin point weights” on page 151.

Note that when you edit joint smooth skin attributes and change Dropoff attributes, you then have to have Maya recalculate the affected skin point weights. In turn, this can alter any other changes you might have made to the skin point weights. Consequently, it’s a good practice to edit the Dropoff attributes first, and then proceed to editing and painting the skin point weights.

You can also control smooth skinning deformation effects with smooth skin influence objects. A smooth skin influence object can be any NURBS surface, NURBS curve, or polygonal surface (mesh). For more information, see ”Using smooth skin influence objects” on page 156.

## Going to the bind pose

The bind pose is the pose that the skeleton is in when you bind skin. When you pose a character’s skeleton after skinning, the skeleton’s actions cause deformations to the skin. The only pose that does not cause deformations to the skin is the bind pose.

You must return to the bind pose if you decide to bind additional objects or add additional influence objects.

Note that if you bound smooth skin to selected joints only, going to bind pose will nevertheless return all the joints the skeleton to the bind pose. Also, the skeleton will go to the bind pose even if it is parented to group nodes. The group nodes will not prevent going to bind pose.

## Overcoming problems with reaching bind pose

The skeleton will not be able to go to the bind pose right away if the attributes of any of its joints are locked. Typically, joint attributes can be locked by constraints, expressions, IK spline handles, or any IK handles with keys set. These features can drive the values of certain joint attributes, locking them up for exclusive use. That they do lock certain attributes is desirable because it provides for the reliable effects of these features. However, if you want to go to the bind pose, you must first disable the nodes that are locking the attributes. A quick way to do this is to disable all of the nodes by selecting Modify > Enable Nodes > Disable All. Next, select Skin > Go to Bind Pose, and then enable all nodes again by selecting Modify > Enable Nodes > Enable All.

### Changing the bind pose

To change the bind pose, detach the smooth skin, adjust the skeleton and deformable objects as desired, and then bind skin again. You can also reset the bind pose from the Command Line. See ”Reset bind pose from Command Line” on page 170.

## Editing maximum influences

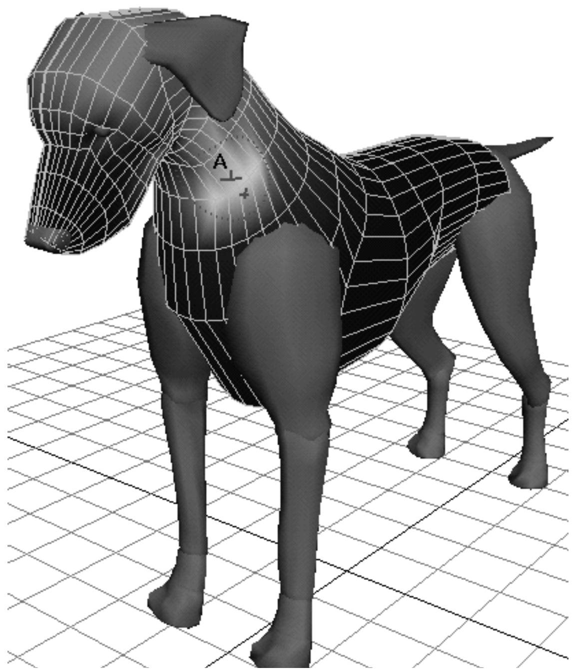
You can change the number of joints or influence objects that can influence a skin object’s points. Note that before you bind skin, the Maximum Influences smooth bind option specifies the maximum number of joints or influence objects.

## Editing skin point weights

With the Component Editor, you can directly modify the values of individual skin point weights. Note that you can also paint these weights with the Paint Skin Weights Tool. If you want to set certain skin point weights to particular values, use the Component Editor. However, if you want to shape the deformation directly and are not concerned with specific values, use the Paint Skin Weights Tool (see ”Painting smooth skin point weights” on page 151).

While you are editing skin point weights, you can reset the weights to their initial values at any time (see ”Reset skin point weights to default weights” on page 188). You can also prevent indirect changes to skin point weights, which can happen if Maya is normalizing the weights (see ”Controlling smooth skin weight normalization” on page 154).

# Painting smooth skin point weights



You can paint skin point weights with the Paint Skin Weights Tool. The Paint Skin Weights Tool provides an intuitive way to change the deformation effects of smooth skinning. If you want to set individual skin point weights to specific values, you can use the Component Editor (see ”Editing skin point weights” on page 150).

While you are painting skin point weights, you can reset the weights to their initial values at any time (see ”Reset skin point weights to default weights” on page 188).

To explore an example that includes painting skin point weights, see ”Skinning a cylinder by smooth skinning” on page 177.

Note that painting smooth skin point weights uses a different painting tool than the tool for painting rigid skin point weights.

| | |
|---|---|
| **Warning** | If you want to paint smooth skin weights on a polygon mesh, then the mesh’s UV maps must be clean and free of overlapping UVs. Otherwise, undesired results will occur. |

## Painting skin weights on masked vertices

You can create a mask on the skin that is unaffected by any weight painting you do. When you apply brush strokes over the mask, the vertices on the masked area retain their weight, regardless of the paint weights operation.

Before creating the mask you must first create the skin. For details on masking surfaces, see *"Restrict an area for painting"* in the Paint Effects, Artisan, and 3D Paint guide.

## Mirroring smooth skin weights

You can mirror smooth skin weights, either from one smooth skin object to another, or within the same smooth skin object. You can also mirror skin weights on selected components of the skin.

Mirroring smooth skin weights greatly speeds up the process of editing and fine-tuning skin deformation effects. For example, you could perfect the smooth skin weighting for a character's right shoulder area and then simply mirror the weighting to the character's left shoulder.

Maya mirrors weights across planes defined by Maya's global workspace axis. For the mirroring to work properly, the skin objects (or character) should be centered on the global axis, or at least aligned along the axes you want to mirror about.

## Copying smooth skin weights

You can copy smooth skin weights from one smooth skin object to another, or from one group of smooth skin objects to another.You can also copy skin weights on selected components of the skin.

For example, suppose you have created a team of very similar characters for a football game, and you're at that point in the character setup process where you have just skinned them all and are about to paint the smooth skin weights to perfect the deformation effects. You first paint the weights of the quarterback. Next you have to paint the weights of all the other players. Instead of having to paint the players individually, you can copy what you did for the quarterback to each of the other players. You might then want to fine-tune the weights of the various other players, but by copying the weights you have still saved yourself a lot of time. You can focus on the creative challenges unique to the case of a particular character rather than on doing the same type of painting over and over again.

For best results, the skeleton of the character you are copying from and the skeleton you are copying to should have the same structure. If the skeletons are similar, Maya will still try to copy the weights. However, if the skeletons are radically different, Maya may not be able to copy the weights.

Also, for best results, the skeletons of each character should be in the same pose during copying. If the orientation of the joints are not similar, the copying can lack some precision, which means you may have to do some touch up painting to the results.

If the skin objects have different numbers of CVs, or if the ordering of the CVs is different, the copying will intelligently take into account the differences and provide the same type of weighting. This is very useful if you want to apply the smooth skin weighting from a high-res character to a low-res version of the character.

You can copy smooth skin weights between skin objects of different types: for example, you can copy from a subdivision surface to a NURBS surface or a polygonal surface.

## Holding smooth skin weights

When you are changing (editing or painting) the weights of smooth skin objects, changing the weights of one object can affect the weights of other objects. This is because Maya must consider the weights of all skin objects being influenced by a particular influence object as being relative to one another. Maya does this by requiring that all the weights add up to one. When you change certain weights, Maya can automatically change various other weights so that the total of all the weights continues to be one. This allows Maya to know the relative influences of the weights.

The process of scaling some numbers so that they all add up to one is called “normalization.” By constantly normalizing weight values, Maya keeps track of their relative influences.

When changing (editing or painting) weights of smooth skin objects, it’s sometimes desirable to specify that the weights of certain objects won’t change. If you’ve perfected the weighting of a particular object, you might want to make sure that its weights are not going to undergo any normalization changes. Maya lets you “hold” the weights of particular smooth skin objects so that their values don’t change when you are editing other smooth skin objects.

Note that if you hold the weights of many objects at the same time, Maya might not be able to normalize the rest of the weights properly, and you could get an error message. In general, you should not hold the weights of many objects at the same time. Typically, you would want to hold the weights of only one or two objects. However, if you don’t want to stop holding the weights of any of the objects, you can turn off weight normalization directly from Maya’s interface (see ”Controlling smooth skin weight normalization” on page 154).

## Controlling smooth skin weight normalization

When you are changing (editing or painting) the weights of smooth skin objects, changing the weights of one object can affect the weights of other objects. This is because Maya considers the weights of all skin objects being influenced by a particular influence object as being relative to one another. Maya does this by requiring that all the weights add up to one. When you change certain weights, Maya can automatically change various other weights so that the total of all the weights continues to be one. This allows Maya to know the relative influences of the weights.

The process of scaling numbers so that they all add up to one is called “normalization.” By constantly normalizing weight values, Maya keeps track of their relative influences.

Maya normalizes smooth skin weights by default, but you can control whether a smooth skin object’s weights are normalized. You can disable weight normalization, and also enable it again. Also, if you’ve changed weight values with normalization disabled, and then decide to normalize them, you can do so.

## Batch export and import of smooth skin weight maps

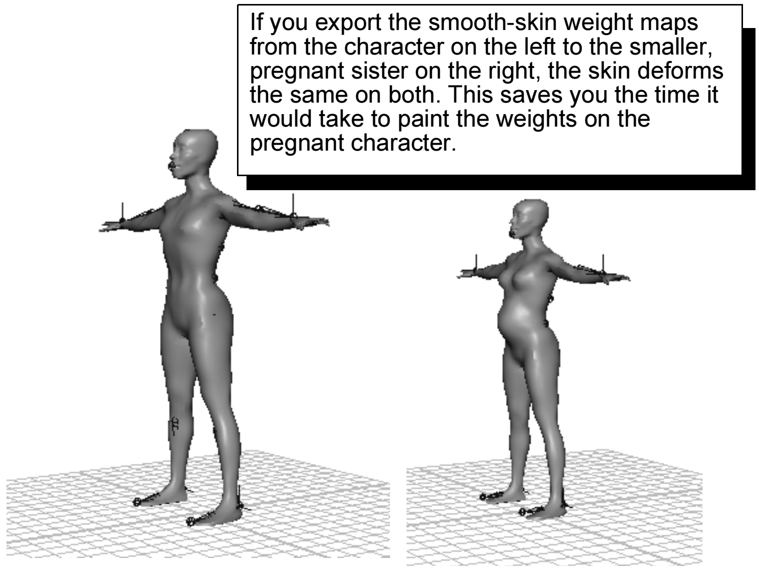
When you smooth skin a surface, Maya creates one weight map per joint. Maya has a menu item that exports all the weight maps at once.

Warning
- You cannot export or import skin wight maps to lattices.
- When importing painted smooth skin weight maps from a polygon mesh, the mesh’s UV maps must be free of any overlapping UVs. Otherwise, undesired results will occur.

### Importing to a character in another scene

After you export the weight maps, you can import them to a smooth-skinned surface in another scene so that its skin deforms the same as the first surface. The surface in the second scene must have the same UV orientation as the original surface, but it can differ from the original as follows:

- It can be scaled differently but must be proportioned similarly in regions of significant deformation, typically around joints.
- It can have a different number of spans and sections.
- It can have a different world space position.

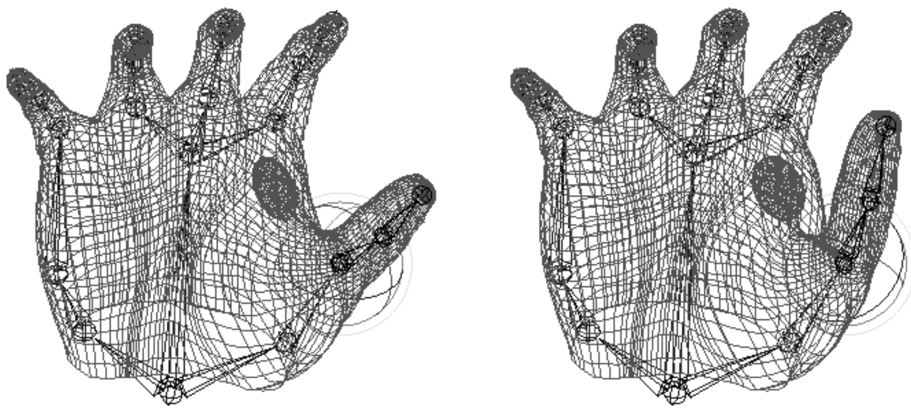



## Importing back to the original character

You can also import the maps back to the original surface. Examples of how this is useful follow:

- If you are roughly satisfied with the skin weights for a surface but want to experiment with different weights to enhance the look, you can export the maps to have a backup of the satisfactory version.
- If you apply an influence object to the surface, Maya alters the weights of the smooth skinning in the region of the influence object, sometimes with undesirable results. To avoid this situation, you can export the maps, add the influence object, import the maps to the surface again, then paint weights near the influence object. This avoids the unintended automatic weight alterations.

## Using smooth skin influence objects



Influence objects can deform smooth skin objects by influencing the position (translation) attributes of their skin points. If you'd like to explore some examples now, see "Hand muscle bulge with influence object" on page 157 and "Using influence objects to prevent unwanted deformation" on page 158.

You can add any object as an influence object so that the object's transform attributes affect the position attributes of skin points. For example, you could use a locator as an influence object so that when you move the locator you move skin points, creating a deformation effect.

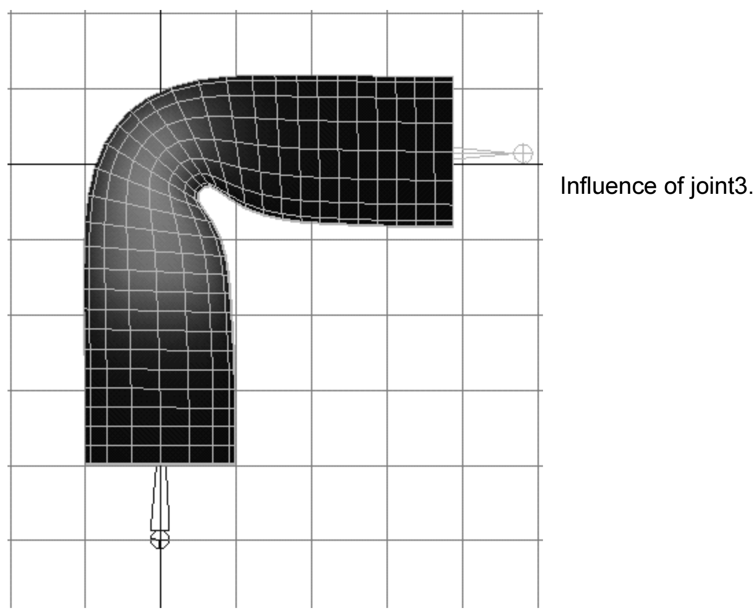
If the influence object is a NURBS or polygonal surface, the skin points can be influenced by the shape of the surface. The surface can push or pull skin points that are in its vicinity, creating deformation effects that reflect the surface's shape. When placed near the surface of the skin, these polygonal influence objects can be very useful for creating deformations that indicate the effects of moving veins, bones, tendons, or muscles. You can also create interesting effects with NURBS curves.

Influence objects influence smooth skin objects in the same manner that joints can influence smooth skin objects. The Dropoff Rate, which is set as a smooth bind option, applies to the influence of influence objects as well as to the influence of joints. You can change the Dropoff Rate for each influence object.

Whether a smooth skin object's skin points can be influenced by the components of the surface (for example, the individual polygonal faces) of an influence object or the overall shape of the object depends on the skin cluster node's Use Components attribute. If Use Components is set to on, skin points can be influenced by components (for example, individual polygonal faces). If Use Components is set to off (the default), the skin points are only influenced by the overall shape of the influence object.

| | |
|---|---|
| Warning | You should avoid changing the number of an influence object's points (for example, CVs, vertices, or lattice points) after you add it as a smooth skin influence object. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the object's topology before you begin using deformers. You might want to save a copy of the object in case you want to do further modeling later. |

## Hand muscle bulge with influence object

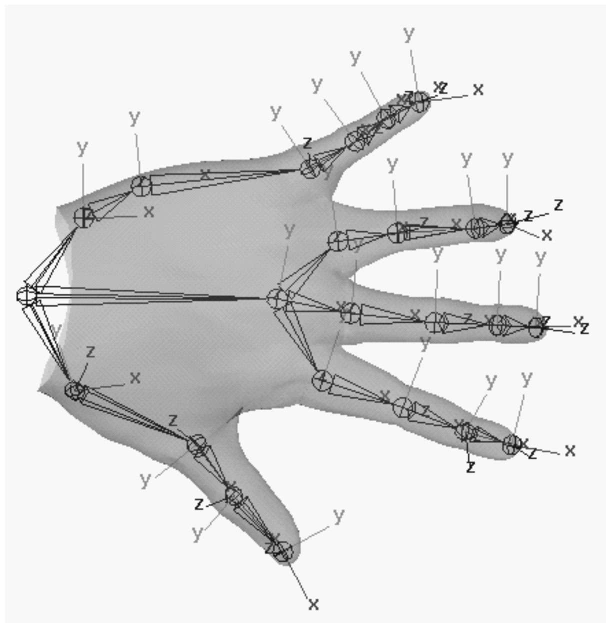



Setting up hands for animation is one of the most demanding aspects of character setup. With smooth skinning, you can achieve more subtle effects by using influence objects whose actions are driven by nearby joints.

When you move your thumb towards your index finger, a muscle along the side of the upper part of your hand (*m. interossesus dorsalis*) tends to bulge out, indicating the tension in your hand. In rigid skinning, you could use a flexor to provide bulge effects, although positioning a flexor right at where this muscle bulges could be tricky. With smooth skinning, you can use an influence object the provide the deformation.

## Creating a hand's skeleton and smooth skin

Suppose you have created a model for a hand. The hand consists of NURBS surfaces, with surfaces for the fingers, thumb, and palm area. Suppose you have also created a skeleton for the hand, and have just bound the NURBS surfaces to the skeleton by smooth skinning.



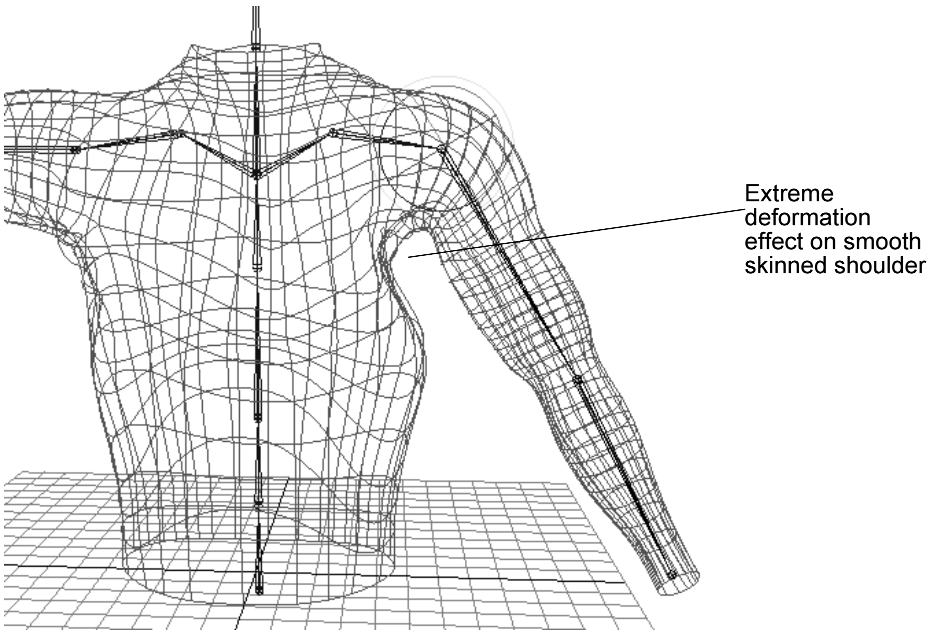
Hand with skeleton (local rotation axes displayed)

Now you could set up an influence object to create a hand muscle bulge by creating a polygonal sphere, setting the sphere as an influence object, and then linking the sphere's scaling to the movement of the skeleton's thumb.

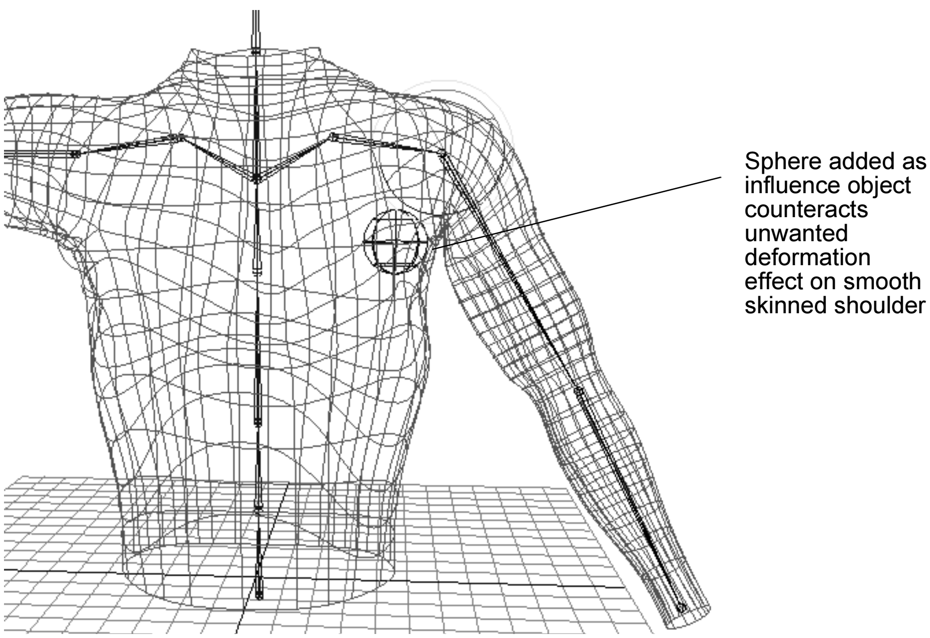
(For more information on smooth skin influence objects, see "Using smooth skin influence objects" on page 156.)

## Using influence objects to prevent unwanted deformation

An important use of influence objects is to prevent unwanted deformations from occurring. After smooth skinning, you may find that certain areas deform more than you would like. For example, deformations around a shoulder may be too extreme when you rotate the shoulder joint.




You can prevent such unwanted effects by adding an influence object whose influence counteracts the influence of other deformations. For example, you could add an influence object near the shoulder area that could counteract any extreme effects resulting from smooth skinning.



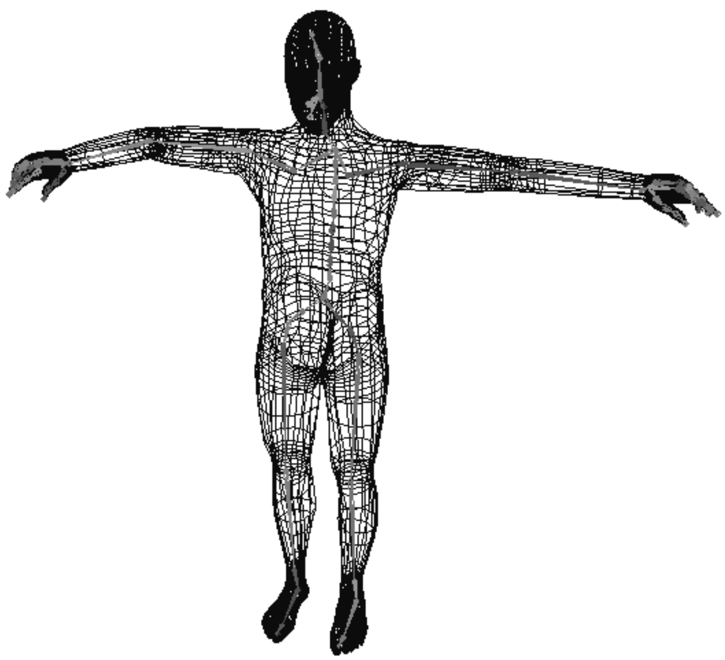
Note that you can use a NURBS or polygonal object of any shape. In this example, you might want to edit the sphere's shape for more precise control over the final deformation around the shoulder.

For more information on smooth skin influence objects, see "Using smooth skin influence objects" on page 156.

## Rigid Skinning

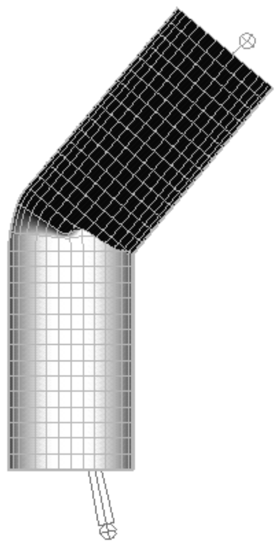
# Rigid Skinning

Rigid skinning provides articulated deformation effects by enabling joints to influence sets of deformable object points. If you'd like to explore an example now, see "Example" on page 205.



**Understanding rigid skinning**

# Understanding rigid skinning



With rigid skinning, only one joint can influence each CV. This can result in rigid bending effects unless you also use flexors or lattice deformers.

With rigid skinning, only one joint can influence each CV, vertex, or lattice point. This provides rigid deformation effects that you can smooth by using lattice deformers, cluster deformers, or flexors.

## Rigid skin objects and points

During rigid skinning, you bind one or more deformable objects to a skeleton. Once bound, the objects are rigid skin objects (or skin objects, or skin) whose position, orientation, and scale are controlled by the skeleton's joints. The points (CVs, vertices, or lattice points) of the deformable objects are then referred to as rigid skin points, or skin points.

Maya binds rigid skin objects to joints by means of joint cluster nodes. One way you can change the rigid skinning deformation effects is by editing joint cluster attributes (or channels).

## Rigid skin point weights

During rigid skinning, for each rigid skin point (for example, each CV of each NURBS surface), Maya assigns a rigid skin point weight that controls the influence of a joint on the point (for example, the CV). By default, each joint can influences the skin points of its nearest skin object equally, but you can edit the amount by which a joint can influence a skin point.

The main difference between rigid skinning and smooth skinning is that in rigid skinning only one joint can influence a particular skin point (CV, vertex, or lattice point), but in smooth skinning, many joints can influence the same skin point. Because smooth skinning allows many joints to influence the same skin point, you can immediately get smoother deformation effects right after binding skin.

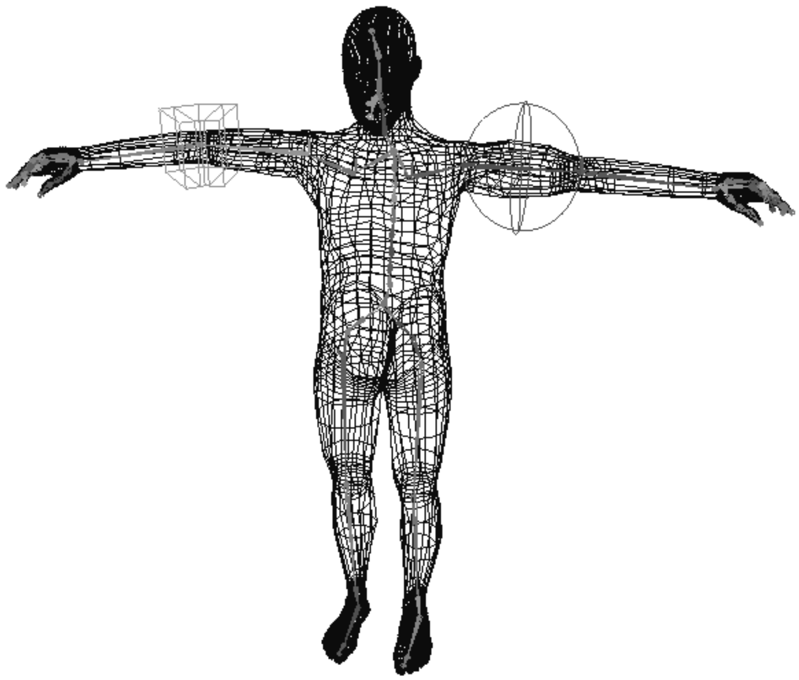
## Rigid skin point sets

During rigid skinning, for each joint, Maya creates a set of rigid skin points. The set contains all the points (NURBS CVs, polygonal vertices, or lattice points) that can be influenced by a particular joint. By default, the sets are organized into a partition, which means the sets can have no members in common.

You can organize points into a partition of sets before you bind skin, arranging for some of the points to become rigid skin points while other points can be influenced by, for example, a cluster or lattice deformer parented to a nearby joint. Using cluster or lattice deformers with rigid skinning can be a good way to get smooth deformation effects around areas such as shoulders, and provide an alternative to using flexors (see ”Flexors” on page 163). However, you must plan ahead, and carefully decide which points are going to be affected by what, or your character could suffer from double transformation effects. See ”Double transformation effects” on page 141.

For more information on sets and partitions, refer to the *Basics* guide.

# Flexors



Flexors are special deformers designed for use with rigid skinning. They provide various types of deformation effects that improve and enhance the effects provided by rigid skinning. Maya includes five types of flexors:

- Joint lattice flexors provide smoothing effects around joints. They are based on lattice deformers.
- Bone lattice flexors provide smoothing and bulging effects around bones. They are based on lattice deformers.
- Joint sculpt flexors provide rounded deformation effects around joints. They are based on sculpt deformers.
- Bone sculpt flexors provide rounded deformation effects around bones. They are based on sculpt deformers.
- Joint cluster flexors provided weighted deformation control around joints. They are based on cluster deformers.

For more information on creating flexors, see "Create all types of flexors" on page 196.

# Binding rigid skin

You can set the bind skin options before you bind skin, or immediately bind skin with the current options. After you bind skin, you can check the binding and adjust the skin's behavior.

# Checking the binding

Exercise the skeleton to check the rigid skin deformation effects. Rotate the skeleton's joints to view the rigid skin's behavior. As you exercise the skeleton, at times you might want to go back to the bind pose. For more information on going to the bind pose, see "Going to the bind pose" on page 165.

As you check the binding, you may find that you want to adjust the rigid skin's behavior.

# Adjusting rigid skin behavior

If you don't like the rigid skin's behavior, you can detach the skin, edit the skeleton or the deformable objects, set new binding options, and then bind again. Skinning can be an iterative process of checking the binding, detaching, editing the skeleton, and then binding again. For more information on detaching skin, see "Detach rigid skin" on page 199.

If you just want to move, rotate, or scale certain skin objects or joints without changing the existing rigid skin point sets and rigid skin point weights, you can do so by detaching and then reattaching the skeleton, or by detaching and then reattaching selected joints only. For more information, see "Detach and reattach to skeletons" on page 200, and "Detach and reattach to selected joints" on page 201.

To change the smooth skinning deformation effects, you can edit the rigid skin point weights with the Component Editor or the Paint Weights Tool, the same tool you can use to paint cluster deformer weights. As you check the binding, you can use the Paint Weights Tool to view the influence of each joint and change the weights by painting. This tool provides an intuitive way to modify deformation effects. For more information, see "To paint weights on a rigid bound skin" on page 193.

### Editing rigid skin

# Editing rigid skin

Editing rigid skin is described in the following sections.

## Going to the bind pose

The bind pose is the pose that the skeleton is in when you bind skin. When you pose a character's skeleton after skinning, the skeleton's actions cause deformations to the skin. The only pose that does not cause deformations to the skin is the bind pose.

You must return to the bind pose if you decide to bind additional objects or add additional influence objects.

Note that if you bound smooth skin to selected joints only, going to bind pose will nevertheless return all the joints the skeleton to the bind pose. Also, the skeleton will go to the bind pose even if it is parented to group nodes. The group nodes will not prevent going to bind pose.

## Overcoming problems with reaching the bind pose

### Locked attributes

The skeleton will not be able to go to the bind pose right away if the attributes of any of its joints are locked. Typically, joint attributes can be locked by constraints, expressions, IK spline handles, or any IK handles with keys set. These features can drive the values of certain joint attributes, locking them up for exclusive use. That they do lock certain attributes is desirable because it provides for the reliable effects of these features. However, if you want to go to the bind pose, you must first disable the nodes that are locking the attributes. A quick way to do this is to disable all of the nodes by selecting Modify > Enable Nodes > Disable All. Next, select Skin > Go to Bind Pose, and then enable all nodes again by selecting Modify > Enable Nodes > Enable All.

### Global and local bind pose

To reach its bind pose, the skeleton's root joint must reach the pose it had during binding, and all the other joints below the root joint must reach the poses they had during binding. The pose of the root joint is relative to the scene's world space, and the poses of the other joints are relative to the joints above them in the skeleton's hierarchy.

A skeleton reaches its global bind pose when the root joint reaches its bind pose. A skeleton reaches its local bind pose when the other joints reach their bind poses.

Depending on what you are doing to the joints (including constraints or expressions, for example), your skeleton might reach its local bind pose, but not its global bind pose. If you get an error message that says your skeleton could only reach its local bind pose, that means that all the joints reached their bind poses except the root joint. You then need only check the root joint for locked attributes or expressions that may be affecting it.

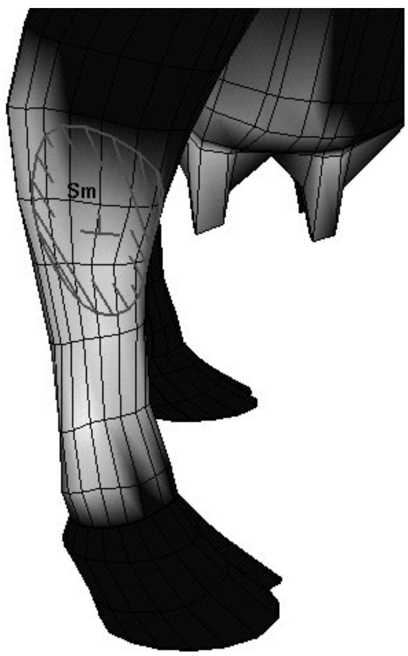
## Changing the bind pose

To change the bind pose, detach the rigid skin, adjust the skeleton and deformable objects as desired, and then bind skin again.

## Editing joint cluster channels

Channels are the keyable attributes displayed in the Channel Box. The Channel Box provides a convenient way to edit a joint cluster channels.

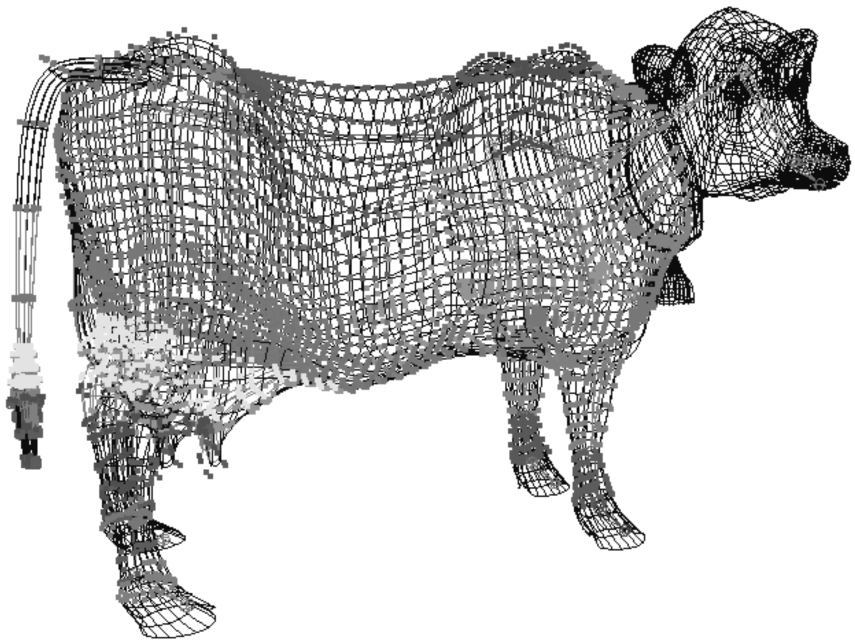
## Painting rigid skin point weights




You can paint rigid skin point weights with the Paint Cluster Weights Tool, the same tool you use to paint cluster deformer weights. For more information about the Paint Cluster Weights Tool, see "Deform > Paint Cluster Weights Tool" on page 397.

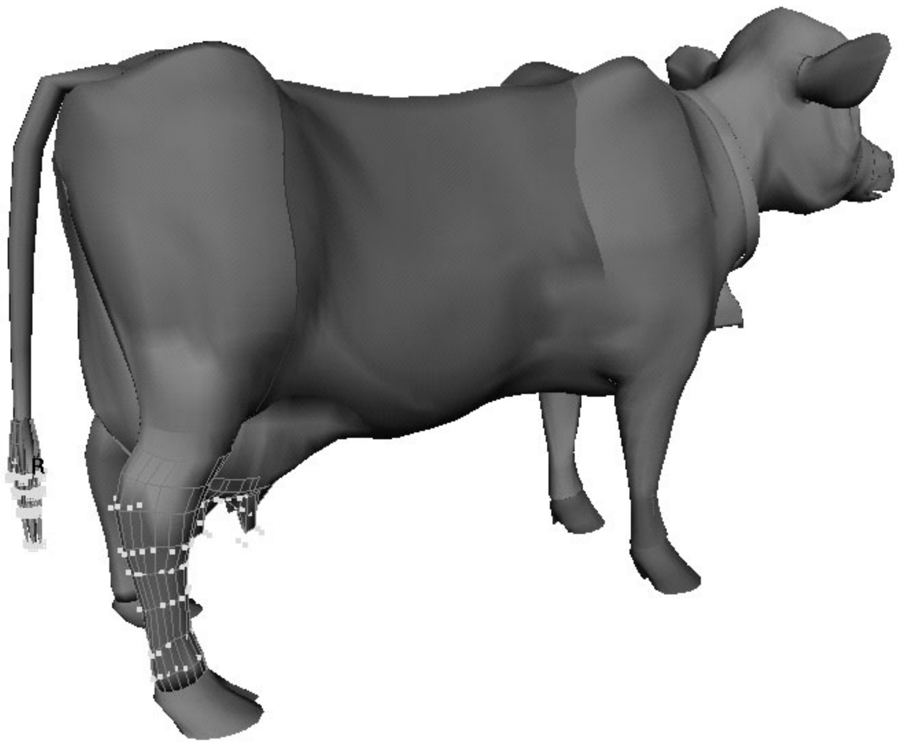
Note that painting rigid skin point weights uses a different painting tool than the tool for painting smooth skin point weights.

## Editing rigid skin point set membership



You can directly edit rigid skin point set membership with the Edit Membership Tool. For a more intuitive approach, you can also paint rigid skin point set memberships with the Paint Set Membership Tool (see "Painting rigid skin point set membership" on page 168). Note that you can also edit rigid skin point set memberships from the Relationship Editor (Window > Relationship Editors > Deformer Sets), but this approach is less intuitive than using the Edit Membership Tool or the Paint Set Membership Tool.

## Painting rigid skin point set membership



You can paint rigid skin point set membership in the same way you paint deformer set membership. For more detailed information, see “Paint deformer set membership” in Chapter 4.

## Manipulating the joint lattice flexor’s influence lattice

You can manipulate the joint lattice flexor’s influence lattice in the same way that you can manipulate the lattice deformer’s influence lattice. You can move, rotate, or scale the lattice, or you can move, rotate, or scale lattice points.

## Manipulating bone lattice flexor’s influence lattice

You can manipulate the bone lattice flexor’s influence lattice in the same way that you can manipulate the lattice deformer’s influence lattice. You can move, rotate, or scale the lattice, or you can move, rotate, or scale lattice points.

## Editing joint or bone sculpt flexor effects

### Manipulating the sculpt sphere

You can directly manipulate the sculpt flexor's sculpt sphere in the same way that you can manipulate a sculpt deformer's sculpt sphere.

# How do I? Set smooth skinning

## Set point tweaking

**To change point tweaking's deformation order**

1. In the scene, move the pointer over the skinned object and press the right mouse button.

   A marking menu appears.
2. From the marking menu, select Inputs > All Inputs.

   The List of input operations window appears for the selected object.
3. Move the pointer over the name of the tweak node (default name: tweak*n*) whose order you want to change. Press the middle mouse button, drag it over the name of the operation where you want point tweaking to take place, and release the mouse button.

**To disable a tweak node**

1. Open the tweak node's Attribute Editor.
2. In the Attribute Editor, open Node Behavior.
3. Set Node State to HasNoEffect.

**To enable a tweak node**

1. Open the tweak node's Attribute Editor.
2. In the Attribute Editor, open Node Behavior.
3. Set Node State to Normal.

## Set bind pose

**To go to bind pose**

1. Select the character's skeleton.
2. Select Skin > Go to Bind Pose.

The skeleton goes to the pose it had during binding.

## Reset bind pose from Command Line

You can reset the bind pose for the joints of your skinned character using the `dagPose` MEL command. For example:

```
dagPose -reset -name "name_of_bindPose joint_to_reset;"
```

| | |
|---|---|
| Warning | If your new bind pose is very dissimilar from the original bind pose, and you add a new influence object, then unwanted bind results will occur in the area of the skin that has changed. |

## Set maximum influences

### To set maximum influences

1. Select the skin object(s) whose maximum influences you want to edit.
2. Select Skin > Edit Smooth Skin > Set Max Influences.

   The Set Max Influences window appears.
3. In the Set Max Influences window, set Max Influences.
4. Click Apply to set the new value.

## Query skin weights

### To query skin point weights

1. Select the skin points whose weights you want to edit.
2. Select Window > General Editors > Component Editor.

   The *Component Editor* appears.

   The Component Editor displays the component data for currently selected components in the workspace.

   By default, the Component Editor updates dynamically as you select components in the workspace. Also, as you select components in the Component Editor, the workspace updates dynamically as well.
3. Click on the Smooth Skins tab.The Smooth Skins section lists the weights assigned to CVs, vertices, or lattice points bound to a skeleton’s joints by smooth skinning.

## Hold weights

### To hold skin point weights

When you are directly editing joint smooth skin influence object weights with the Component Editor, you can quickly specify whether the weights of particular joints or smooth skin influence objects can change. You can tell Maya to hold the weights (see ”Holding smooth skin weights” on page 153).

In the Component Editor, under the Smooth Skins tab, note the new Hold row. For each joint or smooth skin influence object, the value for Hold corresponds to the Hold Weights settings for the Attribute Editor’s Smooth Skin Parameters. On an individual skin point basis, the Hold settings correspond to the skinCluster node’s Lock Weights[*n*] attributes settings.

If Hold is on, Maya holds weights at their current values when you modify the weights of other influence objects. For instance, having Hold on prevents changes because of weight normalization (see ”Controlling smooth skin weight normalization” on page 154). However, you can still modify the weights by entering new values in the Component Editor.

## Paint skin weights

**Warning** If you want to paint smooth skin weights on a polygon mesh, then the mesh’s UV maps must be clean and free of overlapping UVs. Otherwise, undesired results will occur.

### To paint smooth skin point weights

1 Select the smooth skin objects you want to paint.
2 Go into smooth shading mode by selecting Shading > Smooth Shade All (hotkey: press 5).
3 Select the Paint Skin Weights Tool and open the Tool settings editor (Skin > Edit Smooth Skin > Paint Skin Weights Tool > ❐).
4 Check that Color Feedback is turned on in the Display section. Color feedback helps you identify the weights on the surface by representing them as grayscale values (smaller values are darker, larger values are lighter).

| Tip | You can use the default hotkey Alt+c (Windows, IRIX, and Linux) or Option+c (Mac OS X) to turn Color Feedback on and off outside the Tools Settings Editor. |
|---|---|

5. Select a joint. The values you paint set how much this joint influences the painted vertices relative to the other joints making up the smooth body (up to the number specified as the Max Influence in the Smooth Bind Skin Options window).
6. Do on of the following:
   - In the Influence section of the Tool Settings editor, click the joint name in the Transform box.
   - Right-click on the joint you want to paint to display a marking menu, then drag north and select Paint Weights. The Tool Settings editor must be open.

   Values with more influence appear lighter, values with less influence appear darker.
7. Select a brush, paint operation, and value and define other settings as required. See ”Deform > Paint Cluster Weights Tool” on page 397, noting that the settings are the same for the Paint Cluster Weights Tool.
8. Drag the brush across the skin.

| Tip | You can list joints Alphabetically or By Hierarchy in the Transform box. Alphabetically is best if you know the name of the joint that has the skin weights you want to paint.<br>By Hierarchy lists joints in the same order as the Outliner. The top of the list shows the root joint of the hierarchy, while each child joint is listed below its parent. This order is useful if you are painting a single a region of the skin—The joints you need to select from the list while painting are typically next to one another. |
|---|---|

## Paint skin weights for creases

### To paint creasing effects

1. Select smooth shaded display mode (hotkey: press 5).
2. Select the cylinder.
3. Select Skin > Edit Smooth Skin > Paint Skin Weights Tool > ❐.

(For more information, see ”Painting smooth skin point weights” on page 151.)

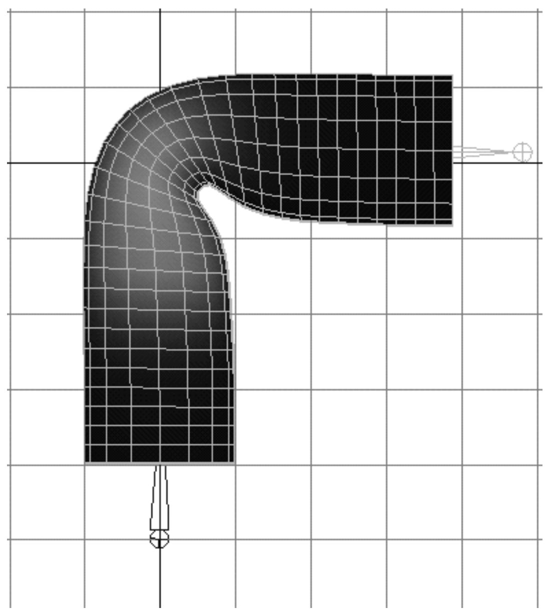
4 In the Tool Settings window, the Influence section should be displayed. For more information, see ”Skin > Edit Smooth Skin > Paint Skin Weights Tool” on page 216.

5 Note the Transform box.

The Transform box lists the names all the joints.

6 Click on a joint name. For example, click joint3.

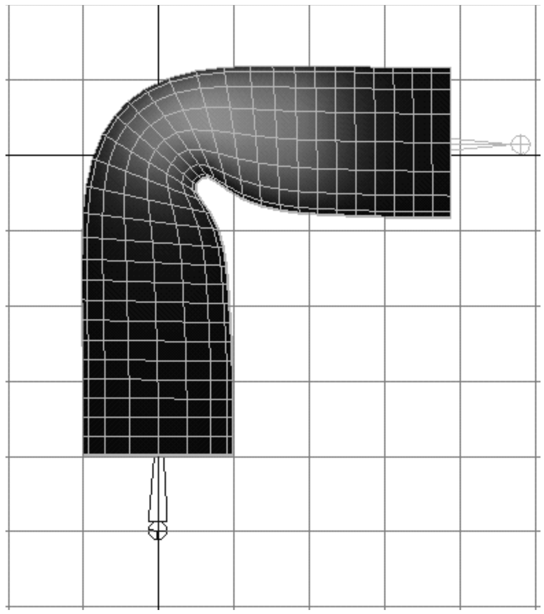
In the scene, the shading indicates the joint’s influence. The whiter the color, the greater the influence of the joint.



Influence of joint3.

Note how the joint’s influence fades into black as the distance from the joint increases.

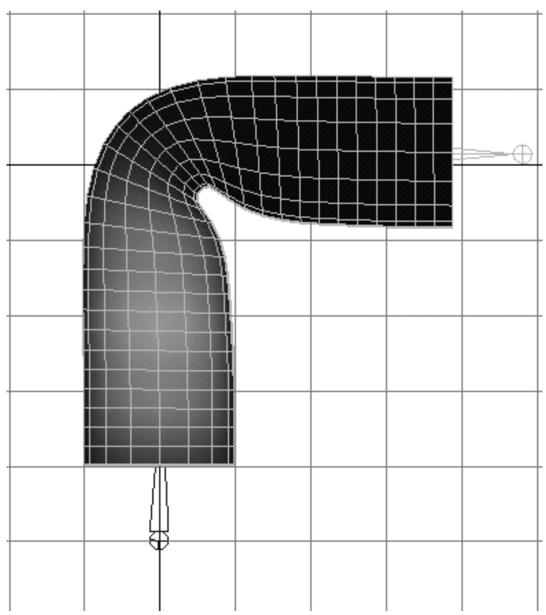
7 In the Influence box, click on another joint name. For example, click joint4.



Influence of joint4.

Again, note how the joint's influence fades into black as the distance from the joint increases. Also note how the joints gradually influence the bending and creasing.

**8** Check the influence of one more of the joints. For example, check the influence of joint2.



Influence of joint2.

Joint2 mainly influences the skin below the bend, but does provides some gradual influence in the bend area.

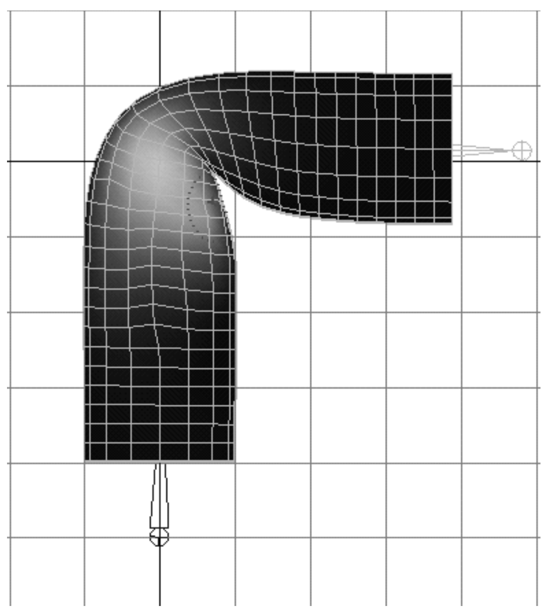
Similarly, joints above the bend (for example, joint5) also gradually influence the deformation in the bend area.

Now you will edit the influence of the joints to get a sharper creasing effect. You can do this by increasing how the joints nearest the bend influence the creasing, and lessening how joints further from the bend influence the creasing.

**9** Use the Paint Skin Weights Tool to paint how the joints influence creasing.

The brush provides an intuitive way to change the influence of the joints. Use the brush's Add operation to increase the influence of nearby joints, and use the Scale operation to decrease the influence of further joints. Use the Smooth operation to smooth out the influences of the joints. For more information on using paint tools, see *Artisan Brush Tool Settings* in the Paint Effects, Artisan, and 3D Paint guide.

Try to get the creasing to look something like the following, which is closer to what you might want for an elbow deformation.



Painting influences of near joints to edit the deformatic effect.
You can now see the influe of joint3.

Experiment with using the brush. With just a little experimentation, you can quickly become proficient at painting the skin point weights of joints.

## Mirror smooth skin weights

### To mirror smooth skin weights

1. To mirror skin weights on the entire skin, select the smooth skin object(s).

   To mirror skin weights on selected components, position the pointer over the skin, right-click and select Control Vertex or Vertex. Select the components on the side from which you want to mirror and the components on the side to be mirrored to.
2. Select Skin > Edit Smooth Skin > Mirror Skin Weights > ❐.

   The Mirror Skin Weights Options window appears.
3. Set the options.
4. Click the Mirror button.

## Add an influence object

### To set add influence options

1. If you plan to add an influence object immediately after setting options, be sure to go to the bind pose, and position the influence object appropriately.
2. Select Skin > Edit Smooth Skin > Add Influence > ❐.

   The Add Influence Options window appears.
3. Do one of the following:
   - Click Add to add the influence object now.
   - Click Save to save add influence options without adding influence object(s) now.
   - Click Reset to reset to the default add influence options.
4. Click Close to close the Add Influence Options window.

### To add an influence object

1. Go to the bind pose.
2. Position the influence object.
3. Select the skin, skin object(s) (for example, NURBS surfaces), or skin points (NURBS CVs or polygonal vertices) that you want the object(s) to influence.
4. Also select the influence object.
5. Select Skin > Edit Smooth Skin > Add Influence to add influence object(s) with previously set add influence options.

If the Use Geometry option is on (the default), Maya creates an influence object base node.

In the Outliner, an influence object base node is added (default name: *influenceObject*Base). Note that this is hidden by default, so you won't see it in the scene.

# Examples

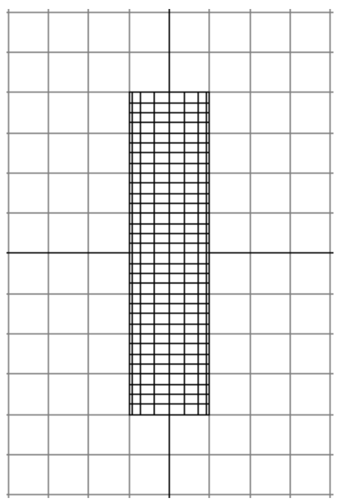
This section offers some examples of smooth skinning:

## Skinning a cylinder by smooth skinning

This example is similar to "Skinning a cylinder by rigid skinning" on page 205, so that you can compare smooth skinning with rigid skinning.

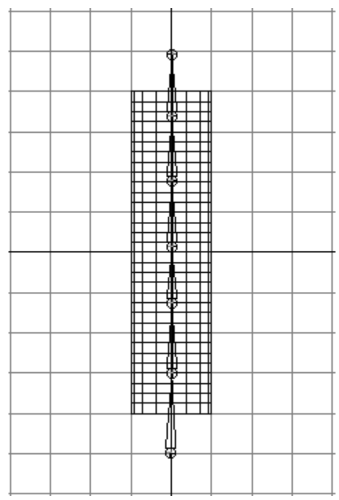
#### To create the cylinder

- Create a NURBS cylinder with the default options, except set Height to 8, Number of Sections to 16, and number of Spans to 32.



#### To create the skeleton for the cylinder

- Build a skeleton for the cylinder. Have the skeleton consist of a single joint chain with about seven joints.



Skeleton consisting of one joint chain (joint1 through joint7). Joint4 starts approximately in the center of the cylinder.

### To bind by smooth skinning

1 Select skeleton’s root joint (default name: joint1).

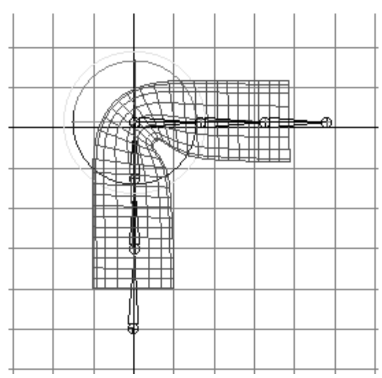
2 Select Skin > Bind Skin > Smooth Bind.

Maya binds the cylinder to the skeleton by smooth skinning, using the default bind skin options. The cylinder is now a smooth skin object. For more information on binding smooth skin, see ”Bind smooth skin” on page 185.

Now you can exercise the skeleton and get immediate deformation effects appropriate for the character.

### To exercise skeleton

1 Select the joint approximately at the center of the cylinder (for instance, joint4), and rotate it about 90 degrees.



Joint4 rotated 90 degrees.

Note that smooth skinning provides a smooth deformation effect around the rotated joint. However, the creasing might be a bit too rounded for the deformation of a character's limb. For example, if you were setting up the deformation around a character's elbow, you might want the creasing to be a bit sharper at the inside angle of the bend, though still rounded around the rest of the joint. You can adjust the deformation effect with the Paint Skin Weights Tool.

### To paint creasing effects

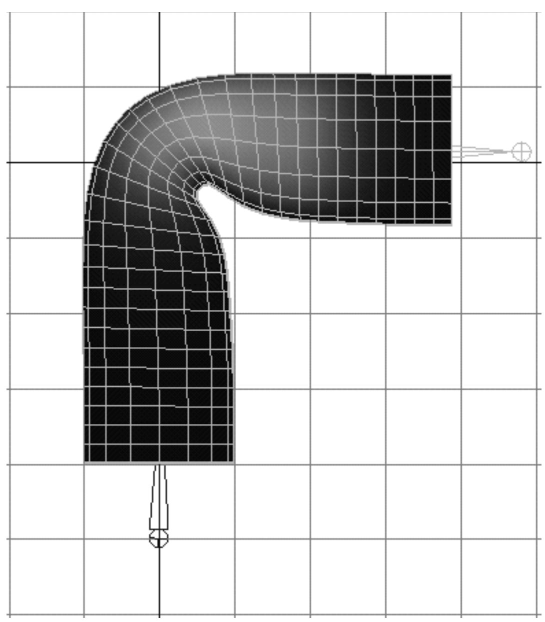
1. Select smooth shaded display mode (hotkey: press 5).
2. Select the cylinder.
3. Select Skin > Edit Smooth Skin > Paint Skin Weights Tool > ❐.

   See "Painting smooth skin point weights" on page 151.
4. In the Tool Settings window, the Skin Paint tab should be displayed.
5. Note the Influence box.

   The Influence box lists the names all the joints.
6. Click on a joint name. For example, click joint3.

   In the scene, the shading indicates the joint's influence. The whiter the color, the greater the influence of the joint.

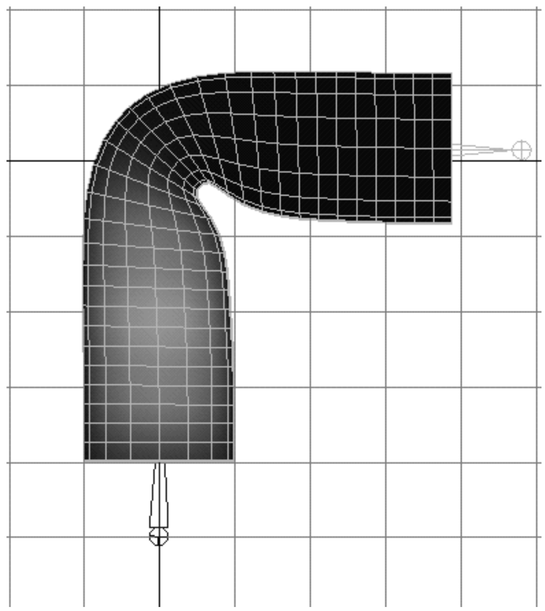
   Note how the joint's influence fades into black as the distance from the joint increases.
7. In the Influence box, click on another joint name. For example, click joint4.



Influence of joint4.

Again, note how the joint's influence fades into black as the distance from the joint increases. Also note how the joints gradually influence the bending and creasing.

**8** Check the influence of one more of the joints. For example, check the influence of joint2.



Influence of joint2.

Joint2 mainly influences the skin below the bend, but does provides some gradual influence in the bend area.

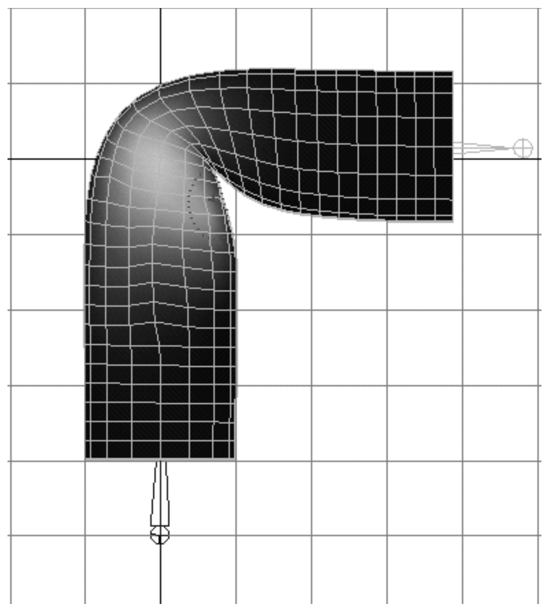
Similarly, joints above the bend (for example, joint5) also gradually influence the deformation in the bend area.

Now you will edit the influence of the joints to get a sharper creasing effect. You can do this by increasing how the joints nearest the bend influence the creasing, and lessening how joints further from the bend influence the creasing.

**9** Use the Paint Skin Weights Tool's brush to paint how the joints influence creasing.

The brush provides an intuitive way to change how the influence of the joints. Use the brush's Add operation to increase the influence of nearby joints, and use the Scale operation to decrease the influence of further joints. Use the Smooth operation to smooth out the influences of the joints. For more information on using paint tools, see *"The paint tools"* the Paint Effects, Artisan, and 3D Paint guide.

Try to get the creasing to look something like the following, which is closer to what you might want for an elbow deformation:



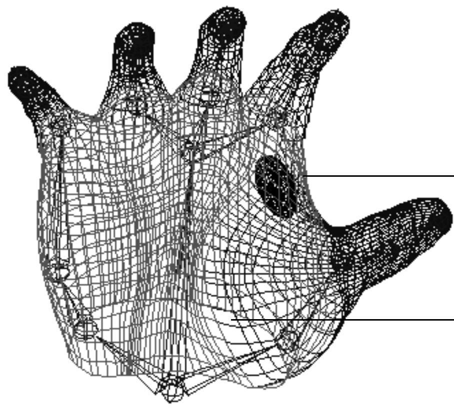

Experiment with using the brush. With just a little experimentation, you can quickly become proficient at painting the skin point weights of joints.

## Skinning a hand

| | |
|---|---|
| **Warning** | If you want to paint smooth skin weights on a polygon mesh, then the mesh's UV maps must be clean and free of overlapping UVs. Otherwise, undesired results will occur. |

#### To create the polygonal sphere to simulate a muscle bulge

1. Create a polygonal sphere, adjusting the scale attributes to approximate the muscle shape (for example, set Scale X to 1.5, Scale Y to 0.7, and Scale Z to 0.7).
2. Position the sphere inside hand, between thumb and index finger.



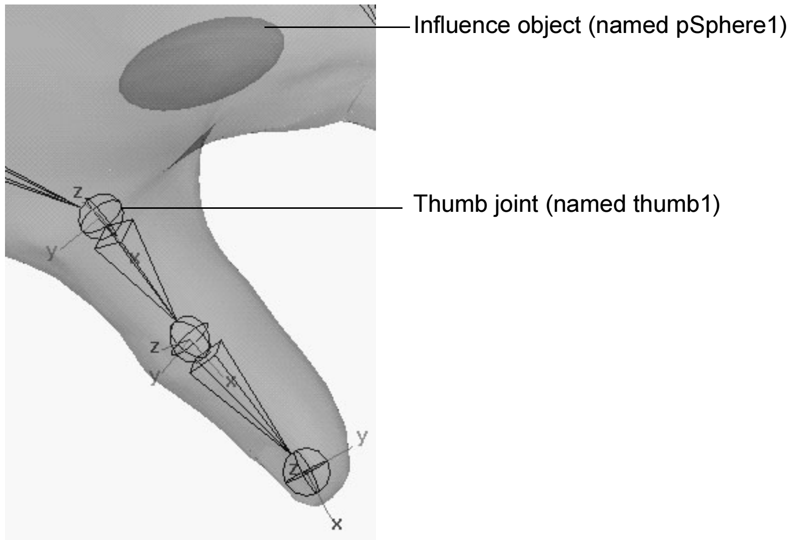

### To make the polygonal surface an influence object

1. Select palm skin object. (This is the smooth skinned NURBS surface that the sphere will deform.)
2. Select the sphere.
3. Select Skin > Edit Smooth Skin > Add Influence.

   (For more information, see "Add an influence object" on page 176.)

### To link the bulge to fist formation

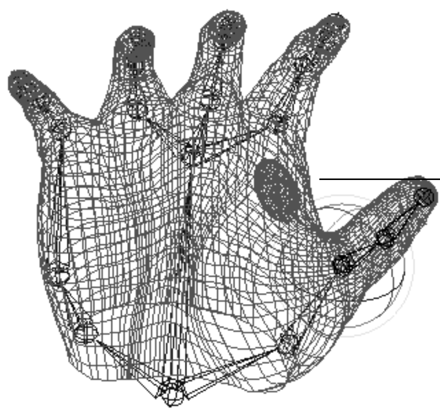
Now you will link the rotation of the thumb joint to the scaling of the influence object (the polygonal sphere).



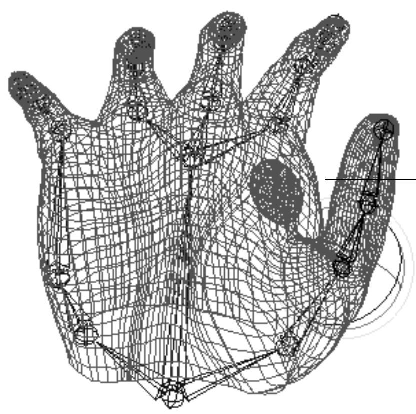

1. Open the Set Driven Key window (Animate > Set Driven Key > Set ❒).
2. Load thumb1 (the thumb joint) as driver, select rotate Z attribute, and set the attribute to 0.
3. Load pSphere1 (the influence object) as driven, select the scale Y and scale Z attributes. (Keep scale Y and scale Z at 0.7.)
4. Click Key.
5. Set thumb1’s rotate Z attribute to -40.
6. Set pSphere1’s scale Y attribute to 0.8.
7. Set pSphere1’s scale Z attribute to 1.
8. Click Key.
9. Click Close to close the editor.

### Testing the deformation

Now when the thumb is away from the palm, the muscle will appear to be relaxed.




But as you rotate the thumb towards the palm, the surface bulges to indicate muscle action.




## Edit smooth skinning

### Change a skinned object's deformation order

When you use skinning with one or more deformers to deform an object, the final effect of the deformations can vary depending on the order in which the deformations occur. By default, the deformations occur in the order that skin was bound and the deformers were created. The skinning or deformer algorithm node created first deforms the object first, and the algorithm node created last deforms the object last. However, you can change, or re-order, the deformation order to get the effect you want.

#### To change skinned object's deformation order

**1** In the scene, move the pointer to the object being deformed and press the right mouse button.

A marking menu is displayed.

**2** From the marking menu, select Inputs > All Inputs.

The List of input operations window is displayed for the selected object.

Note that by default smooth skinning algorithm nodes are named skinCluster*n*, and rigid skinning algorithm nodes are named jointCluster*n*.

**3** Move the pointer over the name of the skinning or deformer algorithm node whose order you want to change. Press the middle mouse button, drag it over the name of the algorithm node you want the node to precede, and release the mouse button.

## Bind smooth skin

Binding smooth skin includes setting the smooth bind options, binding skin, and then checking the binding by exercising the skeleton.

If you'd like to explore an example of binding smooth skin, see "Skinning a cylinder by smooth skinning" on page 177.

#### To set bind options

**1** If you also want to bind skin now, select the skeleton (or joints) and then the deformable object(s) you want to bind.

**2** Select Skin > Bind Skin > Smooth Bind > ❒.

The Smooth Bind Skin Options window is displayed.

**3** Set the options.

**4** Do one of the following:

- Click Bind to bind skin.

Maya binds skin using the smooth bind options you set. For each deformable object, Maya creates a skin cluster node, making each object a smooth skin object.

- Click Save to save creation options without binding skin.
- Click Reset to reset to default skin cluster options.

## Edit joint smooth skin attributes

When you bind smooth skin to a skeleton, Maya assigns each joint some additional attributes. Maya places these attributes in each joint's Attribute Editor, in a section called Smooth Skin.

### To edit attributes with the Attribute Editor

1. Select a joint.
2. Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).
3. Open the Smooth Skin tab.

## Edit skin cluster channels

### To edit channels with the Channel Box

1. Select a skin cluster node (default name: skinCluster*n*).

   One quick way to select a skin cluster node is to select a smooth skin object, and then select the skin cluster node in its history from the Channel Box (under INPUTS). Skin cluster nodes are also part of the history of bound joints (under OUTPUTS).

   Each smooth skin object has its own input skin cluster node.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Set the channels.
3. Click on a channel name with the left mouse button.
4. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you coarser control.

## Edit skin cluster attributes

### To edit attributes with the Attribute Editor

1. Select a skin cluster node (default name: skinCluster*n*).
2. Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).
3. The following sections make available attributes: Smooth Skin Attributes, Deformer Attributes, Node Behavior, and Extra Attributes.

# Modify skin weights

### To modify a skin point's weight

1. In the Component Editor's spreadsheet, click the component data box you want to edit.

   Only the component whose box you've selected is now selected in the workspace.
2. Enter a new value.

### To modify several skin point weights at once

1. In the workspace, select the points whose weights you want to edit.
2. In the Component Editor's spreadsheet, drag through the component data boxes you want to edit.
3. Enter the value you want all the boxes to have.

### To modify an entire row or column (IRIX, Linux, and Mac OS X only)

1. In the workspace, select the points whose weights you want to edit.
2. In the Component Editor's spreadsheet, click one of the boxes in the row or column.
3. Click the row or column heading.

   Now all the boxes for the row or column are selected.
4. Enter a value for all the boxes in the row or column.

### To modify an entire row or column (Windows only)

1. In the workspace, select the points whose weights you want to edit.
2. To change all the entries of a row or column, in the Component Editor's spreadsheet, select the row or column heading.
3. Shift-select any of the boxes in that row or column.
4. Enter a new value to update the entire row or column.

# Copy smooth skin weights

### To copy smooth skin weights

1. To copy skin weights of the entire skin, select the smooth skin object (or group of objects) whose weights you want to copy, and then select the object (or group of objects) to which you want to copy the weights.

To copy skin weights of selected components, position the pointer over the skin, right-click and select Control Vertex or Vertex. Select the components on the source skin and components on the destination skin.

2 Select Skin > Edit Smooth Skin > Copy Skin Weights.

## Reset skin point weights to default weights

While you are editing or painting the skin weights, you can reset the weights the their initial, default values at any time.

### To reset skin point weights

1 Select the skin object (or specific components on the object) whose skin point weights you want to reset.

2 Select Skin > Edit Smooth Skin > Reset Weights to Default.

## Normalize weights

### To disable normalization

1 Select the smooth skin object(s) whose weights you no longer want normalized automatically.

2 Select Skin > Edit Smooth Skin > Disable Weight Normalization.

Now Maya will not normalize the weights automatically.

### To enable normalization

1 Select the smooth skin object(s) whose weights you want to be normalized automatically.

2 Select Skin > Edit Smooth Skin > Enable Weight Normalization.

Maya does not immediately normalize weights. The weights stay at their current values until you edit or paint weights, and then Maya will automatically normalize the weights.

### To normalize weights

1 Select the smooth skin objects (or skin points) whose weights you want to normalize.

2 Select Skin > Edit Smooth Skin > Normalize Weights.

Maya normalizes the weights.

## Prune insignificant smooth skin weights

As you paint smooth skin weights, you might unintentionally create small weight values, for instance, 0.008, on many of the skin points. Maya creates small values as it normalizes total skin weight for each point to 1. The weights have no discernible effect on the skin, but they slow down processing. You can prune the small skin weights to speed up processing.

#### To prune small skin weights

1 Select the skin or specific skin points.
2 Select Skin > Edit Smooth Skin > Prune Small Weights > ❐.
3 In the Prune Below box, set the threshold weight.

   The default value, 0.01 works well in most cases. Skin weights below the valued entered will be reset to 0. By default, the remaining skin weights are normalized to add up to 1. If you want to prevent normalization, select the skin object, display the skinCluster tab in the Attribute Editor, and turn off Normalize Weights.

## Remove unused influences from a smooth skinned surface

To enhance Maya processing speed and make the Paint Skin Weights Tool easier to use, you can disconnect joints and influence objects from a smooth skin that has all of its skin weights at 0. The Paint Skin Weights Tool becomes easier to use because the unused influences do not appear in the Influence list for the skin.

For example, suppose you smooth skin a left foot stocking to a human skeleton with the default options. All joints in skeleton will be connected to the stocking as potential influences, but only the joints closest to the left foot stocking will have nonzero weights.

If you select the elbow and rotate it, Maya computes the skin on the stocking to see if there are any nonzero weights. If you select the stocking and remove the unused influences, the elbow and the other zero-weighted joints will be disconnected from the stocking and performance will improve.

#### To remove unused influences

1 Select the skin.
2 Select Skin > Edit Smooth Skin > Remove Unused Influences.

# Export weights

### To export weight maps

1. Select the skin object or objects. If objects are parented or grouped, select each object individually.
2. Select Skin > Edit Smooth Skin > Export Skin Weight Maps > ❒.
3. Set the following options and click Export. In most applications, only the Map Size X and Y options are useful. The other options are useful if you plan to use specialized map editing techniques. Most users use the default option settings.
4. In the Write file browser that appears, specify a path and name for a folder (or directory) that will be created to hold the map files. By default, Maya puts the folder name you specify under the *sourceimages* folder of your current project. Click the Write button in the file browser after you enter the name for the folder.
5. Maya lists how many maps will be written to disk, and prompts whether you want to proceed.
6. Click Yes.

   The operation has finished when the hourglass icon stops flashing on your screen. It may take 10 or more seconds per map. Maya puts the image files, one per joint, in the folder you specified. For example:

   ```
   jackie_back_root.iff
   jackie_jaw.iff
   jackie_left_ankle.iff
   ```

   Maya also creates a weight map file named *folder*.weightMap in the same location as the folder you specified for the images (*sourcesimages* by default). The contents of the weight map file defines the relationship between surface geometry and the image files. You can view and edit the contents of this file using a basic text editor such as Note Pad. Maya uses the weight map file to successfully import the images. When importing the weight map, the weight map file must be in the same location as the folder that contains the image files. See ”Import weights” on page 190.

# Import weights

| | |
|---|---|
| **Warning** | When importing painted smooth skin weight maps from a polygon mesh, the mesh’s UV maps must be free of any overlapping UVs. Otherwise, undesired results will occur. |

#### To import weight maps

1. Select the skin object or objects to receive the maps. If objects are parented or grouped, select each object individually.
2. Select Skin > Edit Smooth Skin > Import Skin Weight Maps.
3. In the file browser that appears, specify the name of the previously exported .weightMap file for the maps you want to import.

   To import the maps to a skin or skeleton that has a different name than the skin or skeleton from which you exported, you must open the .weightMap file with a text editor and replace the skin or skeleton names with the ones used in the scene where you import the maps.

## Detach smooth skin

After you have bound skin, you might decide that you want to modify the skeleton, change the bind pose, or do some further modeling on the skin. To do these things you must first detach the skin from the skeleton, and then when you're done you must bind skin again.

#### To detach skin

1. If you want to detach now, select the skin object(s) you want to detach.
2. Select Skin > Detach Skin > ❐.

   The Detach Skin Options window appears.
3. Do one of the following:
   - Click Detach to detach skin.
   - Click Save to save detach options without detaching skin.
   - Click Reset to reset to default detach skin options.

# Set rigid skinning

## Set rigid bind options

#### To set bind options

1. If you also want to bind skin now, select the skeleton (or joints) and then the deformable object(s) you want to bind.
2. Select Skin > Bind Skin > Rigid Bind > ❐.

   The Rigid Bind Skin Options window appears.
3. Do the following:
   - Click Bind Skin if you want to bind skin now.

- Click Save to save the options.
- Click Reset to reset to the default options.
- Click Close to close the Rigid Bind Skin Options window.

## Bind skin

### To bind skin

1. Select one or more deformable objects, followed by a skeleton’s root joint or a limb’s parent joint.
2. Select Skin > Bind Skin > Rigid Bind.

   Maya binds skin using the previously set rigid bind options.

### To go to bind pose

1. Select the character’s skeleton.
2. Select Skin > Go to Bind Pose.

   The skeleton goes to the pose it had during binding.

## Reset bind pose from Command Line

You can reset the bind pose for the joints of your skinned character using the `dagPose` MEL command. For example:

```
dagPose -reset -name "name_of_bindPose joint_to_reset;"
```

| | |
|---|---|
| Warning | If your new bind pose is very dissimilar from the original bind pose, and you add a new influence object, then unwanted bind results will occur in the area of the skin that has changed. |

### To edit Rigid channels with the Channel Box

1. Select a joint cluster node (default name: joint*n*Cluster*n*).

   One quick way to select a joint cluster node is to select a rigid skin object, and then select the joint cluster node in its history from the Channel Box (under INPUTS). Alternatively, select a joint and then select the joint cluster node in its history from the Channel Box (under OUTPUTS).

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Edit the channels.

3 Click on a channel name with the left mouse button.

4 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

## Paint rigid skin weights

### To paint weights on a rigid bound skin

1 Select the rigid skin object you want to paint weights on.

2 Go into smooth shading mode (select Shading > Smooth Shade All or press 5 (default hotkey).

3 Select Deform > Paint Cluster Weights Tool > ❐.

4 Check that Color Feedback is turned on in the Display section. Color feedback helps you identify the weights on the surface by representing them as grayscale values (smaller values are darker, larger values are lighter).

| | |
|---|---|
| Tip | You can use the default hotkey Alt+c (IRIX, Linux, and Windows) or Option+c (Mac OS X) to turn Color Feedback on and off outside the Tools Settings Editor. |

5 Select the joint cluster you want to paint weights on. In the Paint Attributes section of the Tool Settings window, click the jointCluster*n*.weights button and select the appropriate joint cluster weights name from the pop-up menu.

Note that you can only paint weights on one cluster at a time. If you select more than one cluster, you can only paint weights on the active one (the one that provides color feedback).

If the surface has only one cluster, you can select the surface alone.

| | |
|---|---|
| Tip | If you are painting on a single surface, you can skip step 3 and select the joint cluster without opening the Tool Settings window by right-clicking the surface and selecting the appropriate joint cluster weights name from the Paint command submenu. |

6 Select a brush, paint operation, and value and define other settings as required. See ”Deform > Paint Cluster Weights Tool” on page 397.

7 Drag the brush across the cluster.

> Tip You can quickly pick weight values from one cluster and paint them on another cluster or the same cluster using hotkeys. Do the following:
>
> - Select the cluster with the weight values you want to pick.
> - Hold down the Pick Color Mode hotkey (default hotkey: /), click on the area of the cluster with the weight you want to pick, then release the hotkey.
> - If you are painting the picked weight on the same cluster, drag the brush across the cluster.
> - If you are painting the picked weight on another cluster, select that cluster, then drag the brush across it.

## Paint rigid skin membership

### To paint rigid skin point set membership

1 Select the rigid skin object(s).

2 Go into smooth shading mode (hotkey: press 5).

3 Select Deform > Paint Set Membership Tool > ❐.

The SetMembership tab should be selected.

4 In the Set Membership box, select the joint set with the point memberships you want to edit.

5 Use the brush to add, transfer, or remove set memberships.

For more information about painting tools, see *“How Artisan brush tools work”* the Paint Effects, Artisan, and 3D Paint guide.

### To paint creasing effects

1 Select smooth shaded display mode (hotkey: press 5).

2 Select the cylinder.

3 Select Deform > Paint Cluster Weights Tool > ❐.

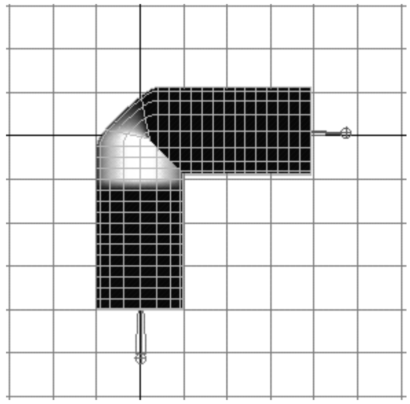
(For more information, see ”To paint weights on a rigid bound skin” on page 193.)

4 In the Paint Weights section of the Tool Settings window, notice the joint1Cluster1.weights button.

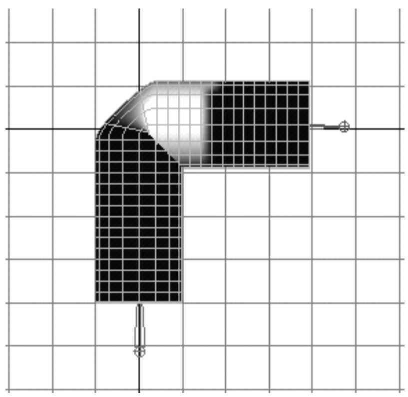
Click this button to list the names of all the rigid skin point clusters.

5 Select a rigid skin point cluster. For example, cluster-joint3Cluster1 > weights.

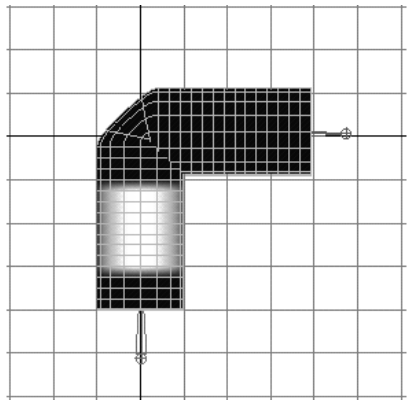
In the scene, the shading indicates the weighting of each point in the set.



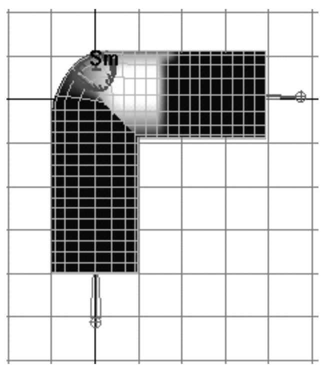
**6** Select another rigid skin point cluster. For example, select cluster-joint4Cluster1 > weights.



**7** Check the other rigid skin point cluster. For example, check cluster-joint2Cluster1.



8 Use the Paint Cluster Weights Tool's brush to smooth the deformation effect.




The brush provides an intuitive way to change the influence of the joints. Use the brush's Add operation to increase the influence of nearby joints, and use the Scale operation to decrease the influence of further joints. Use the Smooth operation to smooth out the influences of the joints.

Experiment with using the brush. With just a little experimentation, you can quickly become proficient at painting the skin point weights of joints. For more information on using paint tools, see the Paint Effects, Artisan, and 3D Paint guide.

To further smooth and deform rigid skinning, you can use flexors. For more information, see "Create all types of flexors" on page 196.

## Create all types of flexors

### To create flexors

1 Put the skeleton into bind pose.

You can create a flexor if the skeleton is not in its bind pose, but you might get unexpected deformation effects.

2. If you want to create one or more joint lattice, joint sculpt, or joint cluster flexors, select the joints on which you want to create the flexors.
3. If you want to create one or more bone lattice or bone sculpt flexors, select the bones on which you want to create the flexors.

   Note that by default bone lattice flexors are driven by the child joints of the joints whose bones you select for them. This is what you would usually want for most character setup situations. However, after you create the bone lattice flexors, you can assign any other joint in the skeleton to drive them. See ”Reassign bone lattice flexor joints” on page 203.
4. Select Skin > Edit Rigid Skin > Create Flexor.

   The Create Flexor window appears.
5. Do one of the following:
   - Click Create to create flexors now.
   - Click Close to save the options and close the window.

# Edit rigid skinning

## Edit joint cluster attributes

### To edit attributes with the Attribute Editor

1. Select a skin cluster node (default name: jointCluster*n*).
2. Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).
3. Set the attributes.

## Edit rigid skin point weights

With the Component Editor, you can directly modify the values of individual rigid skin point weights.

### To query rigid skin point weights

1. Select the rigid skin points whose weights you want to edit.
2. Select Window > General Editors > Component Editor.

   The *Component Editor* appears.

   The Component Editor displays the component data for currently selected components in the workspace.

By default, the Component Editor updates dynamically as you select components in the workspace. Also, as you select components in the Component Editor, the workspace updates dynamically as well.

3. Click on the Rigid Skins tab. The Rigid Skins section lists the weights assigned to CVs, vertices, or lattice points bound to a skeleton's joints by rigid skinning.

### To modify a rigid skin point's weight

1. In the Component Editor's spreadsheet, click the component data box you want to edit.

   Only the component whose box you've selected is now selected in the workspace.
2. Enter a new value.

### To modify several rigid skin point weights at once

1. In the workspace, select the points whose weights you want to edit.
2. In the Component Editor's spreadsheet, drag through the component data boxes you want to edit.
3. Enter the value you want all the boxes to have.

### To modify an entire row or column (IRIX, Linux, and Mac OS X only)

1. In the workspace, select the points whose weights you want to edit.
2. In the Component Editor's spreadsheet, click one of the boxes in the row or column.
3. Click the row or column heading.

   Now all the boxes for the row or column are selected.
4. Enter a value for all the boxes in the row or column.

### To modify an entire row or column (Windows only)

1. In the workspace, select the points whose weights you want to edit.
2. To change all the entries of a row or column, in the Component Editor's spreadsheet, select the row or column heading.
3. Shift-click any of the boxes in that row or column.
4. Enter a new value to update the entire row or column.

# Edit rigid skin membership

**To edit set membership with the Edit Membership Tool:**

1. Go into object selection mode (click the select by object type icon).
2. Select the joint whose set you want to edit.
3. Go into component selection mode (click the select by component type icon).
4. Select Deform > Edit Membership Tool.
5. Using the pointer, select the points whose rigid skin point set membership you want to change.

   The members of the rigid skin point set whose joint you selected are displayed in yellow. This set is the currently selected set. Members of other sets are displayed in the colors associated with the sets' joints. Points displayed in dark red are not in a set.
6. To add points to the currently selected set, select them while pressing the Shift key and left mouse button, and then release the mouse button.

   The selected points are now displayed in yellow, indicating they are in the currently selected set.
7. To remove points from the currently selected set, select them while pressing the Ctrl key and the left mouse button, and then release the mouse button.

   The selected points are now displayed in dark red, indicating they are currently not in a set. Points that are not in rigid skin point set will not be affected by the skeleton's actions, so in general you'll want to have all the points in a set. Of course, the points do not necessarily have to be in a rigid skin point set; they could be in, for example, a cluster deformer set whose handle is parented to some part of the skeleton's hierarchy.
8. To add points to some other rigid skin point set, first select the rigid skin point set's joint. The points currently in the set are displayed in yellow. Now, as before, select the points you want to add while pressing the Shift key and the left mouse button, and then release the mouse button.

# Detach rigid skin

After you have bound skin, you might decide that you want to modify the skeleton, change the bind pose, or do some further modeling on the skin. To do these things you must first detach the skin from the skeleton, and then when you're done you must bind skin again.

Detaching skin does not preserve the rigid skin point sets and the rigid skin point weights. If you want to preserve the rigid skin point sets, see ”Detach and reattach to skeletons” on page 200 and ”Detach and reattach to selected joints” on page 201.

### To set detach skin options

1 If you want to detach now, select the skin object(s) you want to detach.

2 Select Skin > Detach Skin > ❐.

The Detach Skin Options window appears.

3 Do the following:

- Click Detach to detach skin.
- Click Save to save detach options without detaching skin.
- Click Reset to reset to default detach skin options.

### To detach skin

1 Select skeleton(s).

2 Select Skin > Detach Skin to detach skin with previously set detach skin options.

Unless the History detach skin option was set to Bake History, the skin objects move to their undeformed, bind pose positions. Their transform (Translate, Rotate, and Scale) attributes are unlocked. Unless the History detach skin option was set to Keep History, Maya deletes input rigid skinning (jointCluster) nodes of the objects.

## Detach and reattach to skeletons

Detaching a skeleton from its skin objects unlocks the transform attributes of the objects so that you can reposition them. Unlike detaching skin, detaching skeletons preserves the rigid skin point sets and the rigid skin point weights. Detaching and reattaching a skeleton is especially useful if you want to move, rotate, or scale the skin objects directly while not changing which joints influence which rigid skin points.

### To detach skeleton

1 Select the skeleton’s root joint, or any joint on the skeleton.

2 Select Skin > Edit Rigid Skin > Preserve Skin Groups > Detach Skeleton.

See ”Skin > Edit Rigid Skin > Preserve Skin Groups > Detach Skeleton” on page 223.

The skin objects that were being influenced by the now detached skeleton move to their undeformed positions. The transform (Translate, Rotate, and Scale) attributes (or channels) of the objects are now unlocked, so you can now move, rotate, or scale the objects.

#### To reattach skeleton

1. Select the skeleton's root joint, or any joint on the skeleton.
2. Select Skin > Edit Rigid Skin > Preserve Skin Groups > Reattach Skeleton.

   See "Skin > Edit Rigid Skin > Preserve Skin Groups > Reattach Skeleton" on page 223.

## Detach and reattach to selected joints

This procedure is similar to detaching and reattaching a skeleton, except that it only applies to selected joints.

Detaching selected joints from the skin objects they influence unlocks the transform attributes of the objects so that you can reposition them. Detaching selected joints preserves the rigid skin point sets and the rigid skin point weights. Detaching and reattaching selected joints is especially useful if you want to move, rotate, or scale certain skin objects directly while not changing which joints influence which rigid skin points.

#### To detach selected joints

1. Select the joints you want to detach.
2. Select Skin > Edit Rigid Skin > Preserve Skin Groups > Detach Selected Joints.

   See "Skin > Edit Rigid Skin > Preserve Skin Groups > Detach Selected Joints" on page 223.

   The skin objects that were being influenced by the now detached joints move to their undeformed positions. The transform (Translate, Rotate, and Scale) attributes (or channels) of the objects are now unlocked, so you can now move, rotate, or scale the objects.

#### To reattach selected joints

1. Select the joints you want to reattach.
2. Select Skin > Edit Rigid Skin > Preserve Skin Groups > Reattach Selected Joints.

   See "Skin > Edit Rigid Skin > Preserve Skin Groups > Reattach Selected Joints" on page 224.

## Copy joint lattice flexors

After you create a lattice flexor and adjust it, you might decide you want a similar lattice flexor elsewhere. For example, if you create a lattice flexor around a character's left elbow joint, you might then decide to create a similar lattice flexor at the right elbow joint.

You can now copy joint lattice flexors. When you copy, all of the attribute values and connections are copied.

Note that you can also copy bone lattice flexors.

### To copy joint lattice flexors:

1. Go to the bind pose (Skin > Go to Bind Pose).
2. Select a flexor (for example, select a lattice flexor called jointFfd*n*Lattice).
3. Select the joint where you want to have a copy of the flexor.
4. Skin > Edit Rigid Skin > Copy Flexor.

## Edit joint lattice flexor channels

Channels are the keyable attributes displayed in the Channel Box. The Channel Box provides a convenient way to edit a joint lattice flexor's channels.

### To edit channels with the Channel Box

1. Select a joint lattice flexor node (default name: jointFlexor*n*).

   One quick way to select the flexor node is to select the flexor's influence lattice, and then select the joint lattice flexor node in its history from the Channel Box (under SHAPES).

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Set channels.
3. Click on a channel name with the left mouse button.
4. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

## Copy bone lattice flexors

**To copy bone lattice flexors:**

1. Go to the bind pose (Skin > Go to Bind Pose).
2. Select a flexor (for example, select a lattice flexor called jointFfd*n*Lattice).
3. Select the bone where you want to have a copy of the flexor.
4. Select Skin > Edit Rigid Skin > Copy Flexor.

## Edit bone lattice flexor channels

Channels are the keyable attributes displayed in the Channel Box. The Channel Box provides a convenient way to edit a bone lattice flexor's channels.

By default, the effects specified by the bone lattice flexor channels are driven by the child joints of the joints whose bones you selected for them when you created the flexors. This is what you would usually want for most character setup situations. However, you can assign any other joint in the skeleton to drive them (see "Reassigning bone lattice flexor joints" on page 298).

**To edit channels with the Channel Box**

1. Select a bone lattice flexor node (default name: boneFlexor*n*).

   One quick way to select the flexor node is to select the flexor's influence lattice, and then select the bone lattice flexor node in its history from the Channel Box (under SHAPES).

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Set the Channels.
3. Click on a channel name with the left mouse button.
4. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

## Reassign bone lattice flexor joints

**To reassign bone lattice flexor joints**

1. Select the bone lattice flexor's lattice.

2 Select the joint that you want to have drive the bone lattice flexor.

   Note that the bone lattice flexor cannot be assigned to a bone whose joint is the skeleton's root joint.

3 Select Skin > Edit Rigid Skin > Reassign Bone Lattice Joint.

   The bone lattice flexor is now driven by the joint you've selected. Remember that to see the effects of the bone lattice flexor, you must set the bone lattice flexor's channels to values other than zero. For more information, see "Edit bone lattice flexor channels" on page 203.

## Edit sculpt flexor channels

Channels are the keyable attributes displayed in the Channel Box. The Channel Box provides a convenient way to edit a bone lattice flexor's channels.

#### To edit channels with the Channel Box

1 Select a bone lattice flexor node (default name: boneFlexor*n*).

   One quick way to select the flexor node is to select the flexor's influence lattice, and then select the bone lattice flexor node in its history from the Channel Box (under SHAPES).

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).

2 Set the channels.

3 Click on a channel name with the left mouse button.

4 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

## Edit with joint cluster flexor manipulators

Cluster flexors include manipulators you can use to edit their deformation effects. A cluster flexor's manipulators include a pair of rings. Each ring includes two manipulators: a diamond manipulator and a radial manipulator. Located at the center of the ring and along the center of a bone, the diamond manipulator controls the extent of the smoothing provided by the cluster flexor. Located on the ring, the radial manipulator controls the magnitude of the smoothing.

**To edit with the cluster flexor manipulators**

1. Select the cluster flexor handle (the J icon).
2. Select the Show Manipulator Tool.
3. To edit the extent of smoothing, select one of the diamond manipulators.
4. Use the left mouse button to drag the diamond manipulator towards or away from the joint.

   The extent of smoothing changes as you drag the manipulator. Note that the joint cluster’s Upper Bound or Lower Bound channels change as you drag. For more information on these channels, see ”Editing joint cluster channels” on page 166.
5. To edit the magnitude of smoothing, select one of the radial manipulators.
6. Use the left mouse button to drag the radial manipulator towards or away from the diamond manipulator.

   The magnitude of smoothing changes as you drag the manipulator. Note that the joint cluster’s Upper Value or Lower Value channels change as you drag. For more information on these channels, see ”Editing joint cluster channels” on page 166.

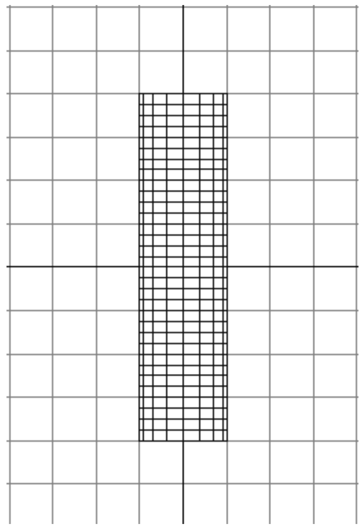
## Example

### Skinning a cylinder by rigid skinning

This example is similar to ”Skinning a cylinder by smooth skinning” on page 177, so that you can compare rigid skinning with smooth skinning.

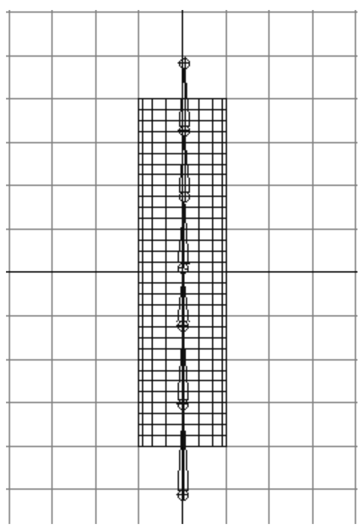
**To create the cylinder**

- Create a NURBS cylinder with the default options, except set Height to 8, Number of Sections to 16, and number of Spans to 32.



#### To create the skeleton for the cylinder

Build a skeleton for the cylinder. Have the skeleton consist of a single joint chain with about seven joints.



Skeleton consisting of one joint chain (joint1 through joint7). Joint4 starts approximately in the center of the cylinder.

#### To bind by rigid skinning

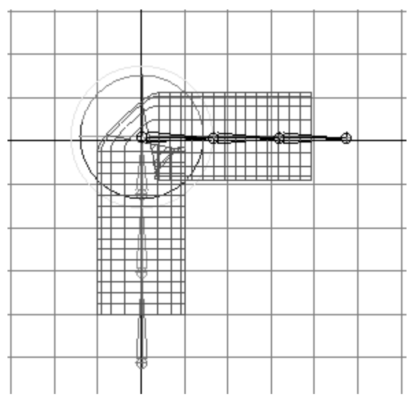
1 Select skeleton's root joint (default name: joint1).
2 Select Skin > Bind Skin > Rigid Bind.

Maya binds the cylinder to the skeleton by rigid skinning, using the default bind skin options. The cylinder is now a rigid skin object. For more information on binding rigid skin, see ”Binding rigid skin” on page 164.

Now you can exercise the skeleton and get immediate deformation effects appropriate for the character.

#### To exercise skeleton

Select the joint approximately at the center of the cylinder (for instance, joint4), and rotate it about 90 degrees.



Note that rigid skinning provides a sharp deformation effect around the rotated joint. You can adjust the deformation effect with the Paint Weights Tool.

#### To paint creasing effects

**1** Select smooth shaded display mode (hotkey: press 5).

**2** Select the cylinder.

**3** Select Deform > Paint Cluster Weights Tool > ❐.

See ”To paint weights on a rigid bound skin” on page 193.

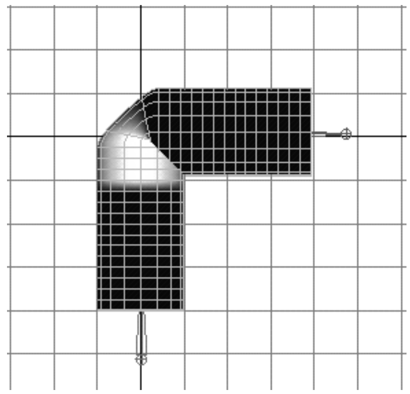
**4** In the Tool Settings window, the paintWeights box should be displayed.

**5** Note the cluster box.

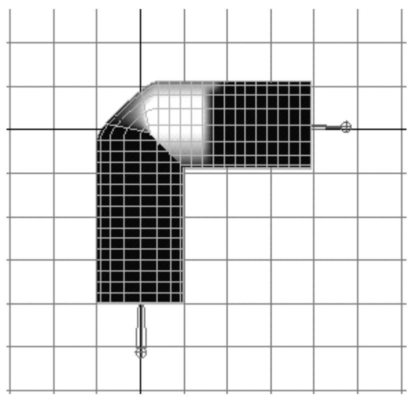
The cluster box lists the names all the rigid skin point sets (default names: joint*n*Set*n*).

**6** Click on a rigid skin point set. For example, joint3Set1.

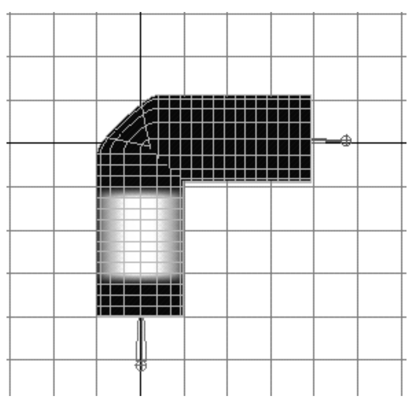
In the scene, the shading indicates the weighting of each point in the set.



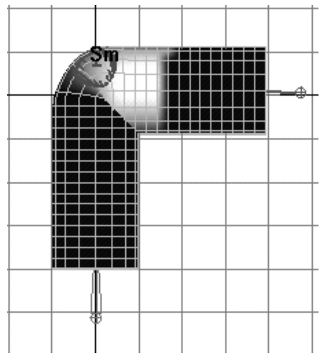
7 In the Clusters box, click on another rigid skin point set. For example, click joint4Set1.



8 Check the other rigid skin point sets. For example, check joint2Set1.



9 Use the Paint Cluster Weights Tool's brush to smooth the deformation effect.



The brush provides an intuitive way to change how the influence of the joints. Use the brush's Add operation to increase the influence of nearby joints, and use the Scale operation to decrease the influence of further joints. Use the Smooth operation to smooth out the influences of the joints.

Experiment with using the brush. With just a little experimentation, you can quickly become proficient at painting the skin point weights of joints. For more information on using paint tools, see the Paint Effects, Artisan, and 3D Paint guide.

To further smooth and deform rigid skinning, you can use flexors. For more information, see "Create all types of flexors" on page 196.

## Delete all non-skin history

## Delete all non-skin history

You can delete the history from your geometry and still retain all its smooth bind skin history.

Use the Delete Non-Skin History option to clean-up a skin's history after you have edited the UVs or topology of your skinned geometry. Deleting all non-skin history improves interactive performance in Maya and simplifies the history node list for export to game engines.

#### To delete all non-skin history from your geometry

**1** Select *all* the geometry that is bound to your character's skeleton.

| Warning | Delete Non-Skin History looks at *all* your skeleton's input connections. If all the geometry that is bound to the skeleton *is not* selected when you select Skin > Edit Smooth Skin > Delete Non-Skin History, then the Delete Non-Skin History *will not* evaluate properly. |
|---|---|

**2** If your character is not already at its bind pose, select Skin > Go to Bind Pose.

Delete Non-Skin History should always be performed at the character's bind pose, otherwise an error may occur.

**3** Select Skin > Edit Smooth Skin > Delete Non-Skin History.

All the geometry's history is deleted except for its smooth bind skin history.

**What went wrong?**

# Character Setup

## Troubleshooting Skinning

### Skinning error messages

#### Warning: Delete Non-Skin History: Skeleton is not in the bind pose and results may be undesirable

##### Cause

You tried to delete all the non-skin history from your character's geometry when it was *not* at its bind pose.

##### Solution

Perform Delete Non-Skin History with your character's geometry at its bind pose.

**1** Select your character.

**2** Select Skin > Go to Bind Pose.

**3** Select Skin > Edit Smooth Skin > Delete Non-Skin History.

# Reference Menus

## Animation menu set

### Skin >

## Skin > Bind Skin > Smooth Bind

### Skin > Bind Skin > Smooth Bind > ❐

Bind to

Specifies whether to bind to an entire skeleton or only to selected joints. Selections include Complete Skeleton or Selected Joints.

| | |
|---|---|
| Complete Skeleton | Specifies that the selected deformable objects will be bound to the entire skeleton, from the root joint on down through the skeleton's hierarchy, even if you have selected some joint other than the root joint. Binding by complete skeleton is the usual way to bind a character's skin. |
| Selected Joints | Specifies that the selected deformable objects will be bound to only the selected joints, not the entire skeleton. |

Select either Complete Skeleton or Selected Joints. Default is Complete Skeleton.

Bind Method

Specifies whether joints will influence nearby skin points based on the skeleton's hierarchy, or only on joint proximity to skin points. Selections include Closest Joint or Closest Distance.

| | |
|---|---|
| Closest Joint | Specifies that joint influence is based on the skeleton's hierarchy. In character setup, you will usually want to use this binding method because it can prevent inappropriate joint influences. For example, this method can prevent a right thigh joint from influencing nearby skin points on the left thigh. |
| Closest Distance | Specifies that joint influence is based only on proximity to the skin points. When binding skin, Maya ignores the hierarchy of the skeleton. In character setup, you will usually want to avoid this binding method because it can cause inappropriate |

joint influences. For example, this method can cause a right thigh joint to influence nearby skin points on the left thigh.

Select either Closest Joint or Closest Distance. Default is Closest Joint.

With either Closest Joint or Closest Distance, you can limit the number of joints that influence nearby skin points by setting Max Influences. You can also limit the joints' range of influence by specifying Dropoff Rate.

#### Max Influences

Specifies the number of joints that can influence each skin point. Default is 5, which is a good choice for most characters. (You can also limit the range of joint influence by specifying the Dropoff Rate.)

#### Dropoff Rate

Specifies how rapidly the influence of each joint on skin points will decrease with the distance from each joint (and the joint's bone). The greater the Dropoff Rate, the more rapid the decrease in influence with distance. The lower the Dropoff Rate, the further the influence of each joint. When you bind skin, the Dropoff Rate applies to all the selected joints. Use the slider to specify values between 0.1 and 10. You can enter values up to 100, but values between 0.1 and 10 are ideal for most situations. Default is 4, which provides good deformation effects for most characters.

After binding skin, you can use the Paint Skin Weights Tool to edit the influence of joints in an intuitive manner. For more information, see "Painting smooth skin point weights" on page 151.

## Skin > Bind Skin > Rigid Bind

### Skin > Bind Skin > Rigid Bind > ❐

#### Bind to

Specifies whether to bind to an entire skeleton or only to selected joints. Selections include Complete Skeleton or Selected Joints.

**Complete Skeleton** Specifies that the selected deformable objects will be bound to the entire skeleton, from the root joint on down through the skeleton's hierarchy, even if you have selected some joint other than the root joint. Binding by complete skeleton is the usual way to bind a character's skin.

| | |
|---|---|
| Selected Joints | Specifies that the selected deformable objects will be bound to only the selected joints, not the entire skeleton. |
| Force All | Specifies that the selected deformable object will be bound to all selected joints, including those with no influence. |

### Coloring

Specifies whether to color the joints according to the colors automatically assigned to skin point sets. Coloring joints can be helpful later when you are editing skin point set memberships. Click Color Joints on or off. Default is off.

### Bind Method

Specifies whether you want to bind by closest point or by partition set.

| | |
|---|---|
| Closest Point | Specifies that Maya automatically organize deformable object points into skin point sets for you based on the proximity of each point to a joint. For each joint with a bone, a skin point set will be created that includes the points that are closest to the given point. The points are then identified as skin points, with each skin point being a member of only one skin point set. Maya places the skin point sets in a partition, which assures that each point can only be in one set. Finally, each set will be bound to the nearest joint. |
| Partition Set | Specifies that Maya bind points that you've already organized into sets in a partition. You should have as many sets as you have joints. Each set will be bound to the nearest joint. |

### Partition

If you select Partition Set, select the name of the partition you wish to bind. Select only partitions containing sets of deformable points.

Click Closest Point or Partition Set. Default is Closest Point.

If you select Partition Set, a list of the currently available partitions is listed. Select the partition you want the rigid skin point sets to be in.

# Skin > Detach Skin

## Skin > Detach Skin > ❐

### History

Set to Delete History, Keep History, or Bake History.

| | |
|---|---|
| Delete History | Detaches the skin, move it to its original, undeformed shape, and delete the skin's skin cluster nodes. Select this option if you want to bind the skin starting anew, for example, because your extensive editing of smooth skin weights gave undesirable results. |
| Keep History | Detaches the skin and move it to its original, undeformed shape. It will not delete the skin's skin cluster nodes. This is the default option. Use this option to preserve smooth skin weights when you bind skin again. This is useful, for instance, if you decide to add an extra joint to a skeleton but want to retain the existing smooth skin weights after you detach the smooth skin and bind the skin again. |
| Bake History | Detaches the skin and delete its skin cluster nodes, but will not move the skin to its original, undeformed shape. The skin will maintain its current shape after detachment. This is useful, for instance, if you won't deform the skin's shape anymore and want to lighten the processing demands of your scene. (You might use the skin, for example, as a stationary character in the background of the scene.) |

### Coloring

(This option only applies to rigid skinning.)

Specifies whether to remove the joint colors assigned during binding. Click on or off. Default is on.

# Skin > Go To Bind Pose

Returns the skeleton to the position where it's bind pose was set. See "Bind pose" on page 141.

# Skin > Edit Smooth Skin > Add Influence

## Skin > Edit Smooth Skin > Add Influence > ❐

### Geometry

Click Use Geometry on if you want the influence object’s shape as well as its transform attributes (translation, rotation, and scale) to influence the skin’s shape. Click Use Geometry off if you only want the influence object’s transform attributes (translate, rotate, and scale) to influence the skin’s shape. Default is Use Geometry on.

### Dropoff

Specifies the rate at which the influence of the influence object’s position drops as the distance from the influence object increases. Specify values between 0.1 and 100. Default is 4.0.

### Polygon Smoothness

Specifies how accurately the smooth skin points follow a given polygonal influence object. The higher the value, the more rounded the deformation effect. Set values between 0.0 and 50.0. Default is 0.0.

### NURBS Samples

Specifies the number of samples used to evaluate the influence of a NURBS influence object’s shape. The greater the number of samples, the more closely the smooth skin follows the influence object’s shape. Set values between 1 and 100. Default is 10.

### Weight Locking

Specifies that you want to prevent the influence object’s weights from being changed indirectly, typically because of weight normalization during weight painting and editing (see ”Holding smooth skin weights” on page 153). Instead, Maya holds the weights to the Default Weight. Default is off.

### Default Weight

Specifies the default holding weight if Weight Holding is on. Default is 0.000.

# Skin > Edit Smooth Skin > Remove Influence

Removes the current influence object’s influence from the skin. Remove Influence does not delete the current influence object from the scene.

## Skin > Edit Smooth Skin > Set Max Influences

Specifies the number of joints and influence objects that can influence each skin point. Use slider to select values from 0 to 30. Default is 5.

## Skin > Edit Smooth Skin > Paint Skin Weights Tool

Lets you paint a weight intensity value on the current skin. See:

- "Painting smooth skin point weights" on page 151
- "Paint skin weights" on page 171
- "Paint skin weights for creases" on page 172

See also *"How Artisan brush tools work"* in the Paint Effects, Artisan, and 3D Paint guide.

### Skin > Edit Smooth Skin > Paint Skin Weights Tool > ❐

Lets you specify the settings for the Paint Skin Weights Tool in the Tool Settings editor. Both the Influence and Paint Weights sections are unique to the Paint Skin Weights Tool. These unique attributes are described below. For descriptions of all other attributes in all other sections, see *"Common Artisan Brush Tool Settings"* in the Paint Effects, Artisan, and 3D Paint guide.

#### Influence section

These are descriptions of the attributes in the Influence section.

##### Sort Transforms

Sorts the joints that influence skin weights for the current character.

| | |
|---|---|
| Alphabetically | Sorts the joint names alphabetically. |
| By Hierarchy | Sorts the joint names by hierarchy (parent-child). |

##### Toggle Hold Weights On Selected

Locks the weight of the current influence so that when the weights of other influences are painted, the influence that is *held* is not affected.

#### Paint Weights section

These are descriptions of the attributes in the Paint Weights section.

##### Paint Operation

| | |
|---|---|
| Replace | The brush stroke replaces the skin weight with the weight set for the brush. |

| | |
|---|---|
| Add | The brush stroke increases the influence of nearby joints. |
| Scale | The brush stroke decreases the influence of far away joints. |
| Smooth | The brush stroke smooths out the influences of the joints. |

### Value

The specified weight value the brush stroke applies.

### Min/Max Value

Sets the minimum and maximum possible paint values. By default, you can paint values between 0 and 1. By setting the Min/Max Values you can extend or narrow the range of weight values.

Negative values are useful for subtracting weight. For example, if you set Min Value to -1, Value to -0.5, and select Add for the operation, you would then subtract 0.5 from the weight of your skin when you paint. Positive values are used as multipliers.

### Clamp

Sets whether you want to clamp the values within a specified range, regardless of the Value set when you paint.

| | |
|---|---|
| Lower | Turn this on to clamp the lower value to the Clamp Value specified below. For example, if you clamp Lower and set the lower Clamp Value to 0.5, the values you paint will never be less than 0.5, even if you set the Value to 0.25. |
| Upper | Turn this on to clamp the upper value to the Clamp Value specified below. For example, if you clamp Upper, set the upper Clamp Value to 0.75, and set Value to 1, the values you paint will never be greater than 0.75 |

### Clamp Values

Turns on the Lower and Upper clamping fields.

### Flood

Click Flood to apply the brush settings to all the weights on the selected skin. The result depends on the brush settings defined when you perform the flood.

# Skin > Edit Smooth Skin > Export Skin Weight Maps

## Skin > Edit Smooth Skin > Export Skin Weight Maps > ❐

### Export Value

### Alpha

Exports alpha channel (opacity) values.

### Luminance

Exports luminance (brightness) values.

### Map Size X, Y

Sets the width and height of the image. If the skin has 1000s of CVs or vertices, a large map will ensure the destination skin matches the original skin. However, large maps take more time to save to disk and also use more disk space. A map that is too small might cause inadvertent averaging of the alpha or luminance values.

The default values 512 by 512 work well for most skins. If the skin lots of CVs or vertices, use Map Size values of 1024 by 1024.

If the skin has relatively few CVs or vertices, use values 256 by 256.

If you use a skin made from dozens of NURBS patches, consider using 256 by 256 to avoid wasting lots of disk space.

### Keep Aspect Ratio

Maintains the height to width ratio of the attribute map when you export.

### Image Format

Specifies the type of image, for instance, TIFF, JPEG, and so on.

# Skin > Edit Smooth Skin > Import Skin Weight Maps

Lets you import skin weight maps to your scene. See ”Importing to a character in another scene” on page 154, ”Importing back to the original character” on page 155, and ”Import weights” on page 190.

Launches the Maya Import file browser. For more information, see the *Basics* guide.

# Skin > Edit Smooth Skin > Mirror Skin Weights

## Skin > Edit Smooth Skin > Mirror Skin Weights > ❐

### Mirror Across

XY specifies mirroring weights about the global XY plane (the default).

YZ specifies mirroring weights about the global YZ plane.

XZ specifies mirroring weights about the global XZ plane.

### Direction

Positive to Negative (+Z to -Z) specifies direction of the mirroring along the specified Mirror Across plane.

# Skin > Edit Smooth Skin > Copy Skin Weights

Copies the skin weights of the selected joints to the virtual clipboard. If you selected an entire skeleton, then all the skin weights of the joints in the skeleton are copied. See "Copying smooth skin weights" on page 152 and "Copy smooth skin weights" on page 187.

Once you copy the skin weights, you can paste the copied skin weights to another skeleton in your scene.

# Skin > Edit Smooth Skin > Reset Weights to Default

Resets the weights of the selected joints to their initial, default values. See "Reset skin point weights to default weights" on page 188.

# Skin > Edit Smooth Skin > Prune Small Weights

Prunes the small weight of the current joints. Pruning the small weights speeds up processing. See "Prune insignificant smooth skin weights" on page 189.

## Skin > Edit Smooth Skin > Prune Small Weights > ❐

### Prune Below

Specifies the weight below which all smaller weights are reset to 0. By default, the remaining skin weights are normalized to add up to 1.

## Skin > Edit Smooth Skin > Remove Unused Influences

Disconnects all joints and influence objects with skin weights of 0 from their smooth skin. See "Remove unused influences from a smooth skinned surface" on page 189.

## Skin > Edit Smooth Skin > Disable Weight Normalization

Turns off the automatic normalization of smooth skin weights for the current skeleton. See "Controlling smooth skin weight normalization" on page 154.

## Skin > Edit Smooth Skin > Enable Weight Normalization

Turns on the automatic normalization of smooth skin weights for the current skeleton. The process of scaling weights so that they all add up to one is called "normalization." See "Controlling smooth skin weight normalization" on page 154.

## Skin > Edit Smooth Skin > Normalize Weights

Adjusts the weighting of the selected skeleton so that all its smooth skin weights add up to one. See "Controlling smooth skin weight normalization" on page 154 and "Normalize weights" on page 188.

## Skin > Edit Smooth Skin > Delete Non-Skin History

Deletes all the non-skin history from the selected character's geometry. See "Delete all non-skin history" on page 209.

Use this menu item to clean-up a skin's history after you have edited the UVs or topology of your skinned geometry. Deleting all non-skin history improves interactive performance in Maya and simplifies the history node list for export to game engines.

Delete Non-Skin History looks at *all* your skeleton's input connections. If all the geometry that is bound to the skeleton *is not* selected when you select Skin > Edit Smooth Bind > Delete Non-Skin History, then the Delete Non-Skin History operation will not evaluate properly.

# Skin > Edit Rigid Skin > Create Flexor

### Flexor Type

Specifies whether to create lattice flexors, sculpt flexors, or joint cluster flexors. Select lattice, sculpt, or joint cluster. Default is lattice.

### Joints

Specifies whether to create joint lattice, joint sculpt, or joint cluster flexors at selected joints only, or at all a skeleton's joints. Click At Selected Joint(s) or At All Joint(s). Default is At Selected Joint(s).

### Bones

Specifies whether to create bone lattice or bone sculpt flexors at selected bones only, or at all bones. Click At Selected Bone(s) or At All Bone(s) to create. Default specifies no bone flexors will be created. This option does not apply to joint cluster flexors.

### Lattice Options

If Flexor Type is lattice, specify the Lattice Options:

| | |
|---|---|
| S, T, U Divisions | Specifies the structure of the lattice in the lattice's local STU space. (STU space provides a special coordinate system for specifying the structure of lattices.)<br><br>You can specify the lattice's structure in terms of S, T, and U divisions. When you specify the divisions, you also specify the number of lattice points in the lattice, because the lattice points are located where the divisions meet on the lattice's exterior. The greater the number of divisions, the greater the lattice point resolution. Though your control over the deformation increases with the number of lattice points, the performance may be affected.<br><br>The default settings are S has 2 divisions, T has 5 divisions, and U has 2 divisions, which provides 20 lattice points. You can quickly change the settings by using the sliders to select values from 2 to 20. |
| Position the Flexor | Specifies that you want to move, rotate, or scale the lattice now, before you create the lattice flexor. This enables you to adjust the lattice before it starts having an effect on skin objects. Click Position the Flexor, and then use the Move Tool, Rotate Tool, or Scale Tool to adjust the lattice now. |

### Sculpt Options

If Flexor Type is sculpt, specify the Sculpt Options:

| | |
|---|---|
| Max Displacement | Specifies the distance that the sculpt sphere can push a skin object's points from the sculpt sphere's surface.Use slider to select values from 0.000 to 2.000. Default is 0.000. |
| Dropoff Distance | Specifies the sculpt sphere's range of influence. (How the range of influence can decline is specified by Dropoff Type.) Use slider to select values from 0.000 to 20.000. Default is 0.000. |
| Dropoff Type | Specifies how the sculpt sphere's range of influence declines or drops off. (The range of influence is specified with the Dropoff Distance.) There are two Dropoff Types: None and Linear. Default is None. |
| Mode | Specifies the sculpt sphere's deformation mode as flip, project, or stretch. Select Flip, Project, or Stretch. Default is Stretch. |
| Inside Mode | Specifies how the sculpt sphere influences the skin points located inside the sculpt sphere. There are two modes: Ring and Even.<br><br>**Ring** mode pushes inside points outside of the sculpt sphere, creating a contoured, ring-like effect around the sculpt sphere.<br><br>**Even** mode spreads the inside points all around the sculpt sphere evenly, creating a smooth, spherical effect.<br><br>Select Ring or Even. Default is Ring. |

### Cluster Options

(No options for joint cluster flexors.)

## Skin > Edit Rigid Skin > Copy Flexor

Copies the current flexor to the virtual clipboard. See ”Flexors” on page 163,”Copy joint lattice flexors” on page 202, and ”Copy bone lattice flexors” on page 203.

## Skin > Edit Rigid Skin > Reassign Bone Lattice Joint

Sets a new driver joint for the current bone lattice flexor. See "Reassign bone lattice flexor joints" on page 203.

By default, the joint that drives a bone lattice flexor is the child of the bone that the flexor's influence lattice surrounds. This makes sense for situations such as creating arm bicep muscle effects, where the rotation of the elbow joint drives a bulge deformation around the shoulder joint's bone. In fact, it makes sense for most of the muscle deformation effects you might want for conventional human characters.

However, you can have the action of the bone lattice flexor be driven by any other joint you wish. The most common situation might be where you want the rotation of the joint whose bone the influence lattice surrounds drive the bone lattice flexor. For instance, for some reason you might want to have the rotation of the shoulder joint drive the bulge around the shoulder joint's bone. Keep in mind that you can have any joint drive the bone lattice flexor. For instance, you could have the rotation of a finger joint drive a bone lattice flexor around a character's head.

## Skin > Edit Rigid Skin > Preserve Skin Groups > Detach Skeleton

Disconnects the current skeleton from its skin. Detaching the skeleton preserves the rigid skin point sets and the rigid skin point weights, and unlocks the transformation attributes of its skin objects. See "Detach and reattach to skeletons" on page 200.

## Skin > Edit Rigid Skin > Preserve Skin Groups > Detach Selected Joints

Disconnects the current joints from their skin. Detaching joints preserves the rigid skin point sets and the rigid skin point weights, and unlocks the transformation attributes of its skin objects. See "Detach and reattach to selected joints" on page 201.

## Skin > Edit Rigid Skin > Preserve Skin Groups > Reattach Skeleton

Reconnects the current skeleton to its skin objects. Reattaching the skeleton locks the transformation attributes of its skin objects.

## Skin > Edit Rigid Skin > Preserve Skin Groups > Reattach Selected Joints

Reconnects the current joints with their skin objects. Reattaching joints locks the transformation attributes of their skin objects.

# Nodes

## Character nodes

## General skin node attributes

### Deformer Attributes

#### Envelope

Specifies the deformation scale factor. Values can vary from 0 to 1. Default is 1.

### Node behavior

#### Caching

Specifies that Maya store the results of input evaluations, and then provide those results to the node. This saves Maya from having to re-evaluate the input nodes every time the node needs the results. If there are no changes to the input nodes, then this setting can improve display performance with no loss of results. However, note that caching uses more memory than would otherwise be used, which could adversely affect performance. Also, if there are changes to input nodes, more memory is allocated and then freed during each deformation, which could also adversely affect display performance.

#### Node State

| | |
|---|---|
| Normal | Specifies that Maya evaluate and display the deformation. Maya will evaluate the node as usual. This is the default. |
| HasNoEffect | Specifies that Maya prevent the deformation, but display the object. Maya will evaluate the nodes in the node's history, but not the node itself. |
| Blocking | Specifies that Maya prevent the deformation, and not display the object. Maya will not report the results of any evaluations of input nodes to this node. |

| | |
|---|---|
| Waiting-Normal | (For Maya internal use only.) Specifies that if the dependency graph evaluation refresh performance setting (Window > Settings/ Preferences > Performance Settings) is set to Demand or Release, the node will take the Normal state when you click Update or release the mouse button. |
| Waiting-HasNoEffect | (For Maya internal use only.) Specifies that if the dependency graph evaluation refresh performance setting is set to Demand or Release, the node will take the HasNoEffect state when you click Update or release the mouse button. |
| Waiting-Blocking | (For Maya internal use only.) Specifies that if the dependency graph evaluation refresh performance setting is set to Demand or Release, the node will take the Blocking state when you click Update or release the mouse button. |

## Smooth skin nodes

# skinCluster

## Smooth Skin Attributes

### Use Components

Specifies whether changes to the components of smooth skin influence objects can change their deformation effects on smooth skin objects. If Use Components is off (the default), changes to components won't change the deformation effect. If Use Components is on, changes to components can change the deformation effect.

For example, if your smooth skin influence objects are NURBS surfaces and Use Components is on, moving the influence objects' CVs can change the deformation effect.

Alternatively, if your smooth skin influence objects are polygonal surfaces (meshes), setting Use Infl Components on makes it possible for changes to individual polygons to in turn deform skin. Otherwise, with Use Infl Components off, the entire shape of the influence object influences the skin but changes to individual polygons can not influence the skin.

Default is off.

#### Use Component Matrix

Specifies that you can use a component-based influence object on a character's skin that changes scale. Default is on.

#### Normalize Weights

Specifies whether the weights are normalized automatically. For more information about normalization, see "Controlling smooth skin weight normalization" on page 154. Default is on.

## tweak

Stores transformation information for the vertices of the current skin. Tweak also ensures the proper transformation of skin vertices when you manipulate a character's skeleton. When you delete the tweak node, all the transformation information for the vertices that are affected by the influence joints and object is erased. Deleting the tweak node is an easy way of removing deformations from a skin.

# Rigid skin nodes

## jointCluster

### Joint Cluster Attributes

#### Upper Bound

Specifies percent of the length of parent bone along which the cluster weights decrease. The cluster weights decrease from the Upper Value as they approach the joint according to the Upper Dropoff Type. Default is 10.

#### Upper Value

Specifies the initial cluster weight value at the Upper Bound location. Default is 0.5.

#### Upper Dropoff Type

Specifies how the cluster weights along the parent bone decrease as they approach the joint. Select linear (0), sine (1), exponential (2), or none (3). Default is linear.

#### Lower Bound

Specifies the percent of the length of the joint's bone along which the cluster weights decrease. The cluster weights decrease from the Lower Value as they approach the joint according to the Lower Dropoff Type. Default is 10.

### Lower Value

Specifies the initial cluster weight value at the Lower Bound location. Default is 1.

### Lower Dropoff Type

Specifies how the cluster weights along the joint's bone decrease as they approach the joint. Select linear (0), sine (1), exponential (2), or none (3). Default is linear.

### Relative

Specifies that the deformation take place only when the parent of the joint cluster handle is moved, rotated, or scaled.

### Partial Resolution

Specifies whether Maya provides the complete deformation, or only an approximation of the deformation. Selections include full and partial. Full specifies the complete deformation. Partial specifies an approximation of the deformation, which can improve Maya's display performance. With partial, Maya rounds down the cluster weights based on the Percent Resolution. Default is full.

### Percent Resolution

Specifies the increment percentage by which the joint cluster deformation resolution is rounded down. Maya uses the increment percentage to round off the skin point cluster weights to the next lowest increment. For example, with a Percent Resolution of 5.00, a skin point's joint cluster weight of .94 would be rounded down to .90. A joint cluster weight of .46 would be rounded down to .45. Default is 5.00. (Available only if Partial Resolution is set to partial.)

### Angle Interpolation

Specifies the interpolation direction. Use this attribute to correct undesirable discontinuities in the deformation effect when you change joint angles or weight percentages even by a small amount. The discontinuities occur when the joint cluster node is using an inappropriate interpolation direction to guide the deformation effect. To change the interpolation direction, you can set Angle Interpolation to closest, positive, or negative. By default, Angle Interpolation is closest, which provides the usual rigid skinning deformation effects. The default setting is fine for most situations, but when you encounter discontinuities you can adjust the deformation effect by selecting a positive or negative interpolation.

# jointFlexor

## Joint Lattice Flexor

### Creasing

Specifies the bulging of skin points on the inside of the joint rotation. As you change Creasing, the lattice points on the inside of the joint rotation move inward or outward to change the shape of the bulge. Positive values cause the skin to bulge inward. Negative values cause the skin to tuck inward. Default is 0.

### Rounding

Specifies the bulging of skin points on the outside of the joint rotation. As you change Rounding, the lattice points on the outside of the joint rotation move inward or outward to change the shape of the bulge. Positive values cause the skin to bulge outward. Negative values cause the skin to tuck inward. Default is 0.

### Length In

Specifies the positions of the lattice divisions that are near the parent bone above the joint. As you change Length In, the lattice divisions along the parent bone move away from or towards the joint. This changes the extent of the effects of Creasing, Rounding, Width Left, and Width Right. Positive values spread the bulging effects up along the parent bone by moving lattice divisions away from the joint. Negative values compress the bulging effects closer to the joint by moving the lattice divisions closer to the joint. Default is 0.

### Length Out

Specifies the positions of the lattice divisions that are near the joint's child bone. As you change Length Out, the lattice divisions along the joint's child bone move away from or towards the joint. This changes the extent of the effects of Creasing, Rounding, Width Left, and Width Right. Positive values spread the bulging effects down along the joint's child bone by moving lattice divisions away from the joint. Negative values compress the bulging effects closer to the joint by moving the lattice divisions closer to the joint. Default is 0.

### Width Left

Specifies the bulging of skin points on the left side of the joint rotation. As you change Width Left, the lattice points on the left side of the joint rotation move outward or inward to change the shape of the bulge. Positive values cause the skin to bulge outward. Negative values cause the skin to bulge inward. Default is 0.

#### Width Right

Specifies the bulging of skin points on the right side of the joint rotation. As you change Width Right, the lattice points on the right side of the joint rotation move outward or inward to change the shape of the bulge. Positive values cause the skin to bulge outward. Negative values cause the skin to bulge inward. Default is 0.

## boneFlexor

### Bone Lattice Flexor

#### Length In

Specifies the positions of the lattice divisions that are near the bone's joint, above the center of the bone. As you change Length In, the lattice divisions near the bone's joint move away from or towards the bone's center. This changes the extent of the effects of Bicep, Tricep, Width Left, and Width Right. Positive values spread the bulging effects up along the bone by moving lattice divisions away from the bone's center. Negative values compress the bulging effects by moving the lattice divisions closer to the bone's center. Default is 0. Note that Length In does not affect the position of the furthest lattice division.

#### Length Out

Specifies the positions of the lattice divisions that are near the bone's child joint, below the center of the bone. As you change Length Out, the lattice divisions along the bone move away from or towards the center of the bone. This changes the extent of the effects of Bicep, Tricep, Width Left, and Width Right. Positive values spread the bulging effects down along the bone by moving lattice divisions away from the bone's center. Negative values compress the bulging effects by moving the lattice divisions closer to the bone's center. Default is 0. Note that Length Out does not affect the position of the furthest lattice division.

#### Width Left

Specifies the bulging of skin points on the left side of the child joint's rotation. As you change Width Left, the lattice points on the left side of the joint rotation move outward or inward to change the shape of the bulge. Positive values cause the skin to bulge outward. Negative values cause the skin to bulge inward. Default is 0.

### Width Right

Specifies the bulging of skin points on the right side of the child joint's joint rotation. As you change Width Right, the lattice points on the right side of the joint rotation move outward or inward to change the shape of the bulge. Positive values cause the skin to bulge outward. Negative values cause the skin to bulge inward. Default is 0.

### Bicep

Specifies the bulging of skin points on the inside of the child joint's rotation. As you change Bicep, the lattice points on the inside of the joint rotation move inward or outward to change the shape of the bulge. Positive values cause the skin to bulge inward. Negative values cause the skin to tuck inward. Default is 0.

### Tricep

Specifies the bulging of skin points on the outside of the child joint's rotation. As you change Tricep, the lattice points on the outside of the joint rotation move inward or outward to change the shape of the bulge. Positive values cause the skin to bulge outward. Negative values cause the skin to tuck inward. Default is 0.

## jointFtd

See "ffd" on page 408.

## ffd

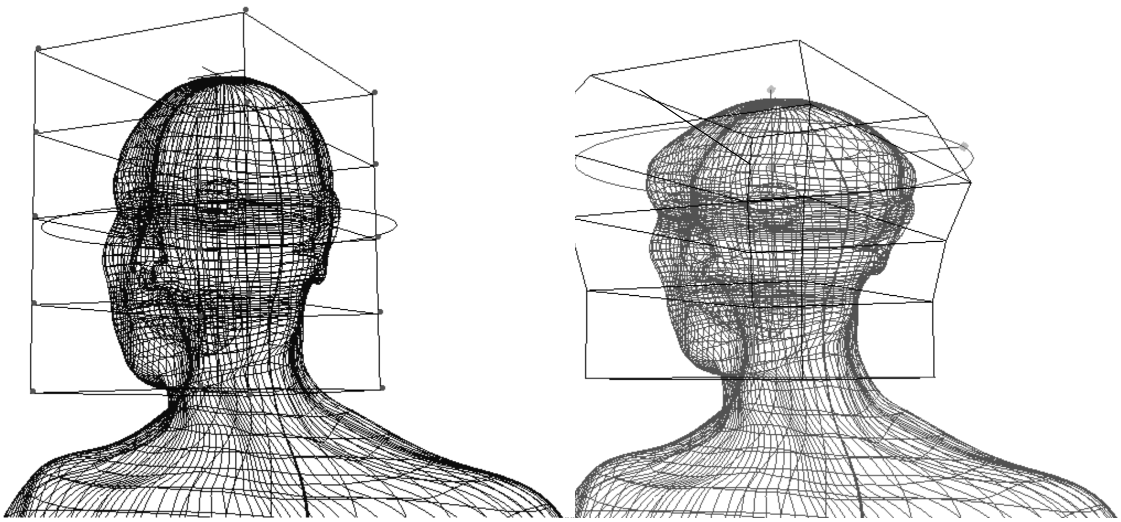
See "ffd" on page 408.

# 4 Deformers

## About

## Deformers overview

With Maya's deformers, you can change the shape of objects.



Lattice and squash deformers acting on the head

| | |
|---|---|
| Note | Other software packages use the terms *modifiers* and *space warp* to refer to what Maya calls deformers. |

### Setting deformers

## Deformable objects, points, and sets

A deformer can create deformation effects on any deformable object. A deformable object is any object whose structure is defined by control points. Control points include NURBS control vertices (CVs), polygonal vertices, and lattice points. NURBS curves, NURBS surfaces, polygonal surfaces (meshes), and lattices are all deformable objects. In this documentation, control points are often called points, and the control points of deformable objects are often called deformable object points.

Maya Unlimited's subdivision surfaces are deformable. For more information, see the *Subdivision Surface Modeling* guide.

With Maya's API development tools, you can define your own, custom deformable objects. For more information, see the *API* guide.

A character's model can consist of one deformable object (for example, a large polygonal surface) or groups of deformable objects (for example, groups of NURBS surfaces).

When you create a deformer, Maya puts all the deformable object points that the deformer can affect into a set, called a deformer set. This set is editable. For more information, see "Editing deformer set membership" on page 234.

# Nodes, history, and the deformation order

One way to think about a scene in Maya is that it is a web of nodes. Each node consists of specific information and actions associated with that information. Each node can receive, hold, and provide information by means of attributes. A node's attributes can connect to the attributes of other nodes, thus forming the web of nodes. As you use Maya's interface, Maya creates, connects, evaluates, and destroys nodes. At any moment, what you see in the workspace is the result of how Maya is continuously evaluating the web of nodes that underlies and comprises your work. In short, underlying everything you do in Maya lies Maya's dynamic, node-based architecture.

## Dependency graph

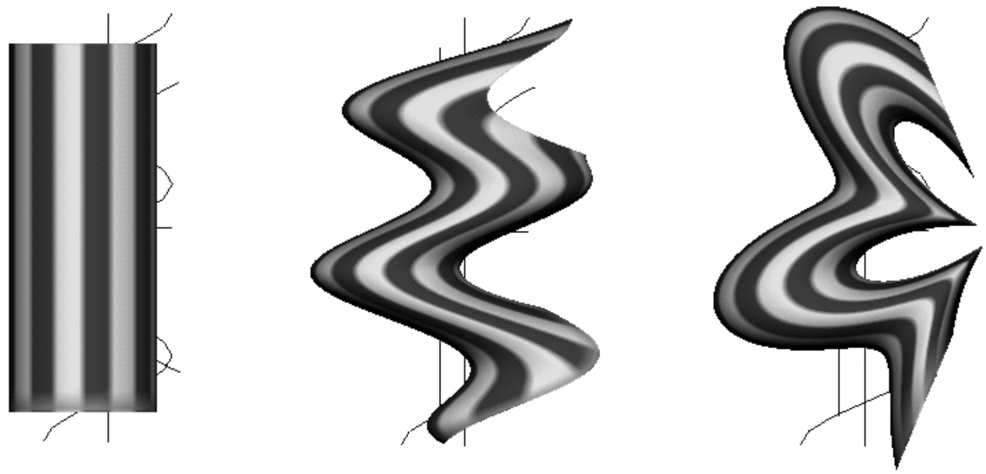
Maya's dependency graph provides a representation of the relationships between connected nodes. To view the dependency graph, you can use the Hypergraph. For more information, see the *Basics* guide.

For any particular node, the dependency graph shows the node's history. The node's history includes all the nodes that are connected to it, or are connected to nodes that are connected to it, and so on. For discussing a node's history, the terms input and output connections can be useful. Input nodes are nodes that can be evaluated *before* the node itself is evaluated, and Output nodes are nodes that can be evaluated only *after* the node itself is evaluated. Note that, from Maya's perspective, a node's history includes its future as well as its past.

## Deformation order

It's important to keep a node's history in mind when using deformers. The deformation effect provided by a particular deformer can depend on where Maya places the deformer in the node's history. The reason is that the deformation effect can vary depending on the order in which Maya evaluates the deformations. The order in which Maya evaluates deformations is called the *deformation order*.

In general, you can apply as many deformers to an object as you like. Since the effects depend on the order in which the deformers deform the object, you can create a great variety of effects. For example, if you create a bend deformer and then create a sine deformer, the result is different than if you first created the sine deformer and then created the bend deformer.



Undeformed cylinder Bend followed by sine Sine followed by bend

By default, the order in which deformers act on a deformable object's original shape is the order in which the deformers were created. The deformers created first act on the original shape first, and the deformers created last act on the original shape last.

## Deformation chain

Consider the history of a NURBS sphere being deformed by one or more deformers. In the dependency graph, the original (undeformed) shape node (nurbsSphereShape*n*Orig) would follow immediately after the make node (makeNurbSphere*n*).

> **Note** Maya sometimes refers to an object displayed as the original shape node as an *intermediate object*.

Maya places a tweak node (tweak*n*) after the original shape node (for more information on tweak nodes, see "Point tweaking objects" on page 235). After the tweak node, Maya places the deformer nodes that carry out deformations, typically in the order that the deformers were created. The order in which Maya places the deformer nodes (outputs) from the original shape node determines the deformation order. However, you can control deformer placement (see ”Deformation order” on page 234) when you create a deformer, and change the deformation order (see ”Changing an object’s deformation order” on page 236) after creation. Finally, after the deformer nodes, Maya places a shape node that provides the final (deformed) shape of the sphere (nurbsSphereShape*n*).

The sphere’s history determines the deformation order. Since Maya evaluates the sphere starting from the make node and working all the way through in order to the final (deformed) shape node, the node connections involved are said to provide a *deformation chain*.

## Deformation order

When you create a deformer, you can specify the deformer’s placement in the deformable object’s deformation order. See ”Advanced deformer options” on page 403. The placement can affect the deformer’s effect and performance.

> Note After you create a deformer, you can edit a deformer node’s placement by changing the deformation order. See ”To change deformation order” on page 356.

### Modifying deformers

## Editing deformer set membership

Maya provides several ways that you can edit deformer set membership. You can edit deformer set membership with the Relationship Editor. See ”Deform > Edit Membership Tool” on page 363. The Relationship Editor lists all the deformer sets in your scene, and lists all the points in each set. With the Relationship Editor, you can also provide for exclusive deformer set membership so that a point can be in only one set. You can directly edit deformer set memberships by picking deformable object points with the Edit Membership Tool. Also, you can paint deformer set memberships with the Paint Set Membership Tool. This tool provides an intuitive, easy-to-use way to edit set membership. For the cluster, sculpt, lattice, and wire deformers, you can quickly prune all points from the deformer set.

### Exclusive deformer set membership

The exclusive option helps you to create non-overlapping deformations by ensuring that the sets belonging to each deformer are mutually exclusive. See ”Advanced deformer options” on page 403.

The mutual exclusivity of the deformers is accomplished by placing the deformer's sets into a partition. The partition guarantees that the sets will continue to be mutually exclusive even if you edit the membership of the sets. To put deformer sets into a partition, use the Relationship Editor. For more information on sets, partitions, and the Relationship Editor, refer to the *Basics* guide.

## Point tweaking objects

Point tweaking is moving or setting keys on the individual points of an object. When you tweak the points of an object for which you have already created some deformers, Maya automatically prevents the unexpected effects that can occur when you use deformers. Maya does so by applying the tweaks to the object before applying any deformations to the object.

When you create deformers, Maya creates tweak nodes as well as deformation nodes. In the dependency graph. Maya places the tweak nodes as inputs to the deformation nodes so that any point tweaking is carried out before the evaluation of the deformation nodes. This placement means that, by default, an object's deformation order includes point tweaking first, and then includes deformations in the order that the deformers were created.

| | |
|---|---|
| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Be sure you are happy with the deformable object's topology before you begin using deformers. You can save a copy of the object in case you want to do further modeling in the future. |

When the deformation order includes point tweaking first (the default), CVs may not move in the same direction as the Move Tool's manipulator if the attributes of the deformers do not have their initial (reset) values. If you would like to change this, reset the deformers to have their initial (reset) values. Alternatively, you can change the deformation order so that Maya applies the point tweaking after applying deformations. However, if Maya applies point tweaking after applying deformations, you may get unexpected results when you use the deformers.

If you do some point tweaking and then want to check how the object deforms without the tweaking, you can disable the tweak node.

## Displaying and hiding intermediate objects

An intermediate object is an object's shape prior to its deformation. After you deform an object, you can still view its prior shape by displaying its intermediate object. Comparing the intermediate object with the deformed object can be a useful way to judge the effect of a deformation. See "To display intermediate deformation object(s)" on page 357 and "To hide intermediate deformation object(s)" on page 357.

## Changing an object's deformation order

When you use more than one deformer to deform an object, the final effect of the deformations can vary depending on the order in which the deformations occur. By default, the deformations occur in the order that the deformers were created for the object. The deformer created first deforms the object first, and the deformer created last deforms the object last. However, you can change or re-order the deformation order to get the effect you want.

## Changing evaluation performance

You can change dependency graph node evaluation performance so that the scene refreshes right after you drag the mouse, only when you tell the scene to refresh, or only when you release the mouse button. Changing the evaluation performance can improve scene display speed if you have many complex objects being deformed.

## Modeling with deformers

You can use deformers as modeling tools for shaping NURBS or polygonal objects. When you are finished modeling, be sure to delete the deformer along with the rest of the object's history. Note that in the context of modeling, an object's history can be called its *construction history*.

## Setup and animation with deformers

You can set keys on any of deformer's keyable attributes (or channels). Keys can be set in the Channel Box, the Timeline, the Graph Editor, the Dope Sheet, or by using the Maya Embedded Language (MEL) commands.

When setting up characters, you can create attributes that drive deformer attributes by adding new attributes, and then defining relationships between the new attributes and the deformer attributes.

Add new attributes from the Add Attribute window (Modify > Add Attribute). Define relationships between attributes with the Connection Editor (Window > General Editors > Connection Editor), the Set Driven Key window (Animate > Set Driven Key > Set > ❐), or by writing expressions in the Expression Editor (Window > Animation Editors > Expression Editor).

This book refers to channels as keyable attributes that are displayed in the Channel Box. You can use the Channel Control editor to specify that a node's animatable (potentially keyable) attributes that are not in the Channel Box by default be displayed as channels in the Channel Box. Note that it's possible to put non-animatable (and therefore non-keyable) attributes into the Channel Box, but in general all channels in the Channel Box are keyable attributes.

For more information on Maya's animation features, refer to the *Animation* guide.

## Editing node behavior to improve performance

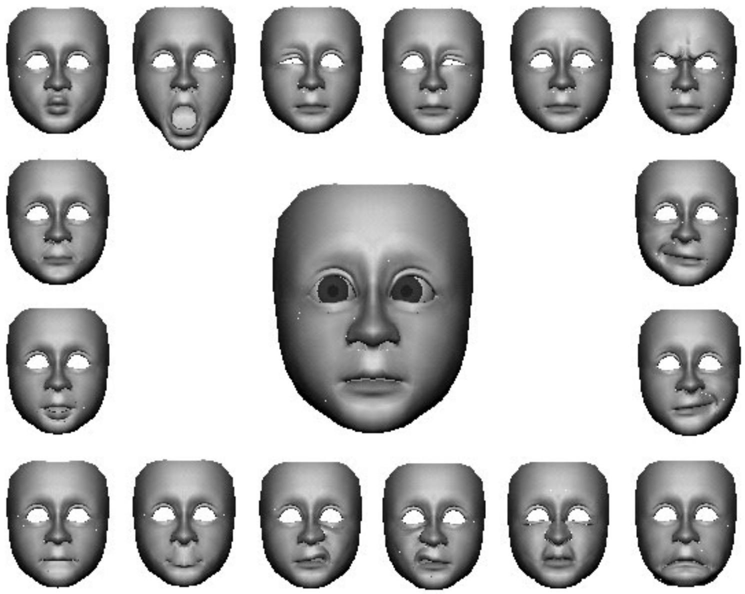
For each object in your scene, if there has been any change to its node or any of the nodes in its history (its input or output nodes), Maya will evaluate the nodes and update the display. A deformed object has more nodes in its history than an undeformed object. If you have many deformed objects in your scene, you can improve the display performance by editing the node behavior attributes of the deformed object nodes.

### Blend Shape deformers

## Blend Shape deformers

Blend shape deformers let you change the shape of one object into the shapes of other objects.

Blend shape deformers let you to deform a surface into the shapes of other surfaces. You can blend shapes with the same or a number of   vertices (or CVs). In character setup, a typical use of a blend shape deformer is to set up poses for facial animation.



Unlike other deformers, the blend shape deformer has an editor that lets you control all the blend shape deformers in your scene. You can use the editor to control the influence of the targets of each blend shape deformer, create new blend shape deformers, set keys, and so on.

> **Note** Other software packages use the terms "morph," "morphing," and "shape interp" to refer to what Maya provides with blend shape deformers.

When creating a blend shape deformer, you identify one or more objects whose shapes you want to use to deform the shape of some other object. Objects whose shapes you want to use to deform the shape of another object are called *target object*s, and the object being deformed is called the *base object*.

## Target shapes, base shapes, and blend shapes

The shapes of the target objects are called target shapes, or target object shapes. The base object's resulting deformed shape is called the blend shape, and its original shape is called the base shape, or base object shape.

## Targets

A blend shape deformer includes a keyable attribute (channel) for evaluating each target object shape's influence on the base object's shape. These attributes are called targets, though by default they are named after the various target objects. Each target specifies the influence, or weight, of a given shape independently of the other targets. Depending on how you create or edit the blend shape deformer, however, a target can represent the influence of a series of target object shapes instead of just one shape.

## Scaling influence of all targets

You can scale the effect of all targets on the base by editing the Envelope channel or attribute.

Though the slider Envelope only specifies values from 0 to 1, you can enter values from -2 to 2. If Envelope is 2, the influence of every target is doubled. If Envelope is 0.5, the influence of every target is halved. If Envelope is negative, the influence of every target is inverted. If the cumulative effect of the targets deforms the base more than you want, you can scale down the overall deformation effect by setting Envelope to some value between 0 and 1.

For more information on setting the Envelope channel or attribute, see ”To edit Blend Shape channels with the Channel Box” on page 316 and ”To edit Blend Shape attributes with the Attribute Editor” on page 316.

## Matching position, rotation, and scaling of targets

You can control whether the deformation of the base is influenced by the position, rotation, or scaling of targets with the Origin attribute.

## Blending objects with different topologies

You can blend shapes with the same or different number of vertices or CVs.

When you create a blend shape deformer, you should turn the Check Topology creation option off if you want to blend objects that have different numbers of CVs or vertices. For more information on Check Topology, see ”Deform > Create Blend Shape” on page 364.

If objects have the same number of CVs or vertices but their order is different, Maya blends the shapes whether Check Topology is on or off. However, the position of the base CVs will be transformed to the position of the target CVs. This change might cause the object to blend in a way you might not expect. To ensure a smooth transition between base and target, make sure the order of CVs is the same in both objects.

In addition to blending individual objects, you can blend hierarchies of objects. Make sure both hierarchies have the same number of children and parenting relationships.

To blend hierarchies, you must select the parent of the target hierarchy (or hierarchies) first and the parent of the base hierarchy last before creating the blend shape. The parent of each must be a transform.

Each child in the base blends into its corresponding child in the target. The order of children in the Outliner (and Hypergraph) determines which children blend. If necessary, use the Outliner to change the order of objects in the hierarchies.

A common blend shape technique is to create duplicates of a base, deform the duplicates, then use them as targets. For example, you might make several copies of a face, and then alter the copies to create a smiling face, frowning face, a crying face, and so on. If you use this technique, turn on the Check Topology creation option when you create the blend shape deformer. This checks that the base and target hierarchy shapes have the same number of CVs. If the CVs are different and Check Topology is off, you might see, for instance, an eye blending into the nose. If Check Topology is on, the members of the hierarchies must have corresponding numbers of CVs.

## Deleting a target's object

After you create a blend shape, you can delete the target objects to free up memory and so improve Maya's performance. When you delete a target, the blend shape node keeps the target deformations in memory and the target slider deforms the base as if the target remained. The object is removed.

You save the most memory when you have complex targets that have only a few components that have moved slightly from the base. For complex targets that have many components moved from the base, you save the least memory.

Don't delete the targets if you want to modify their shapes or remove them from the blend shape. Remember that when you modify targets, Maya updates the resulting blend shape.

You can delete the object manually from your scene, or you can have Maya delete the targets when you create the blend shape.

## Setting target weights

To set the influence of targets on the blend shape, you adjust each target's weight slider. Each target's name is in a box under the slider. If the entire name of a target does not fit inside a box, drag left or right inside the box to see the undisplayed part.

You can move each slider from 0 to 1. A setting of 0 means that the target has no effect on the base. A setting of 1 makes the base identical to the target unless other targets also affect the base.

You can enter values beyond the slider range in the weight boxes below the sliders. A value above 1 exaggerates the target's influence. Negative values move the base in a direction opposite the target components. To reset all sliders to 0, click Reset All.

To adjust weight sliders, in the Blend Shape editor (Window > Animation Editors > Blend Shape), drag the slider or enter a value in the weight box.

## Setting keys for blend shapes

You can set key blend shapes with the Blend Shape editor. You can set keys on all the target sliders at their current values, or key an individual target slider at its maximum influence value (1). Keying an individual target slider at 1 keys the influence of that target slider only, ignoring the possible blending influences of the other target sliders.

## Saving a blend shape as a new target

After you create a blend shape from a mix of slider settings, you can save the shape as a new target for the base. After creating the new target, you can drag a single slider to deform the base object to that target.

## Selecting a blend shape deformer node

When you create a blend shape deformer, a blend shape deformer algorithm node (default name: blendShape*n*) appears in the scene's dependency graph as an input of the base object's shape node. This node uses the target slider weight settings to create a blend shape from the base.

The blend shape deformer node name appears in the Blend Shape editor above and to the left of the associated target sliders. To display the animation keys of weights in the Time Slider, Graph Editor, and the Dope Sheet, you must select the blend shape deformer node.

### Creating a new blend shape deformer

You can create a blend shape deformer using the Blend Shape editor instead of by selecting Deform > Create Blend Shape. The new blend shape node is chained sequentially by default. If you want to put one blend shape node in conjunction with another one, change the deformer placement to parallel.

## Adding target object shapes

You can add target object shapes to a blend shape deformer. When adding target object shapes, you can first set the add options and then add the target object shapes, or you can immediately add the objects with the current add options. See "To add a target shape for blend shape deformers" on page 317.

## Removing target object shapes

You can remove target object shapes from a blend shape deformer. When removing target object shapes, you can first set the remove options and then remove the target object shapes, or you can immediately remove the objects with the current remove options. See "To remove a target shape" on page 318.

## Swapping target object shapes

You can swap the order of target object shapes. When swapping target object shapes, you can first set the swap options and then swap the target object shapes, or you can immediately swap the objects with the current swap options. See "To swap two target shapes" on page 318.

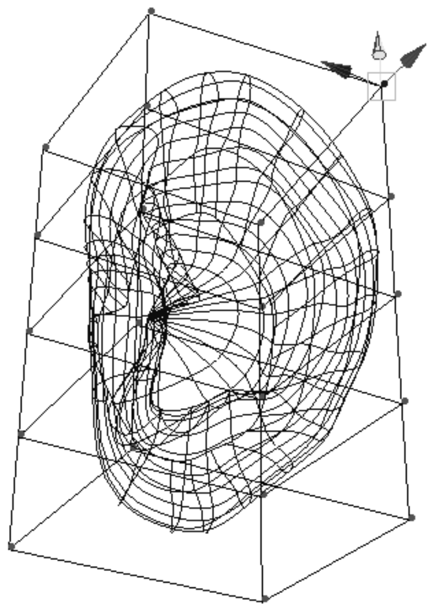
## Changing Blend Shape editor slider orientation

You can control the orientation of the sliders in the Blend Shape window (Window > Animation Editors > Blend Shape). The sliders can be arranged vertically or horizontally, whichever is most intuitive for you. To orient the sliders vertically, select Options > Orientation > Vertical (the default). To orient the sliders horizontally, select Options > Orientation > Horizontal.

### Lattice deformers

## Lattice deformers

Lattice deformers let you deform objects with lattices. A lattice deformer surrounds a deformable object with a lattice that you can manipulate to change the object's shape.



Lattice deformer acting on an ear. You can create deformation effects by moving the lattice's points.

## Lattices

A lattice is a structure of points for carrying out free-form deformations on any deformable object. To create deformation effects, you edit the lattice by moving, rotating, or scaling the lattice structure, or by directly manipulating the lattice points. In general, you create effects by editing any of the lattice deformer's attributes.

## Influence lattice and base lattice

A lattice deformer includes two lattices: an influence lattice and a base lattice. By itself, the term "lattice" typically refers to the influence lattice. You create deformation effects by editing or animating the influence lattice. The lattice deformer's effect is based on any difference between the base lattice's lattice points and the influence lattice's lattice points. By default, the base lattice is hidden so that you can focus on manipulating the influence lattice. However, remember that the deformation effect depends on the relationship between the influence lattice and the base lattice.

## Lattices as deformable objects

Unique among deformer influence objects, lattices are themselves deformable objects. That means that you can create deformers that deform a lattice. For example, you can deform a lattice with a sculpt deformer, and the effect of the deformation on the lattice points will in turn deform the object the lattice is deforming. You can also assign deformation weights to lattice points by creating a cluster deformer for the lattice. Also, you can bind a lattice to a skeleton. When you move the skeleton, the lattice will deform with the action of the joints.

## Lattice deformers and lattice flexors

Flexors are special objects you use to control the deformation effects of rigid skinning. Two types of flexors, joint lattice flexors and bone lattice flexors, use lattice deformer nodes.

### Resetting influence lattice shape and location

You can reset the deformed lattice to return it to the location and shape of the base lattice.

Reset the lattice to clear all adjustments you have made to the influence lattice. Do this when you want to:

- start over with the deformation
- rotate, scale, or translate the base lattice and the deformed lattice together, from their initial positions
- parent the base lattice at the center of the deformed lattice, before manipulating the lattice.

### Resetting influence lattice points and removing tweaks

After lattice point tweaking or changing the STU divisions, you can reset the lattice points to their original positioning in local space. This does not modify the lattice object’s transformation in world space. Use this when you want to change the number of divisions on the lattice or start over with the deformation.

## Pruning lattice deformer sets

By pruning lattice deformer sets, you can remove points from the set that are not presently being affected by the deformer. You can prune the deformer set to avoid unnecessary calculations for points that are not part of the deformation effect.

## Changing influence lattice resolution

To deform the geometry by a finer or coarser resolution, you can change the number of lattice points. Using the Divisions attribute, you can increase or decrease the divisions along S, T, and U (the X, Y, and Z axis, respectively, if the lattice were in the default position at the origin).

Note that the greater the number of divisions, the more calculations Maya has to do to deform the geometry and the slower the performance. To speed up the performance to counteract the effect of a high-resolution lattice, see "Improving performance" on page 99.

You gain no resolution in the deformation by having more lattice points in the lattice than points on the geometry. The resolution is limited by the spacing of points across the geometry. Note that if you've moved the points, you have to reset the points before changing the resolution.

You can reset the lattice by choosing Deform > Edit Lattice > Reset Lattice. However, you cannot change the resolution of a lattice if the lattice points have been moved from their reset position or the lattice has history. If you want to change the number of divisions on a lattice whose points have been moved, choose Deform > Edit Lattice > Reset Lattice Tweaks, and then change the divisions of the lattice. If you want to change the divisions on a lattice with history, find the input lattice shape and change its divisions. You can find the input lattice by selecting the lattice and looking in the attribute editor tabs for the original lattice shape, which will typically share the same base name as the output lattice appended by "Orig."

## Weighting lattice points to alter their influence

The influence of individual lattice points can vary if the lattice points have been assigned weights. You can have weights assigned to lattice points in two ways:

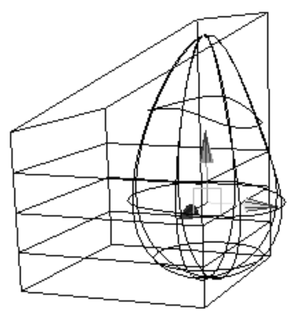
- You can create a cluster deformer that acts on the lattice deformer. You can then control the weights assigned to each lattice point by the cluster deformer. For more information on cluster deformers, see "Cluster deformers" on page 250.
- You can do skinning with lattice deformers. For more information, see "Skinning with lattice deformers" on page 249.

## Sculpting the influence lattice

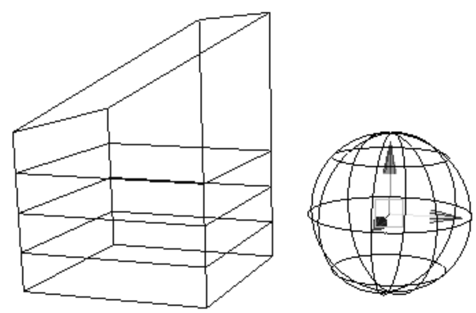
You can use a sculpt deformer to shape a lattice deformer's influence lattice. Using a sculpt deformer in this way can provide a great way to get smooth, rounded lattice deformations. Trying to achieve the same rounded effect by tweaking (moving) lattice points could require some painstaking effort. For more information on sculpt deformers, see "Sculpt deformers" on page 264.

## Freezing the lattice deformation mapping

A lattice deformer's deformation effects normally depend on whether the objects being deformed are inside the base lattice (default name: ffd*n*Base). If the objects are completely outside of the base lattice, deformation effects cease. The effects cease because the lattice deformer calculates the effects based on the spatial relationship between the base lattice, the influence lattice, and the positions of the objects inside the base lattice. If the objects are outside of the base lattice, Maya cannot calculate the effects. Similarly, if an object is only partially inside the base lattice, only the components (for example, CVs) of the object inside the base lattice can be affected by the influence lattice.

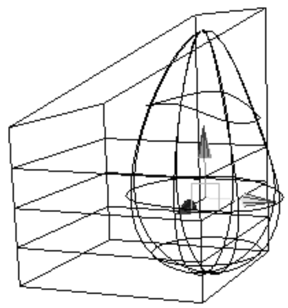


Influence lattice deforming sphere partially inside lattice.

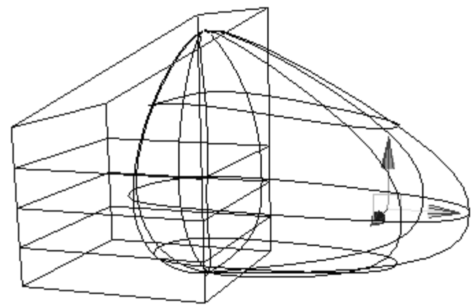


Sphere no longer affected by influence lattice because it is now outside of the base lattice. (The base lattice is not displayed unless selected, but by default it is located inside the influence lattice, and has the influence lattice's original shape.)

Freezing the lattice deformation mapping locks the influence lattice's control over deformable object components inside the influence lattice. If you want to have the components of deformable objects inside of the influence lattice to remain inside and under the influence of the influence lattice even when you move the objects or the base lattice, you can do so by freezing the lattice deformation mapping.



Influence lattice deforming sphere partially inside lattice. Now the lattice deformation mapping is frozen.



Because lattice deformation mapping has been frozen, components of spheres inside the influence lattice remain inside it even as you move the sphere outside of the base lattice and influence lattice.

Freezing the lattice deformation mapping freezes an influence lattice's relationship with all of the components inside it. Object components (for example, CVs) inside the influence lattice at the time of freezing will stay fixed inside the lattice, and will only be affected by changes to the influence lattice. Even if you move the base lattice so that those components are no longer in it, the components themselves will remain under the influence of the influence lattice. However, components outside the influence lattice will move when you move the object, causing deformation effects on the object.

You can freeze the deformation mapping when you create a lattice deformer by turning the Freeze Mode creation option on. See "Deform > Create Lattice" on page 368. However, if the influence lattice surrounds the object, you won't be able to move any of its components outside of the influence lattice unless you change the Freeze Geometry attribute. After you've created a lattice deformer, you can change whether it freezes the deformation mapping by editing the Freeze Geometry attribute. See "Lattice deformer nodes" on page 407.

## Editing the base lattice

The base lattice is not visible unless you select it. You can move, rotate, or scale a base lattice. However, unlike the influence lattice, the base lattice does not have lattice points.

## Grouping base and influence lattices

To move the base lattice and the deformed lattice together, you can group them. Group the lattices when you are moving the lattice to deform a stationary geometry. For example, suppose you wanted to slam a fish bowl on a character's head so that the head then conforms to the shape of the bowl. To do this, you could use a lattice deformer to shape a geometry for the fishbowl and a geometry for the head. You could then group the base lattice and deformed lattice, parent them to the fishbowl geometry, and move them all away from the head. When you move them away, the head will return to its normal shape, but when you move them back the head will take the shape of the fishbowl.

## Parenting lattices to objects being deformed

You can parent the lattice and base lattice to the objects being deformed by them so that you can move and continue the deformation. For example, if your character’s squashed hat is deformed with a lattice, parent the lattice to the hat and the base lattice so the hat stays deformed as it moves.

## Deforming a lattice with other deformers

You can deform any deformed lattice just as you can deform a geometry with other deformers. For example, you can put a sculpt or cluster deformer on a lattice, and deform the lattice shape.

## Assuring a smooth deformation through the base lattice

If you are setting up or animating the deformable object so that it passes through the base lattice, note that editing the outermost parts of the influence lattice can cause a sudden deformation in the object’s shape. If you have edited the outermost parts of the influence lattice, the deformable object will suddenly deform as it enters the base lattice.

## Improving performance

The greater the number of lattice points, the greater your control over the deformation, but the slower the performance. Note that a lattice should always have fewer lattice points than the deformable object has points. You gain no increase in control over the deformation by having more lattice points than the object itself has points.

You can edit node behavior to improve performance. For more information, see ”To change lattice deformer performance settings” on page 322.

You can also change the lattice resolution performance. To change the lattice resolution performance, set the Partial Resolution attribute. See ”To change lattice resolution settings” on page 322.

## Skinning with lattice deformers

Skinning is the process of binding deformable objects to a skeleton. Typically, the deformable objects that are bound are NURBS or polygonal surfaces. These geometry objects become the character's surface, or skin, and their shapes are influenced by the action of the skeleton's joints. Once you've built a skeleton for a character, you can skin your character by using a smooth skinning method or a rigid skinning method.

Because influence lattices are deformable objects, you can also bind them to a skeleton by smooth or rigid skinning. In turn, these can influence the NURBS or polygonal surfaces that provide the character's skin.

In skinning with lattice deformers, you create lattice deformers for the deformable objects that you want to use for the character's skin. Then you bind the influence lattices to the skeleton. The result is that the skeleton's movement influences the objects indirectly through the lattice deformers. Meanwhile, you can manipulate the influence lattices for more control over the deformation. This approach, skinning with lattice deformers, is called lattice skinning.

If you wish, you could use lattice skinning to skin an entire character. A more common approach is to use smooth or rigid skinning for much of the character's skin, but then also to use lattice skinning for finer control over certain areas. In many situations, lattice skinning can provide superior smoothing effects, particularly in areas near where a character's limbs and main body meet (for example, a shoulder and armpit area). However, if you do use lattice skinning with smooth or rigid skinning, you need to be very careful about how all the many control points (CVs, polygonal vertices, or lattice points) involved are organized. You will need to organize points into various sets and those sets into various partitions to make editing easier and to avoid double transformation effects.

**Note** Lattice skinning should not be confused with lattice flexors. Lattice flexors are for use with rigid skinning only. They help to smooth out deformations provided by rigid skinning, and their influence is by default limited to the skin area near a particular joint.

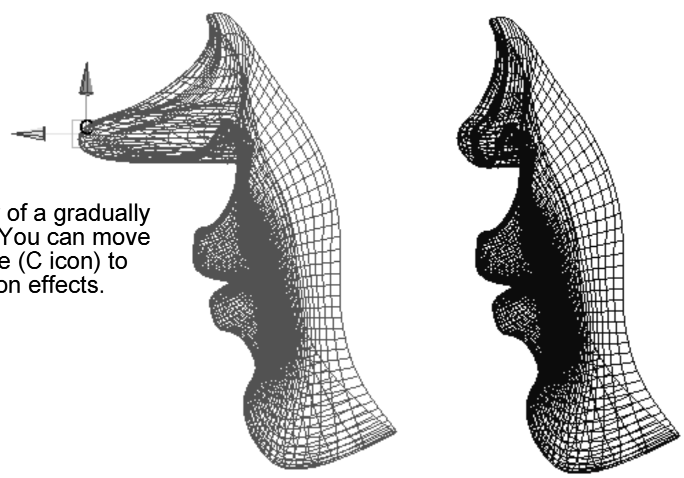
For more information on skinning, see the Skinning chapter of the Character Setup Guide.

### Cluster deformers

## Cluster deformers

Cluster deformers let you control a set of an object’s points (CVs, vertices, or lattice points) with varying amounts of influence.

A cluster deformer creates a set whose members consist of selected points (CVs, vertices, or lattice points). You can assign a percentage weight to each point, indicating how much you want each point to be affected by any translation, rotation, or scale of the cluster set. When you transform the cluster, the points are transformed according to the percentages you specify.



Cluster deformer of a gradually stretching nose. You can move the cluster handle (C icon) to create deformation effects.

## Understanding cluster deformers

A cluster deformer applies a transformation to a geometry’s points in such a way that you can adjust the percentage that each point is affected by the transformation.

Use clusters when you want to affect geometry in different amounts by one or more transformations. For example, you can create a cluster for a door so that when it is slammed, the middle bows slightly.

## Creating cluster deformers

When creating cluster deformers, you can first set creation options and then create a deformer, or you can immediately create a deformer with the current creation options. If you're not sure what the current creation options are, checking them before you create a deformer can save you time adjusting the deformer's attributes afterwards.

| | |
|---|---|
| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. You might want to save a copy of the object in case you want to do further modeling. |

## Editing cluster deformation effects

After you have created a cluster deformer, you can edit the deformer's effects as described in the following topics.

### Manipulating the cluster handle (C icon)

Each cluster deformer includes a cluster deformer handle. In the workspace, the handle is a C icon (the letter "C"). You can select the handle, and move, rotate, and scale it to create deformation effects. Of course, the effects depend on the cluster weights that control the effect of the cluster deformer on the deformable object's points (NURBS CVs, polygonal vertices, or lattice points). For more information on editing and painting cluster weights, see "Editing cluster weights" on page 252 and "Painting cluster weights" on page 252.

Note that wrinkle deformers use cluster deformers, and that you can also manipulate the effects of wrinkle deformers with the cluster deformer handle's C icon. For more information on wrinkle deformers, see "Wrinkle deformers" on page 277.

### Pruning cluster deformer sets

By pruning cluster deformer sets, you can remove points from the set that are not presently being affected by the deformer. You can prune the deformer set to avoid unnecessary calculations for points that are not part of the deformation effect. See "To prune deformer set membership" on page 323.

## Editing cluster weights

After you create the cluster, set up the percentages based on the amount that you want the points or control vertices to be affected.

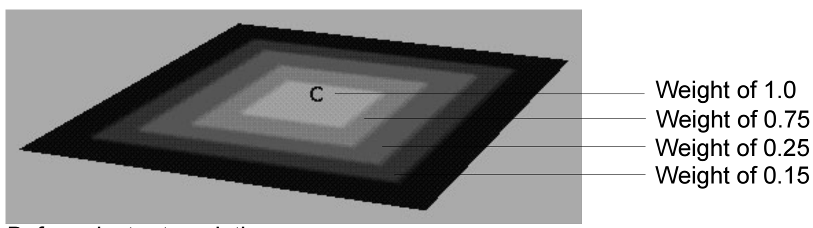
For some cases, you may set one percentage for the whole object. For example, a stick on moving water, in which the cluster handle is set to move at 50% of the transformation of the water.

At other times, you may want some parts of the geometry to be affected more or less than other parts. For example, you could have a scene with waving trees, where the treetops are affected the most, and the trunk near the ground affected by 0%.

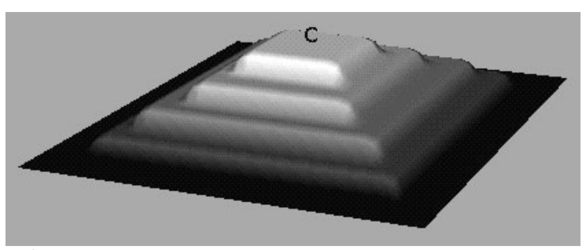
With the Component Editor, you can directly modify the values of individual cluster weights. You can also paint cluster weights.

## Painting cluster weights

Using the Paint Cluster Weights Tool, you can set cluster weights simply by painting over the clustered surface. Although you can transform the cluster first, then paint weights on the surface, the Paint Cluster Weights Tool provides color feedback so you know which parts of the clusters have different weights before you transform the cluster. Weights display as a range of grayscale values, with a weight of 1 displaying as white and 0 as black. See ”Deform > Paint Cluster Weights Tool” on page 397.



Before cluster translation



After cluster translation in Y direction

## Painting weights on restricted areas

When you paint weights over selected vertices, your strokes are applied only to the weights corresponding with the selected vertices. In effect, the unselected vertices act as a mask, where only selected vertices are affected by any painting or flooding you do.

Before creating the mask you must first create the cluster. For details on masking surfaces, see the Paint Effects, Artisan, and 3D Paint guide.

## Flooding clusters

Flooding a cluster is like taking a huge brush and applying its settings to the entire cluster. When you flood a cluster, the weight of each vertex in the cluster changes according to the value and operation set for the tool.

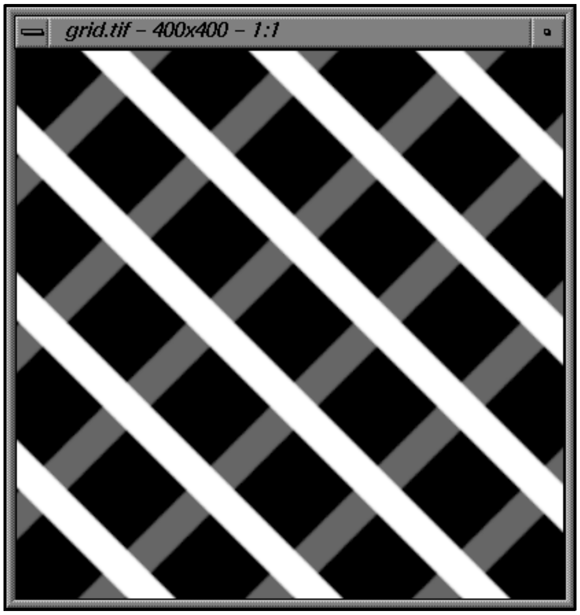
To flood a cluster, follow the steps under ”To paint weights on a cluster” on page 324, but instead of painting in step 7, click the Flood button or use the hotkey Alt+f (IRIX, Linux, and Windows) or Option+f (Mac OS X).

| Tip | To smooth the transition between cluster weights, select the Smooth paint operation and flood the cluster. |
|---|---|

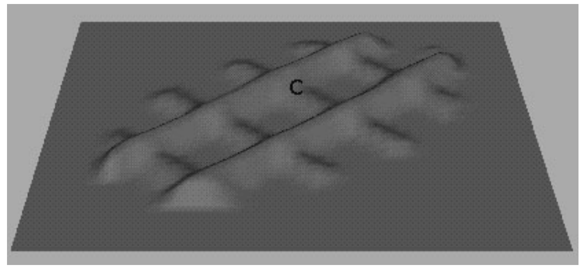
## Mapping weight values to clusters

Using the Paint Cluster Weights Tool you can map attribute values onto surface vertices relative to the UVs. The settings for the tool are applied to the cluster vertices using the mapped values.

In the following example, notice that only the cluster is affected by the map.




Attribute map



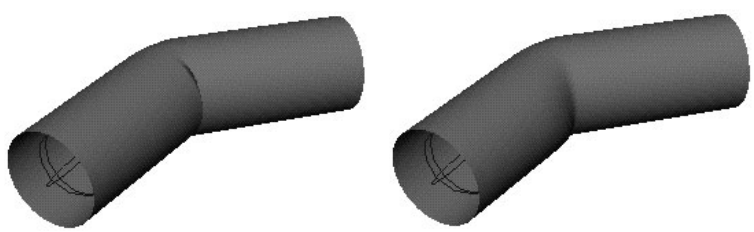

Mapped surface

For details on mapping, see the Paint Effects, Artisan, and 3D Paint guide.

## Painting weights on rigid skin objects

When using rigid skinning, Maya creates a joint cluster node for each joint of the skeleton. The joint cluster nodes assign weights to the rigid skin points to control how the rigid skin objects deform. You can use the Paint Cluster Weights Tool to modify the rigid skin point weights.

In the following example, notice how the surface folds at the joint. You can use the Paint Cluster Weights Tool to quickly smooth out the fold.



Before smoothing After smoothing

Note that painting smooth skin point weights requires the use of a different painting tool. See ”Painting smooth skin point weights” on page 151.

### Setting the cluster relative to the parent transform

Using the Relative attribute, you can set the cluster deformation to be active only when the direct parent of the cluster handle is transformed. This lets you create effects where a hierarchy of parent objects do not all affect the cluster deformation.

For example, if you parent the cluster handle to a wrist joint and turn on its Relative attribute, you can rotate the shoulder without the cluster affecting the skin around the wrist, even though the wrist's position changes. When you move the wrist itself, the cluster deforms the geometry as desired.

### Controlling the deformation percentage of the entire cluster

You can control the percentage of deformation for the entire cluster using the Envelope channel. Change Envelope to scale all the cluster weights in the same way. If you set the Envelope value and also various values for cluster weights, all the values affect the deformation.

### Using weighted nodes

You can use another object for the cluster handle, the movement of which controls the cluster. Specify the object you want to use in the cluster handle shape node, under the Weighted Node attribute.

### Setting the location of the cluster handle

You can control the placement of the cluster handle (displayed as a "C") by specifying the location of the cluster handle's origin. To do so, from the cluster*n*HandleShape tab of the Attribute editor, set the Origin attribute. The Origin attribute includes fields for X axis, Y axis, and Z axis values.

**Bend Nonlinear deformers**

## Bend Nonlinear deformers

Bend deformers let you bend any deformable object along a circular arc. They are useful both for character setup and modeling. Bend deformers include handles that enable you to control the extent and curvature of the bending effects in an intuitive manner.


## Creating bend deformers

When creating bend deformers, you can first set creation options and then create a deformer, or you can immediately create a deformer with the current creation options. If you're not sure what the current creation options are, checking them before you create a deformer can save you some time adjusting the deformer's attributes afterwards.

See "Set bend nonlinear deformers" on page 286.

**Warning** You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. Save a copy of the object in case you want to do further modeling later.

## Editing bend deformation effects

After you create the bend deformer, its handle is displayed in your scene and its nodes are listed in the Channel Box. The nodes include the bend handle node (default name: bend*n*Handle), the bend handle shape node (bend*n*HandleShape), and the bend deformer node (default name: bend*n*).

You can edit the effects of the bend deformer by editing the bend handle node and the bend deformer node. You can move (translate), rotate, and scale the bend handle to edit the effects of the deformation. You can also edit the bend deformer node’s keyable attributes (channels), which are displayed in the Channel Box.

See ”Edit bend nonlinear deformers” on page 326 and ”Nonlinear deformer nodes” on page 410.

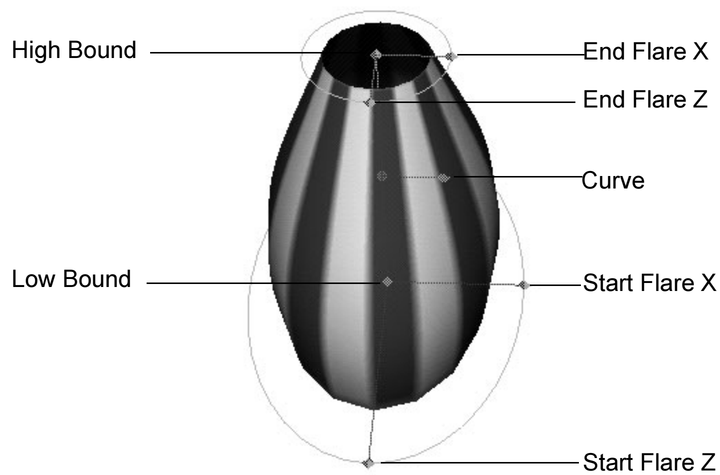
### Flare Nonlinear deformer

## Flare Nonlinear deformer

Flare deformers let you flare out or taper in any deformable object along two axes. They are useful both for character setup and modeling. Flare deformers include handles that let you to control the extent and curvature of the flaring or tapering effects in an intuitive manner.

Flare deformer acting on a cylinder

Flare deformer handle manipulators




## Creating flare deformers

When creating flare deformers, you can first set creation options and then create a deformer, or you can immediately create a deformer with the current creation options. If you're not sure what the current creation options are, checking them before you create a deformer can save you some time adjusting the deformer's attributes afterwards.

| | |
|---|---|
| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. Save a copy of the object in case you want to do further modeling later. |

## Editing flare deformation effects

After you create the flare deformer, its handle is displayed in your scene and its nodes are listed in the Channel Box. The nodes include the flare handle node (default name: flare*n*Handle), the flare handle shape node (flare*n*HandleShape), and the flare deformer node (default name: flare*n*).

You can edit the effects of the flare deformer by editing the flare handle node and the flare deformer node. You can move (translate), rotate, and scale the flare handle to edit the effects of the deformation. You can also edit the flare deformer node's keyable attributes (channels), which are displayed in the Channel Box.

You can edit flare deformation effects in a variety of ways:

- "To edit with the flare handle manipulators"
- "To edit flare deformers by moving, rotating, or scaling the handle"
- "To edit flares with the Channel Box"
- "To edit flare deformers with the Attribute Editor"

### Sine Nonlinear deformer

## Sine Nonlinear deformer

Sine deformers let you manipulate any deformable object along a sine wave. Sine deformers are useful both for character setup and modeling. Sine deformers include handles that you can use to control the extent, amplitude, and wavelength of the sine wave effects in an intuitive manner.

The sine deformer changes the shape of an object along a sine wave.

Sine deformer acting on a cylinder

## Creating sine deformers

When creating sine deformers, you can first set creation options and then create a deformer, or you can immediately create a deformer with the current creation options. If you're not sure what the current creation options are, checking them before you create a deformer can save you some time adjusting the deformer's attributes afterwards.

| | |
|---|---|
| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. Save a copy of the object in case you want to do further modeling. |

## Editing sine deformation effects

After you create the sine deformer, its handle is displayed in your scene and its nodes are listed in the Channel Box. The nodes include the sine handle node (default name: sine*n*Handle), the sine handle shape node (sine*n*HandleShape), and the sine deformer node (default name: sine*n*).

You can edit the effects of the sine deformer by editing the sine handle node and the sine deformer node. You can move (translate), rotate, and scale the sine handle to edit the effects of the deformation. You can also edit the sine deformer node's keyable attributes (channels), which are displayed in the Channel Box.

You can edit bend deformation effects in a variety of ways:

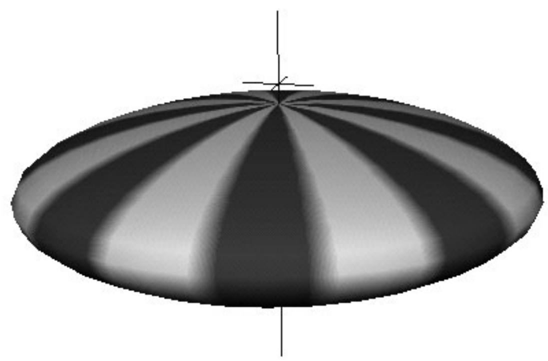
- "To edit sine deformers by using the handle manipulators"
- "To edit sine deformers by moving, rotating, or scaling the handle"
- "To edit sine deformers with the Channel Box"
- "To edit sine deformers with the Attribute Editor"

### Squash Nonlinear deformer

## Squash Nonlinear deformer

Squash deformers let you squash and stretch any deformable object along an axis. They are useful both for character setup (classic squash and stretch effects) and modeling. Squash deformers include handles that let you control the extent and magnitude of the squash or stretch effects in an intuitive manner.

The squash deformer squashes and stretches objects.




## Creating squash deformers

When creating squash deformers, you can first set creation options and then create a deformer, or you can immediately create a deformer with the current creation options. If you're not sure what the current creation options are, checking them before you create a deformer can save you some time adjusting the deformer's attributes afterwards.

| | |
|---|---|
| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. Save a copy of the object in case you want to do further modeling. |

## Editing squash deformation effects

After you create the squash deformer, its handle is displayed in your scene and its nodes are listed in the Channel Box. The nodes include the squash handle node (default name: squash*n*Handle), the squash handle shape node (squash*n*HandleShape), and the squash deformer node (default name: squash*n*).

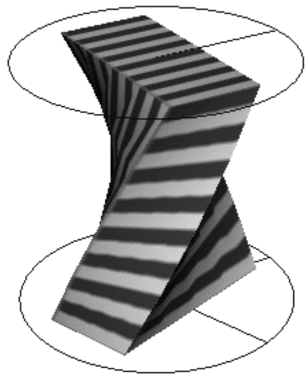
You can edit the effects of the squash deformer by editing the squash handle node and the squash deformer node. You can move (translate), rotate, and scale the squash handle to edit the effects of the deformation. You can also edit the squash deformer node's keyable attributes (channels), which are displayed in the Channel Box.

**Twist Nonlinear deformer**

## Twist Nonlinear deformer

Twist deformers let you twist any deformable object about an axis. They are useful both for character setup and modeling. Twist deformers include handles that enable you to control the extent and degree of the twisting effects in an intuitive manner.

The twist deformer twists the shape of an object.



Twist deformer
acting on a box

## Creating twist deformers

When creating twist deformers, you can first set creation options and then create a deformer, or you can immediately create a deformer with the current creation options. If you're not sure what the current creation options are, checking them before you create a deformer can save you some time adjusting the deformer's attributes afterwards.

| | |
|---|---|
| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. Save a copy of the object in case you want to do further modeling. |

## Editing twist deformation effects

After you create the twist deformer, its handle is displayed in your scene and its nodes are listed in the Channel Box. The nodes include the twist handle node (default name: twist*n*Handle), the twist handle shape node (twist*n*HandleShape), and the Twist deformer node (default name: twist*n*).

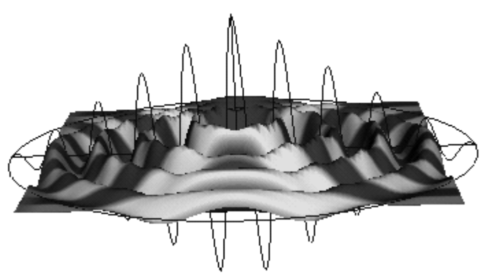
You can edit the effects of the twist deformer by editing the twist handle node and the twist deformer node. You can move (translate), rotate, and scale the twist handle to edit the effects of the deformation. You can also edit the twist deformer node's keyable attributes (channels), which are displayed in the Channel Box.

### Wave Nonlinear deformer

## Wave Nonlinear deformer

The wave deformer is similar to the sine deformer. The sine deformer's sine wave propagates in the deformer's local Y axis, with the amplitude along the X axis. The wave deformer's sine wave propagates along the deformer's local X and Z axes, with the amplitude along the Y axis. Wave deformers include handles that enable you to control the extent, amplitude, and wavelength of the wave effects in an intuitive manner.

The wave deformer deforms an object based on a circular sine wave for ripple effects. If you'd like to explore an example now, see "Wave deformer example" on page 344.



Wave deformer acting on a plane

## Creating wave deformers

When creating wave deformers, you can first set creation options and then create a deformer, or you can immediately create a deformer with the current creation options. If you're not sure what the current creation options are, checking them before you create a deformer can save you time adjusting the deformer's attributes afterwards.

| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. Save a copy of the object in case you want to do further modeling. |
|---|---|

## Editing wave deformation effects

After you create the wave deformer, its handle is displayed in your scene and its nodes are listed in the Channel Box. The nodes include the wave handle node (default name: wave*n*Handle), the wave handle shape node (wave*n*HandleShape), and the wave deformer node (default name: wave*n*).

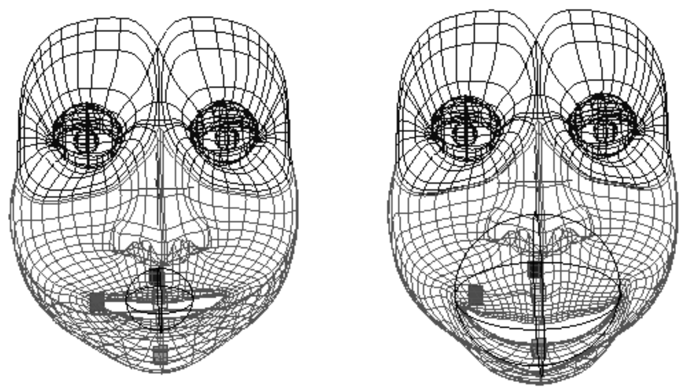
You can edit the effects of the wave deformers by editing the wave handle node and the wave deformer node. You can move (translate), rotate, and scale the wave handle to edit the effects of the deformation. You can also edit the wave deformer node's keyable attributes (channels), which are displayed in the Channel Box.

### Sculpt deformers

## Sculpt deformers

Sculpt deformers are useful for creating any kind of rounded deformation effect. For example, in setting a character for facial animation, you could use sculpt deformers to control the character's chin, brow, or cheek actions.

With sculpt deformers, you can deform objects with a spherical influence object called a sculpt sphere, or any NURBS object you create.



## Sculpt Tool

You can use the following objects as sculpt tools: an implicit sphere or a NURBS object.

The implicit sphere is the spherical wireframe influence object you manipulate to create deformation effects. The implicit sphere's deformation effects depend on the mode of the sculpt deformer.

In addition to the sculpt sphere, you can also use a secondary NURBS object as the sculpt tool. Using a NURBS object as your sculpt tool is also referred to as using the *custom sculpt tool*. See "To use a NURBS surface with the sculpt deformer" on page 290.

| | |
|---|---|
| Note | Deleting the sculpt node or the deformed object does not delete the custom sculpt tool. |

## Flip mode

A sculpt deformer in flip mode has an implicit locator in the center of the sculpt tool. When the sculpt tool is near the geometry, deformation occurs. This mode is called flip mode because as the center of the sculpt tool passes through the surface, the deformed surface flips to the other side of the sculpt tool.

## Project mode

In project mode the sculpt deformer projects the geometry onto the surface of the sculpt tool. The extent to which the projection takes place depends on the sculpt deformer's Dropoff Distance attribute.

While the Dropoff Distance specifies the extent of the projection directly onto the sculpt tool, the Maximum Displacement attribute specifies whether the projection takes place directly onto the sculpt tool, inside the sculpt tool, or outside of the sculpt tool.

With a Maximum Displacement of 1.0, the projection takes place on the surface of the sculpt sphere. This is the effect you would usually want to achieve with project mode. However, by changing the Maximum Displacement you can displace the projection from the surface of the sculpt sphere. With a Maximum Displacement between 0 and 1.0, the projection takes place within the sculpt tool. With a Maximum Displacement greater than 1.0, the projection takes place outside of the surface of the sculpt tool. With a Maximum Displacement of 0, the geometry is projected into the center of the sculpt tool.

With a Maximum Displacement of less than 0, the projected geometry turns inside out as it is projected through the center of the sculpt sphere.

### Stretch mode

In stretch mode, as you move the sculpt sphere away from the geometry, the affected surface of the geometry stretches or bulges to stay with the sculpt sphere. The stretch direction extends from the point marked by a stretch origin locator to the surface of the sculpt sphere.

When you create a sculpt deformer in stretch mode, you can select and move the stretch origin locator as you do any object, or parent it to the sculpt tool and move them in combination. Depending on the effect you want to create, you could also parent the locator to some other object in the animation.

## Creating sculpt deformers

When creating sculpt deformers, you can first set creation options and then create a deformer, or you can immediately create a deformer with the current creation options. If you’re not sure what the current creation options are, checking them before you create a deformer can save you some time adjusting the deformer’s attributes afterwards.

As with previous versions of Maya, it is possible to have multiple sculpt deformers deforming the same object.

You can also use the connection editor or the MEL connectAttr command to manually connect a different object as the sculpt tool.

| | |
|---|---|
| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. Save a copy of the object in case you want to do further modeling later on. |

### Limitations

- Geometry types other than implicit spheres and NURBS objects are not currently supported. To use another type of surface, convert it to a NURBS surface.
- Performance can degrade as the number of vertices in the sculpt tool and/or the object being deformed increase.
- For best results use closed, convex objects as sculpting tools.

### Advanced

- As with previous versions of Maya, it is possible to have multiple sculpt deformers deforming the same object.
- You can use the connection editor or the MEL connectAttr command to manually connect a different object as the sculpt tool.

## Editing sculpt deformation effects

You can create sculpt deformation effects as described in the following topics:

### Manipulating the stretch origin locator

If the sculpt deformer is in stretch mode (Mode attribute is set to Stretch), you can create deformation effects by directly manipulating the stretch origin locator as well as the sculpt sphere.

### Pruning sculpt deformer sets

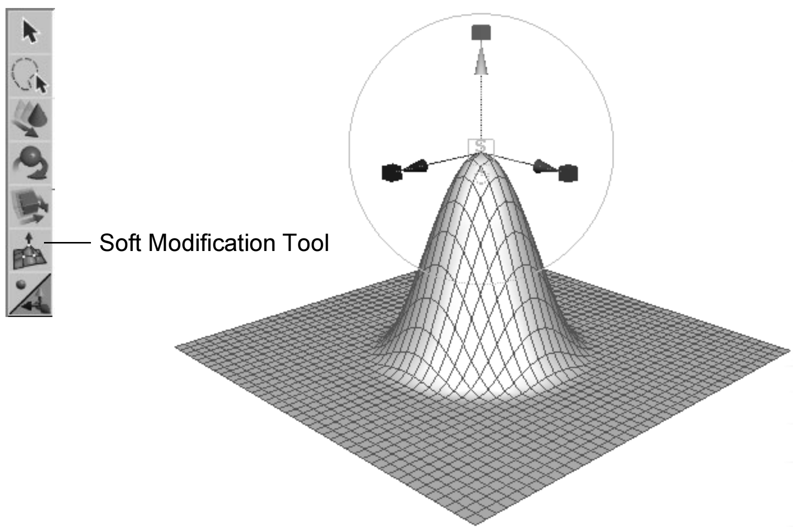
By pruning sculpt deformer sets, you can remove points from the set that are not presently being affected by the deformer. You can prune the deformer set to avoid unnecessary calculations for points that are not part of the deformation effect.

## Soft Modification

# Soft Modification

The Soft Modification Tool and deformer lets you push and pull geometry as a sculptor would push and pull on a piece of clay.

By default, the amount of deformation is greatest at the center of the push/pull, and gradually falls off further away from the center. However, you can control the falloff of the deformation to create various types of effects.




The Soft Modification Tool is located in the Tool Box. You can also access the tool from Deform > Soft Modification in the Maya main menu bar.

You can use the Soft Modification Tool on NURBS surfaces, polygonal surfaces, subdivision surfaces, curves, particles or any object with components.

### Related topics

- ❖ "Use the Soft Modification Tool" on page 291
- ❖ "Deform > Soft Modification" on page 384

## Jiggle deformers

Jiggle deformers let you cause points on a surface or curve to shake as they move, speed up, or slow down.

Image by Lee Graft

For instance, you can create effects such as:

- a wrestler’s stomach shaking
- hair jiggling
- an insect’s antennae vibrating

You can apply jiggle to specific points or to the entire object. In the context of jiggle deformers, the term *points* means CVs, lattice points, or the vertices of polygonal or subdivision surfaces.

A useful technique is to apply jiggle to an influence object that underlies and alters the skin.

Be aware that you can create two or more jiggle deformers on different points of a single object. You can get the same effect more simply by applying a single jiggle deformer to the points and adjusting the jiggle weights. See ”Adjusting jiggle weight by painting” on page 270.

## Creating Jiggle deformers

You can set creation options and then create a deformer, or you can create a deformer with the current creation options and edit the options later.

### Adjusting jiggle weight by painting

After you create a jiggle deformer for an object or specific points, you can tune the jiggle of individual points by painting their jiggle weight values. Typically you'll get best results if you use higher jiggle weight values at the central area of jiggle region and lower values at the edge of the jiggle region. Fading the values from center to edge often works well. See ”To paint jiggle weights” on page 299.

### Painting weights on masked jiggle deformers

When you paint weights over selected vertices, your strokes are applied only to the weights corresponding with the selected vertices. In effect, the unselected vertices act as a mask, where only selected vertices are affected by any painting or flooding you do.

Before creating the mask you must first create the jiggle deformer. For details on masking surfaces, see the Paint Effects, Artisan, and 3D Paint Guide.

### Flooding jiggle deformers

Flooding a jiggle deformer is like taking a huge brush and applying its settings to the entire cluster. When you flood a deformer, the weight of each vertex in the deformer changes according to the value and operation set for the tool.

To flood a cluster, follow the steps under ”To paint jiggle weights” on page 299, but instead of painting in step 7, click the Flood button or use the hotkey Alt+f (IRIX, Linux, and Windows) or Option+f (Mac OS X).

| | |
|---|---|
| Tip | To smooth the transition between jiggle weights, select the Smooth paint operation and flood the jiggle deformer. |

### Mapping weight values to jiggle deformers

Using the Paint Cluster Weights Tool you can map attribute values onto surface vertices relative to the UVs. The settings for the tool are applied to the cluster vertices using the mapped values.

For details on mapping, see the Paint Effects, Artisan, and 3D Paint guide.

### Paint Jiggle Weights Tool settings

To modify Paint Jiggle Weights Tool settings, select the Paint Jiggle Weights Tool and open the Tool Settings editor (Deform > Paint Jiggle Weights Tool > ❐).

## Adjusting jiggle weight by painting

| | |
|---|---|
| Tip | You can define hotkey combinations to change most of the settings without opening the Tool Settings editor. For details, see the Paint Effects, Artisan, and 3D Paint guide. |

See ”To paint jiggle weights” on page 299.

### Using disk cache for jiggle animation

When you create jiggle disk cache for your scene, Maya stores on disk the frame-by-frame processing of jiggle animation. Before you can render jiggle animation with motion blur, you must create jiggle disk cache.

More specifically, if the Motion Blur option is turned on in the Render Globals, you must create jiggle disk cache before you render the animation. Otherwise the rendered animation sequence will display the animation incorrectly. Note that the jiggling object's Motion Blur option setting is irrelevant to this issue.

Creating the cache is also beneficial for scenes that play slowly because of their complexity. If you create the cache, you can go directly to any frame in the Time Slider and view an accurate jiggle animation. Without the cache, you would need to wait for the animation to play from the beginning of the Time Slider up to the current frame.

Jiggle processing is efficient and does not slow a scene much, unless your scene plays unusually slowly because of its complexity. You'll want to create disk cache only as a prelude to rendering with motion blur.

## Creating jiggle disk cache

When you create jiggle disk cache, you can set creation options and then create the cache, or you can immediately create the cache with the current creation options.

Usually you do not need to know the location of the files. However, if you move the scene to a different project, you must also move the jiggle cache file to the corresponding data folder of that project so that the scene plays or renders using the cache.

See "To create disk cache for jiggle deformers" on page 301.

**Note**

- By default, if you save an existing scene as a new name, Maya makes a copy of the jiggle cache file and gives it a name that corresponds with the new scene name. To save disk space, you can prevent this copy from being created. First, select File > Save Scene As > ❐. Then click the Options button to display the options window. Next, in the options window, turn on turn on Never for Copy Jiggle Disk Cache Files on Save Scene As. Finally, click the Save Scene As button.
- If you've never saved your scene (the scene is untitled), Maya creates the jiggle cache file(s) only after you save the scene.

## Disabling or deleting jiggle animation after caching

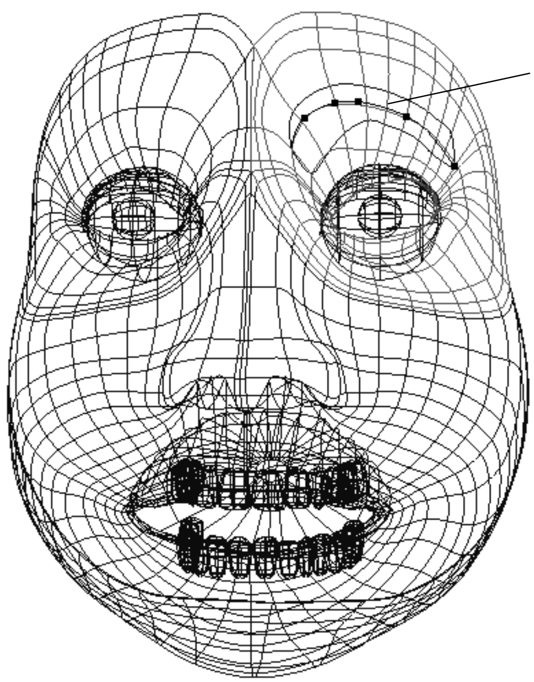
If you want to modify the jiggle animation after caching, for example, by changing the Stiffness, you must disable or delete the cache so the scene plays with the modifications. Before you render again, you must create the cache again. Deleting cache is also useful if you need to free up disk space.

Be aware that if you disable a jiggle deformer (rather than the cache), you also disable playback of the jiggle cache and cannot create the jiggle cache.

### Wire deformers

## Wire deformers

Wire deformers are like the armatures used by sculptors to shape objects. With a wire deformer, you use one or more NURBS curves to change the shape of objects. For a quick example of creating a wire deformer, see "Sculpt deformers" on page 264.




Wire deformers let you change the shapes of deformable objects with one or more NURBS curves. In character setup, wire deformers are especially useful for setting up lip and eyebrow deformations. To create further wrinkling effects, you can also use the wrinkle deformer. Wire deformers can also be useful for shaping NURBS or polygonal objects during modeling.

## Influence wires and base wires

The NURBS curves you use to create deformations are called influence wires, or simply wires. When you create a wire deformer, another curve, called a base wire, is created for each influence wire. The deformation effect provided by an influence wire is based on the difference between the influence wire and the base wire.

## Holders

Holders are curves that you can use to limit the deformation region. As with other curves, you can move, rotate, or scale holders. You can also edit a holder's shape. Moving, rotating, scaling, or editing holders can change the deformation effect.

## Wire dropoff locators

Wire dropoff locators provide a way for you to create localized deformation effects along an influence curve.

## Creating wire deformers

To create a wire deformer, you use the Wire Tool. The characteristics of the wire deformer you create depend on the Wire Tool's tool settings. By default, the Wire Tool is set to create a wire deformer without holders.

| | |
|---|---|
| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. Save a copy of the object in case you want to do further modeling later on. |

### Creating a wire deformer without holders

You can create a wire deformer without holders or with holders. Holders are curves that you can use to limit the deformation region. The workflow for creating wire deformers without holders is shorter than the workflow for creating wire deformers with holders. Further, you can limit the deformation region by adding holders later, or by using a variety of other methods. See ”To create a wire deformer without holders” on page 301.

### Creating a wire deformer with holders

Creating a wire deformer with holders is similar to creating a wire deformer without holders. The main difference is that after you select the curves you want to use as influence wires, you then select the other curves you want to use as holders for each influence wire. See "To create a wire deformer with a holder" on page 302.

## Editing wire deformation effects

After you have created a wire deformer, you can edit a wire deformer's effects.

| | |
|---|---|
| Warning | If a wire deformer does not deform an object when you manipulate the influence wire(s), the influence wire curve(s) may not have been placed close enough to the object when you created the wire deformer. You can get deformation effects by moving the base wire(s). |

### Moving, rotating, and scaling influence wires

You can create deformation effects by moving, rotating, or scaling the influence wires individually or as a group. You can move, rotate, or scale an influence wire in the same way that you would move, rotate, or scale any object in Maya.

### Moving, rotating, and scaling deformable objects

You can also create deformation effects by moving, rotating, or scaling the deformable object(s) through the influence wires. You can move, rotate, or scale a deformable object in the same way that you would move, rotate, or scale any object in Maya.

### Editing the shape of influence wires

You can create deformation effects by editing the influence wires. You edit the shape of the influence wires in the same way that you edit NURBS curves during modeling.

### Moving, rotating, and scaling base wires

You can move, rotate, or scale the base wires to create various deformation effects.

The base wires are hidden by default. However, you can select them in the Outliner, display them, and then directly manipulate them.

Note that you cannot edit the shape of the base wires, though you can edit the shape of the influence wires.

**Warning** Don't use freezeGeometry on wire deformers if you plan to move the base wire.

Otherwise, if you have a wire deformer whose freezeGeometry attribute is turned on and you move its base wire relative to the deformed surface (ore vice versa), the deformation's behavior will change the next time the scene is loaded into Maya.

## Adding influence wires

After you have created a wire deformer with one or more influence wires, you might decide you need more influence wires to get the effect you want.

## Removing influence wires

You can remove influence wires from a wire deformer. Note that removing all of a wire deformer's influence wires also removes the wire deformer node from the deformed object's history. See "To remove an influence wire" on page 304.

## Controlling the effects of crossed influence wires

If a wire deformer includes more than one influence wire, you can create interesting deformation effects by positioning the wires so that they cross. When two wires cross, you can get an additive deformation effect where the wires cross. This is because both wires are influencing some of the same points.

### Getting an additive effect where wires cross

You can control to what extent the deformation effect is the sum of the influences of both wires by editing the wire deformer's Crossing Effect attribute. The Crossing Effect attribute can have a value from 0 to 1. A value of 1 makes the total influence the sum of the influence of the two wires, creating an additive deformation effect where the wires cross. A value of 0 smooths out the deformation, so that there is no additive deformation effect where the wires cross. By default, Crossing Effect is 0, resulting in a smooth rather than an additive effect. You can edit Crossing Effect from the Channel Box or the Attribute Editor.

### Localizing the influence of crossed wires at different distances

If the wires are at different distances from the deformed object, you can control which wires influence the deformation effect more by editing the wire deformer's Local Influence attribute. The Local Influence attribute controls how localized each wire's influence is. The greater the Local Influence, the more the points closest each wire are influenced by the wire closest to them. You can edit Local Influence from the Channel Box or the Attribute Editor.

## Resetting influence wires

You can reset an influence wire so that it does not create deformation effects. By resetting an influence wire, you put the influence wire in the same position as the base wire.

## Displaying the base wire

When you create a wire deformer, another curve, called a base wire, is created for each influence wire.

## Creating wires groups that parent influence wires to base wires

After you create a wire deformer, by default the base wire will not move when you move the influence wire. Since the deformation effect is based on the relationship between the influence wire and the base wire, when you move the influence wire you get an effect that always originates from the base wire's location. This is useful for creating effects that always originate from the same place. However, you can have the base wire move with the influence wire.

## Using wire dropoff locators for localized deformation effects

You can vary the deformation effect at specific points along an influence wire by using wire dropoff locators.

Wire dropoff locators are locators that you place along an influence wire. Each locator has attributes that you can then edit to create localized deformation effects. For each influence wire, you can add as many locators as you like.

To add a wire dropoff locator, identify a curve point on the influence wire curve and then set that point as a wire dropoff locator.

### Smoothing jagged effects

In certain situations, a wire deformer can produce an undesirable jagged effect along the surface of an object.

An influence wire placed diagonally along a NURBS surface can create a jagged effect if the spacing between the surface's control vertices is too large for the value of the wire deformer's dropoff distance attribute.

In general, the spacing of a deformable object's points should be at least twice as dense as the Dropoff Distance.

### Limiting the wire deformation region

To limit the deformation region, you can use a wire deformer with holders, edit the deformer set, or prune the deformer set.

Holders are curves you can use to limit the deformation region. To create a wire deformer with holders, see "To create a wire deformer with a holder" on page 302. To add or remove holders, see "Add and remove holders" on page 351. To edit how the deformation region is limited by holders, you can move, rotate, or scale the holders. To move, rotate, or scale holders, see "Move, rotate, scale holders" on page 352. You can also edit the shape of holders. To edit the shape of holders, see "Edit the shape of holders" on page 352.

A wire deformer set includes the points of a deformable object that are influenced by a wire deformer. To limit the wire deformation region, you can edit which points are in the wire deformer set. To edit a deformer set, see "Edit wire deformer sets" on page 352.

You can also prune all of the points that are not currently being deformed from the set. Pruning the set is a quick way to limit the deformation region because you can do it as you interact with the influence wires. To prune a deformer set, see "Prune wire deformer sets" on page 352.

**Wrinkle deformers**

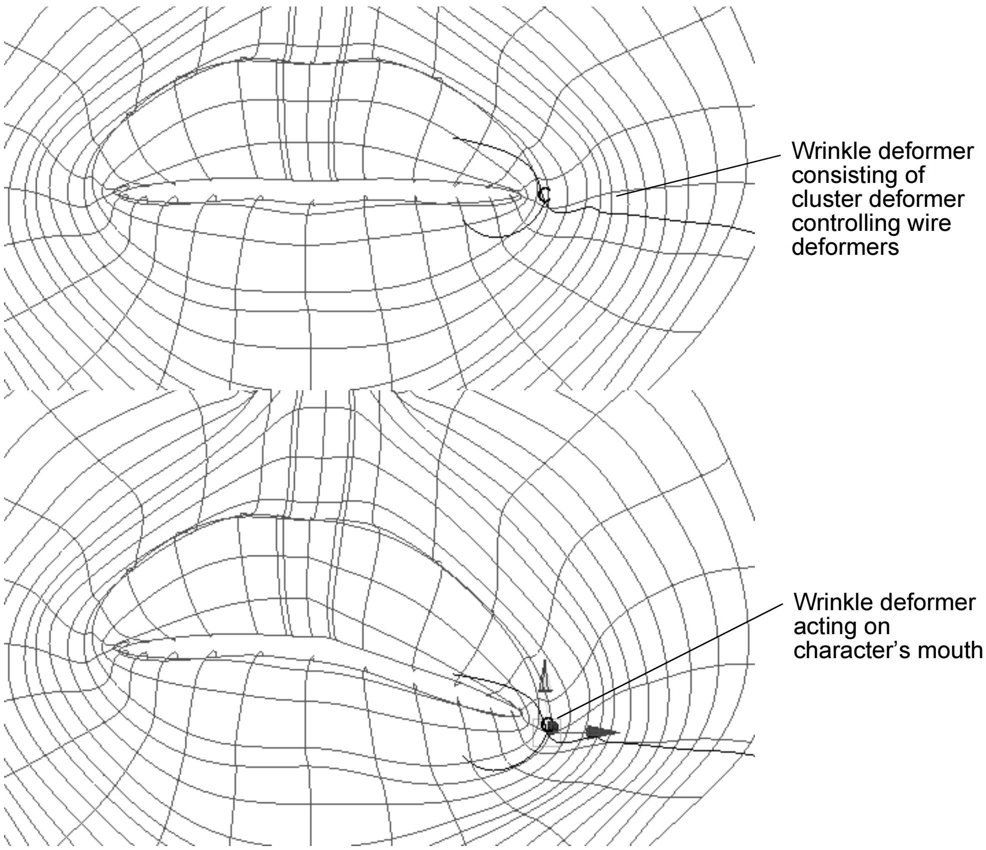
## Wrinkle deformers

Wrinkle deformers let you create detailed wrinkling effects by combining wire deformers with a cluster deformer.

A wrinkle deformer creates a cluster of wire deformers. You can create deformation effects by controlling the entire cluster of wire deformers, or by manipulating individual influence wires.

Since a wrinkle deformer is a combination of a cluster deformer and one or more wire deformers, animating a wrinkle deformer involves animating the attributes of the cluster deformer and the wrinkle deformers rather than the attributes of the wrinkle deformer.

A wrinkle deformer combines a cluster deformer with one or more wire deformers. Wrinkle deformers are useful for creating detailed wrinkling effects.




For wrinkling a single NURBS surface, you can use three types of wrinkle deformers: radial wrinkles, tangential wrinkles, and custom wrinkles. See "Wrinkle Tool Settings" on page 394.

## Creating wrinkle deformers

To create a wrinkle deformer, you use the Wrinkle Tool. The characteristics of the wrinkle deformer you create depend on the Wrinkle Tool's tool settings.

| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. You might want to save a copy of the object in case you want to do further modeling later. |
|---|---|

## Editing wrinkle deformation effects

After you create a wrinkle deformer, you can edit the deformer's effects as described in the following sections.

### Manipulating the wrinkle deformer's cluster deformer handle

The techniques for manipulating a wrinkle deformer's cluster deformer handle (the C icon) are the same as the techniques for manipulating a regular cluster deformer handle. For information on manipulating cluster deformer handles, see "Manipulating the cluster handle (C icon)" on page 251.

### Editing the wrinkle deformer's cluster deformer

The techniques for editing a wrinkle deformer's cluster deformer are the same as the techniques for editing a regular cluster deformer. For information on editing cluster deformers, see "Editing cluster deformation effects" on page 251.

### Editing the wrinkle deformer's wire deformers

The techniques for editing a wrinkle deformer's wire deformers are the same as the techniques for editing regular wire deformers. For information on editing wire deformers, see "Edit wire deformers" on page 349.

**Wrap deformers**

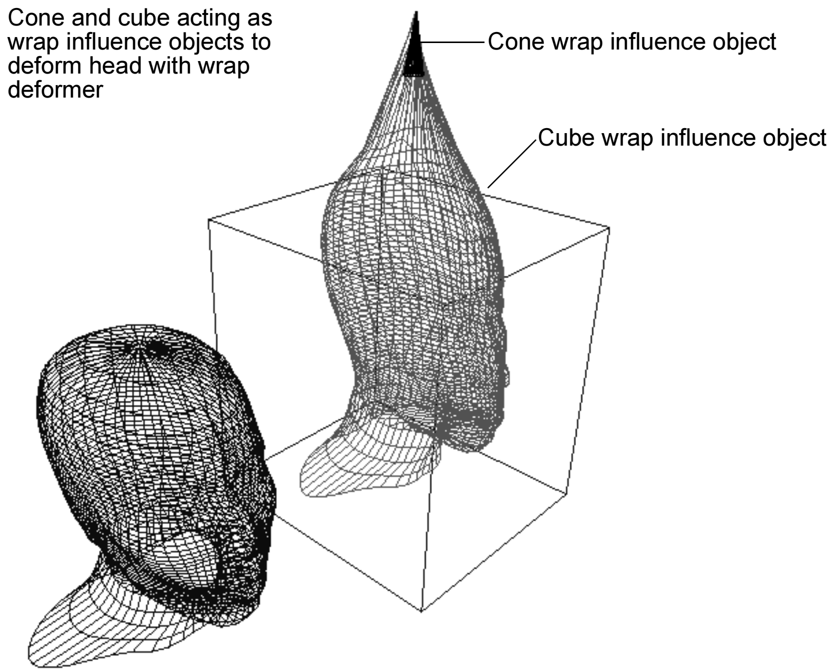
## Wrap deformers

Wrap deformers let you deform objects with NURBS surfaces, NURBS curves, or polygonal surfaces (meshes).

## Understanding wrap deformers

A wrap deformer can deform deformable objects with NURBS surfaces, NURBS curves, or polygonal surfaces (meshes).

With wrap deformers, you can shape deformable objects with NURBS or polygonal objects. The shapes of the NURBS or polygonal objects you use provide the shapes of the deformation. If you'd like to explore some examples now, see "Wrap deformers example" on page 310.




### Deformable objects

A deformable object is any object whose structure is defined by NURBS control vertices (CVs), polygonal vertices, or lattice points. NURBS curves, NURBS surfaces, polygonal surfaces (meshes), and the lattices of lattice deformers are all deformable objects.

### Wrap influence objects and wrap base objects

A wrap influence object is a NURBS or polygonal object being used by a wrap deformer (the wrap deformer algorithm node) to deform an object. The shape and the transformations of the wrap influence objects and their points provide the shape of the deformation.

When you create a wrap influence object, Maya makes a copy of the influence object and uses it as a base shape for the deformation. Any difference in position, orientation, or shape between the base shape and the wrap influence object results in a deformation of the surface being influenced by the wrap deformer.

A wrap deformer can include one or more influence objects. You'll often use several wrap influence objects to create deformation effects based on the competing influences of the objects.

You can influence one or more deformable objects with the same wrap influence objects. When you create a wrap deformer, Maya creates a wrap deformer node for each deformable object.

Note that wrap influence objects are themselves deformable objects. You can deform them with other deformers, or use them with smooth or rigid skinning.

## Creating wrap deformers

Creating a wrap deformer includes creating the objects you want to use as wrap influence objects, setting creation options, and then creating the wrap deformer.

| | |
|---|---|
| Warning | You should avoid changing the number of a deformable object's points (for example, CVs, vertices, or lattice points) after you create a deformer. Changing the number of points can lead to unexpected deformation effects. Try to be sure you are happy with the deformable object's topology before you begin using deformers. You might want to save a copy of the object in case you want to do further modeling later. You should also avoid changing the number of a wrap influence object's points (CVs or vertices) after you create a wrap deformer. |

Before you create a wrap deformer, you need to create the objects you want to use as wrap influence objects. (For more information, see the previous section.)

Unlike most other deformers, wrap deformers do not have any creation options. You can create wrap deformers immediately without having to consider the default creation options.

Note that the time required to create a wrap deformer can vary depending on the resolution of the wrap influence object(s). The resolution can also affect the deformation calculation time as you manipulate the wrap deformer.

# Editing wrap deformation effects

You can edit wrap deformation effects as described in the following sections.

## Moving, rotating, or scaling wrap influence objects

You can produce deformation effects by manipulating the wrap influence objects.

If you have a group of influence objects, note that moving, rotating, or scaling the group node will not produce deformation effects because the base objects are in the same group as the influence objects. Maya provides deformation effects based on differences between the influence objects and the base objects. To produce deformation effects, you have to manipulate the influence objects individually. If you want to manipulate all the influence objects as a group, you can create a new group that includes only the influence objects and not the base objects. Alternatively, you can remove the base objects from the already existing group. To do so, select the base object(s), and then select Edit > Unparent.

## Manipulating wrap influence object points

You can produce further deformation effects by manipulating the points of wrap influence objects. For example, you can create effects by moving one or more CVs of a NURBS wrap influence object, or by rotating or scaling several CVs.

## Moving, rotating, or scaling deformed object

You can produce deformation effects by manipulating the deformed object in the vicinity of the wrap influence object(s).

## Editing NURBS wrap influence object channels

NURBS curves or surfaces acting as wrap influence objects get two attributes added to them: the Dropoff and Wrap Samples attributes. The most convenient way to edit these channels is to use the Channel Box, but these attributes are also listed by the wrap influence object's Attribute Editor, under the Extra Attributes tab.

## Editing polygonal wrap influence object channels

Polygonal surfaces acting as wrap influence objects get three attributes added to them: the Dropoff, Smoothness, and Infl Type attributes. The most convenient way to edit these channels is with the Channel Box, but these attributes are also listed by the wrap influence object's Attribute Editor, under the Extra Attributes tab.

### Adding and removing wrap influence objects

You can add more wrap influence objects at any time after you create wrap deformers.

In removing a wrap influence object, what you remove is the object's role as a wrap influence object. Removing does not delete the object.

### Improving performance

You can improve the performance of a wrap deformer with the Max Distance creation option, channel, and attribute. For more information, see "To edit wrap deformer channels with the Channel Box" on page 353, "To edit wrap deformer attributes with the Attribute Editor" on page 354, and "Deform > Create Wrap" on page 395.

You can also improve performance by changing dependency graph evaluation performance, and by changing node behavior.

## Skinning with wrap deformers

Skinning is the process of binding deformable objects to a skeleton. Typically, the deformable objects that are bound are NURBS or polygonal surfaces (meshes). These geometry objects become the character's surface, or skin, and their shapes are influenced by the action of the skeleton's joints. Once you've built a skeleton for a character, you can skin your character by using a smooth skinning method or a rigid skinning method.

Because wrap influence objects are themselves deformable objects, you can also bind them to a skeleton by smooth or rigid skinning. In turn, these can influence the NURBS or polygonal surfaces that provide the character's skin.

In skinning with wrap deformers, you create wrap deformers for the deformable objects that you want to use for the character's skin. Then you bind the wrap influence objects to the skeleton. The result is that the skeleton's movement influences the objects being deformed by the wrap influence objects indirectly. Meanwhile, you can manipulate the wrap influence objects for more control over the deformation. This approach, skinning with wrap deformers, is called *wrap skinning*.

**Point on curve deformers**

## Point on curve deformers

You can constrain points on a NURBS curve (curve points) to locators. This is useful for deforming individual curves at specific points along the curves. By moving (translating) the locators you can change the shape of the curve without being limited to being able to move only the curve's

CVs. Also, when modeling, you can use point on curve locator constraints to connect two or more curves together so that they intersect. See ”Set point on curve deformers” on page 314 and ”Deform > Point On Curve” on page 403.

# How do I? Create deformers

## Set blend shape deformers

### To create a blend shape deformer

1 Select one or more deformable objects for target object shape(s), and then select one deformable object as the base object shape.

2 Select Deform > Create Blend Shape > ❐.

The BlendShape Options window appears.

3 Click the Basic and Advanced tabs and set the creation options.

See ”Deform > Create Blend Shape” on page 364 and ”Advanced deformer options” on page 403.

| | |
|---|---|
| **Note** | The default Front Of Chain option is only available for blend shape deformers. |

4 Do one of the following:

- Click Create to create a blend shape. The options window closes.
- Click Save to save the creation options. The options window remains open.
- Click Reset to reset the options to their defaults. The options window remains open.
- Click Close to close the Blend Shape Options window.

### To create a new blend shape using the blend shape editor

1 Select all targets, and then select the base.

You must select the base last.

2 Select Window > Animation Editors > Blend Shape.

The Blend Shape Editor appears. See ”Blend Shape Editor” on page 360.

3 In the Blend Shape Editor's menu bar, select Edit > Create Blend Shape.

The new blend shape node and slider(s) appear in the Blend Shape Editor.

#### To create blend shape deformation effects

1 Edit the blend shape deformer channels and attributes.

2 Use the Blend Shape editor (Window > Animation Editors > Blend Shape) to control the influence of the target object shapes.

For more information, see "Edit blend shape deformers" on page 316.

### Use the Blend Shape editor

#### To use the Blend Shape editor

1 Select Window > Animation Editors > Blend Shape.

The Blend Shape editor appears. The editor includes a section for each blend shape deformer in your scene.

2 Click on the name of the blend shape deformer you want to control (default name: blendShape*n*).

The editor expands to show controls for the selected blend shape deformer.

## Set lattice deformers

#### To create a lattice deformer

1 Select one or more deformable objects.

2 Select Deform > Create Lattice > ❐.

The Lattice Options window appears.

3 Click the Basic and Advanced tabs and set the creation options.

See "Deform > Create Lattice" on page 368 and "Advanced deformer options" on page 403.

4 Do one of the following:

- Click Create to create a lattice deformer. The options window closes.
- Click Save to save the creation options. The options window remains open.
- Click Reset to reset the options to their defaults. The options window remains open.
- Click Close to close the Lattice Options window.

### To create lattice deformation effects

1. Move, rotate, or scale influence lattice points.
2. Edit lattice deformer channels and attributes.

   For more information on creating and editing deformation effects, see "Lattice deformers and lattice flexors" on page 244.

## Set cluster deformers

### To create a cluster deformer

1. Select one or more deformable objects.
2. Select Deform > Create Cluster > ❐.

   The Cluster Options window appears.
3. Click the Basic and Advanced tabs and set the creation options.

   See "Deform > Create Cluster" on page 371 and "Advanced deformer options" on page 403.
4. Do one of the following:
   - Click Create to create a cluster deformer. The options window closes.
   - Click Save to save the creation options. The options window remains open.
   - Click Reset to reset the options to their defaults. The options window remains open.
   - Click Close to close the Cluster Options window.

## Set bend nonlinear deformers

### To create a bend deformer

1. Select the object(s) you want to deform.
2. Select Deform > Create Nonlinear > Bend > ❐.

   The Bend Options window appears.
3. Click the Basic and Advanced tabs and set the creation options.

   See "Deform > Create Nonlinear > Bend" on page 373 and "Advanced deformer options" on page 403.
4. Do one of the following:
   - Click Create to create a bend deformer. The options window closes.
   - Click Save to save the creation options. The options window remains open.

- Click Reset to reset the options to their defaults. The options window remains open.
- Click Close to close the Bend Options window.

#### To create bend deformation effects

1 Manipulate the bend deformer handle.
2 Edit bend deformer channels and attributes.
  For more information on creating and editing deformation effects, see ”Edit bend nonlinear deformers” on page 326.

## Set flare nonlinear deformers

#### To create a flare deformer

1 Select the object(s) you want to deform.
2 Select Deform > Create Nonlinear > Flare > ❐.
  The Create Flare Deformer Options window appears.
3 Click the Basic and Advanced tabs and set the creation options.
  See ”Deform > Create Nonlinear > Flare” on page 374 and ”Advanced deformer options” on page 403.
4 Do one of the following:
  - Click Create to create a flare deformer. The options window closes.
  - Click Save to save creation options without creating a flare deformer. The options window remains open.
  - Click Reset to reset the options to their defaults. The options window remains open.
  - Click Close to close the window.

#### To create flare deformation effects

1 Manipulate the flare deformer handle.
2 Edit flare deformer channels and attributes.
  For more information on creating and editing deformation effects, see ”Edit flare nonlinear deformers” on page 328.

## Set sine nonlinear deformers

#### To create a sine deformer

1 Select the object(s) you want to deform.

2 Select Deform > Create Nonlinear > Sine > ❐.

The Create Sine Deformer Options window appears.

3 Click the Basic and Advanced tabs and set the creation options.

See ”Deform > Create Nonlinear > Sine” on page 376 and ”Advanced deformer options” on page 403.

4 Do one of the following:

- Click Create to create a sine deformer and close the options window. The options window closes.
- Click Save to save creation options without creating a sine deformer. The options window remains open.
- Click Reset to reset the options to their defaults. The options window remains open.
- Click Close to close the window.

## Set squash nonlinear deformers

### To create a squash deformer

1 Select the object(s) you want to deform.

2 Select Deform > Create Nonlinear > Squash > ❐.

The Create Squash Deformer Options window appears.

3 Click the Basic and Advanced tabs and set the creation options.

See ”Deform > Create Nonlinear > Squash” on page 377 and ”Advanced deformer options” on page 403.

4 Do one of the following:

- Click Create to create a squash deformer. The options window closes.
- Click Save to save creation options without creating a squash deformer. The options window remains open.
- Click Reset to reset the options to their defaults. The options window remains open.
- Click Close to close the window.

## Set twist nonlinear deformers

### To create a twist deformer

1 Select the object(s) you want to deform.

2 Select Deform > Create Nonlinear > Twist > ❐.

The Create Twist Deformer Options window appears.

**3** Click the Basic and Advanced tabs and set the creation options.

See ”Deform > Create Nonlinear > Twist” on page 379 and ”Advanced deformer options” on page 403.

**4** Do one of the following:

- Click Create to create a twist deformer. The options window closes.
- Click Save to save the creation options without creating a twist deformer. The options window remains open.
- Click Reset to reset the options to their defaults. The options window remains open.
- Click Close to close the window.

*Example Twist Deformation: Modeling a Spiral staircase.*

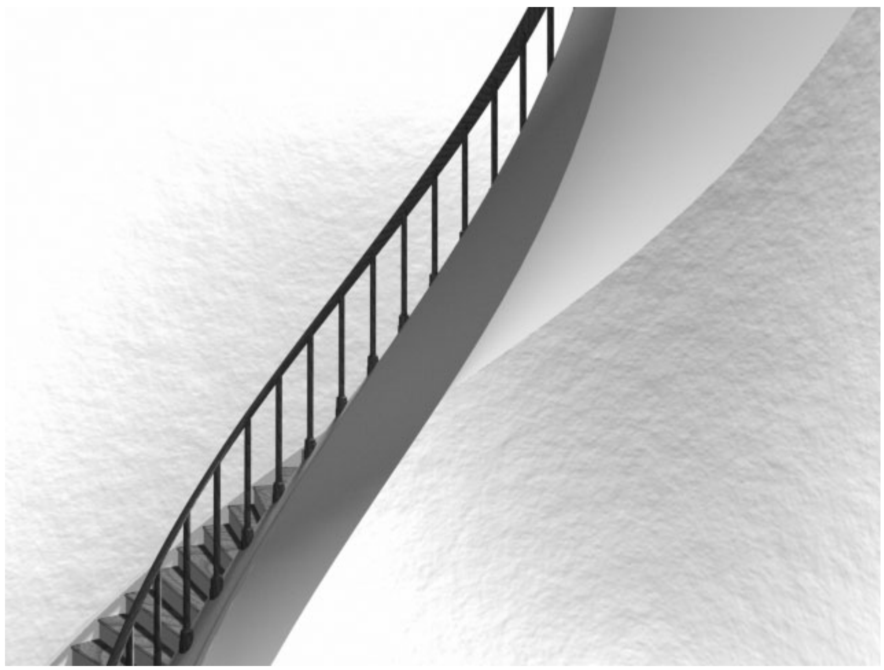


Image by Peter Miller

The foundation, rail, and moldings of the staircase’s model were shaped with the twist deformer.

## Set wave nonlinear deformers

### To create a wave deformer

**1** Select the object(s) you want to deform.

2 Select Deform > Create Nonlinear > Wave > ❒.

The Create Wave Deformer Options window appears.

3 Click the Basic and Advanced tabs and set the creation options.

See ”Deform > Create Nonlinear > Wave” on page 380 and ”Advanced deformer options” on page 403.

4 Do one of the following:

- Click Create to create a wave deformer. The options window closes.
- Click Save to save the creation options without creating a wave deformer. The options window remains open.
- Click Reset to reset the options to their defaults. The options window remains open.
- Click Close to close the window.

## Set sculpt deformers

### To create a sculpt deformer

1 Select one or more deformable objects.

2 Select Deform > Create Sculpt Deformer > ❒.

The Sculpt Options window appears.

3 Click the Basic and Advanced tabs and set the creation options.

See ”Deform > Create Sculpt Deformer” on page 382 and ”Advanced deformer options” on page 403.

4 Do one of the following:

- Click Create to create a sculpt deformer. The options window closes.
- Click Save to save the creation options without creating a sculpt deformer. The options window remains open.
- Click Reset to reset the options to their defaults. The options window remains open.
- Click Close to close the window.

### To use a NURBS surface with the sculpt deformer

1 Select the object to deform, and then the NURBS surface to use as a sculpting tool.

2 Select Deform > Create Sculpt Deformer > ❒.

3 Turn on Use Secondary Object as Sculpt Tool.

4 Click Create.

# Use the Soft Modification Tool

See "Soft Modification" on page 268 and "Deform > Soft Modification" on page 384.

### To deform objects or components with Soft Modification

1. Select the objects or components you want to deform.
2. Select the Soft Modification Tool in the Tool Box or select Deform > Soft Modification.
3. Use the manipulator to create a deformation.
4. Adjust the softMod attributes to interactively modify the deformation.

### To deform an object with Soft Modification

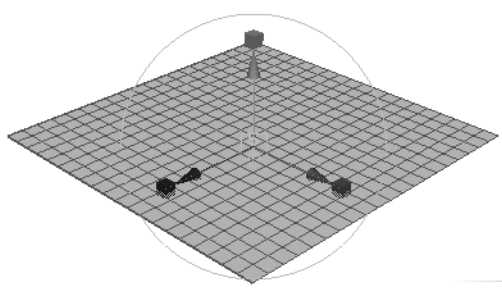
1. Select the Soft Modification Tool in the Tool Box or select Deform > Soft Modification.
2. Click the part of the object you want to modify.
3. Use the manipulator to create a deformation.
4. Adjust the softMod attributes to interactively modify the deformation.

| Tip | When you turn on Window > Settings/Preferences > Preferences > Settings > Selection > Click Drag Select, you can simply choose the Soft Modification Tool, and click-drag on the object to modify it. This lets you quickly perform several modifications on an object. See "Example 2: Modeling without history" on page 293. |
|---|---|

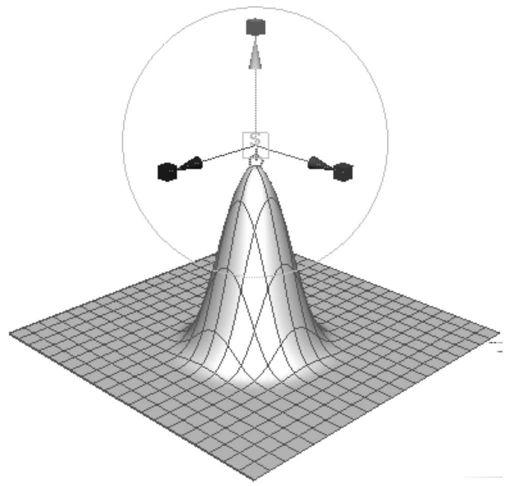
## Example 1: Modeling with history

This example shows you how to interactively deform an object using the Soft Modification Tool.

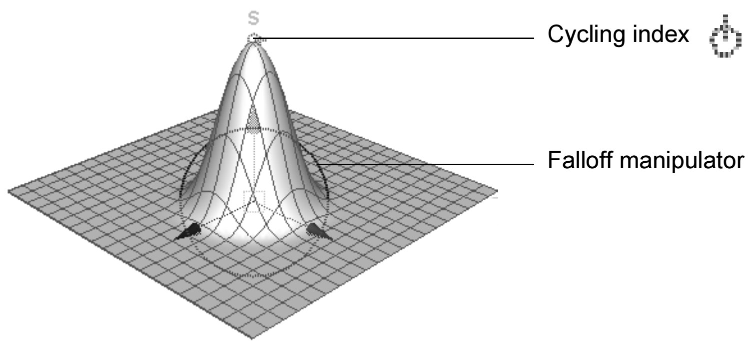
1. Select the Soft Modification Tool in the Tool Box.
2. Click the part of the surface you want to modify.



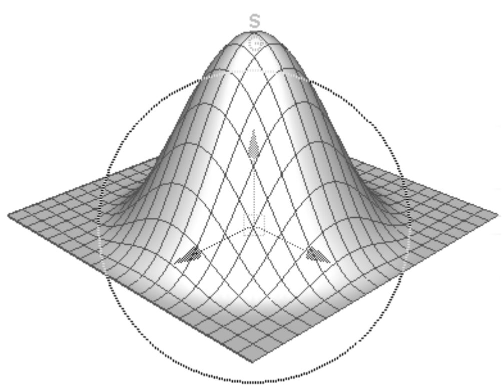
**3** Move the manipulator to create a deformation.



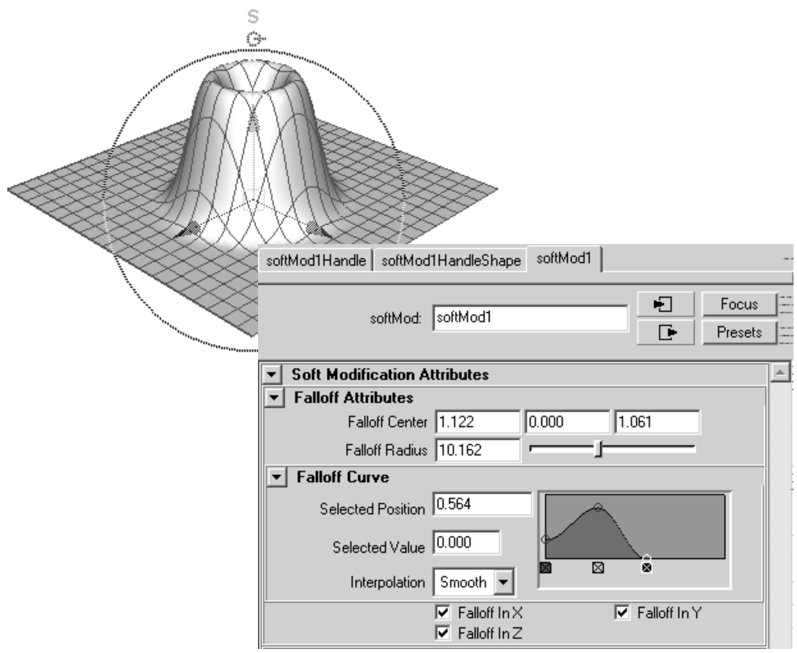
**4** Click the cycling index to display the falloff manipulator.



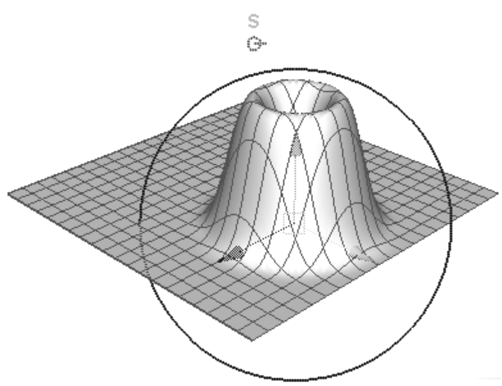

**5** Drag the falloff manipulator circle to adjust the falloff radius.




**6** Adjust the softMod node Falloff Curve to interactively adjust the falloff of the deformation.

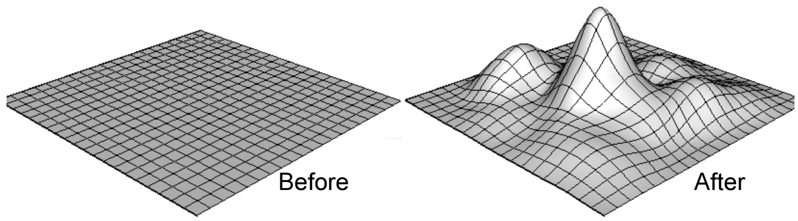



**7** Move the falloff manipulator to re-position the deformation.



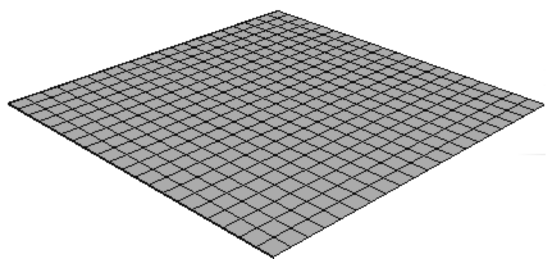

## Example 2: Modeling without history

This example shows you how to quickly model an object using the Soft Modification Tool to push and pull the surface like a piece of clay without creating deformation history.



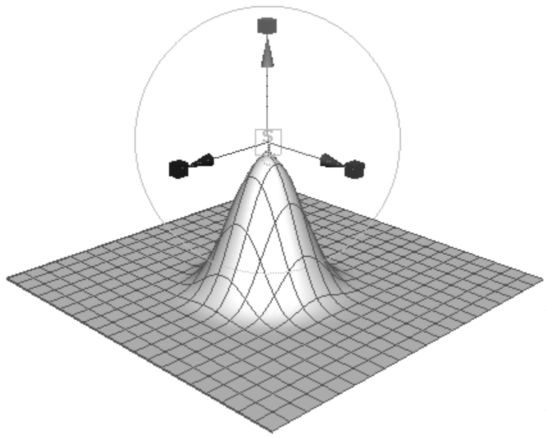

1 Select Window > Settings/Preferences > Preferences to open the Preferences window.
2 Select Selection, turn on Click Drag Select, and click Save to close the Preferences window.
3 Select the object you want to modify, and select Edit > Delete by Type > History to delete history on the object.

   By deleting history on the object, and turning off the Preserve History option (see below), the Soft Modification Tool will attempt to remove the deformation history on the object.
4 Clear the selection (so the object is no longer selected).

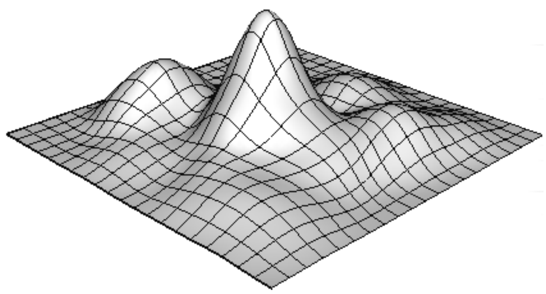


5 Select Deform > Soft Modification >❐, and set the following:
   - turn off Preserve History
   - turn off Mask Unselected
   - turn off Falloff Around Selection

   Then click Soft Modification.
6 Click and hold the left mouse button on the part of the object you want to modify, and without releasing the mouse button, drag in the direction you want to deform the object.

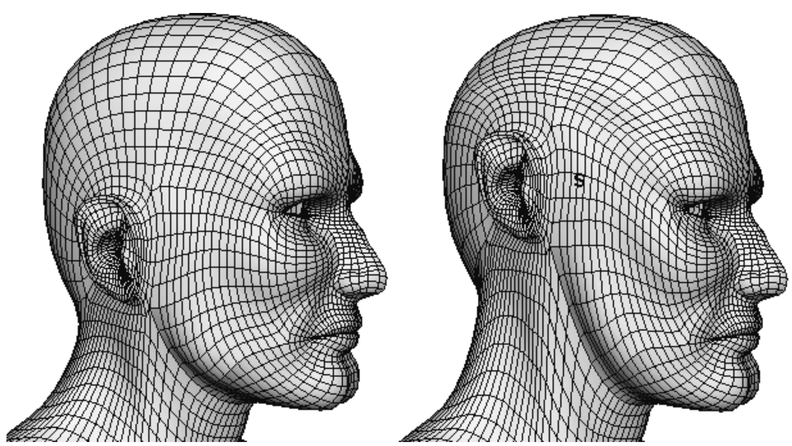


7 Continue dragging different parts of the object to create the desired shape.



## Example 3: Moving a character's ear

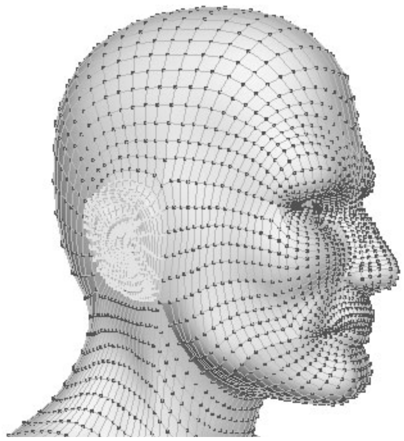
This example shows you how to move a character's ear so that the skin surrounding the ear adjusts to accommodate the new position of the ear.



Before After

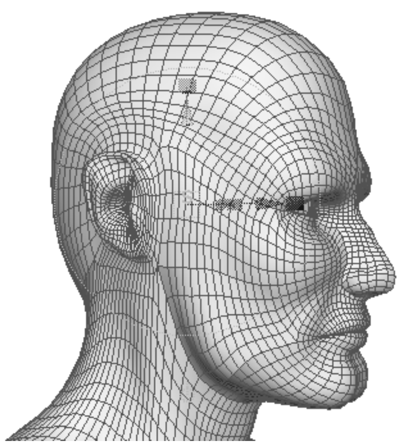
1. Select the vertices of the ear and surrounding skin that you want to move. Make sure you select the ear vertices on both sides of the head so the deformation will be symmetrical.

   (The region surrounding the selection will deform so the vertices you move blend naturally into the surrounding area.)

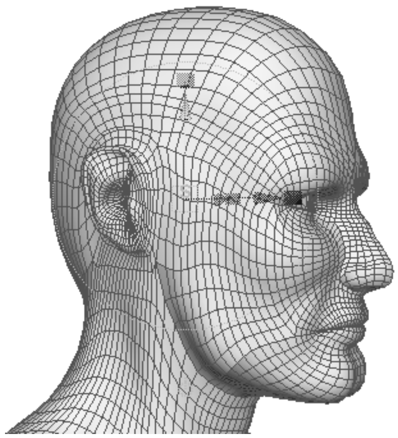


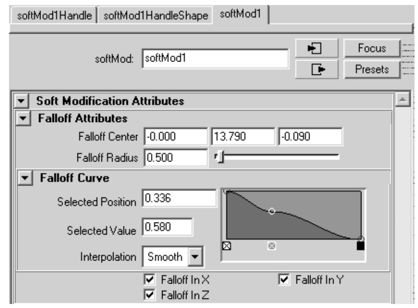
2. Select Deform > Soft Modification > ❐, and set the following:
   - turn on Preserve History
   - turn off Mask Unselected
   - turn on Falloff Around Selection

   Then click Soft Modification.
3. Use the manipulator to move the ear (for example, up).



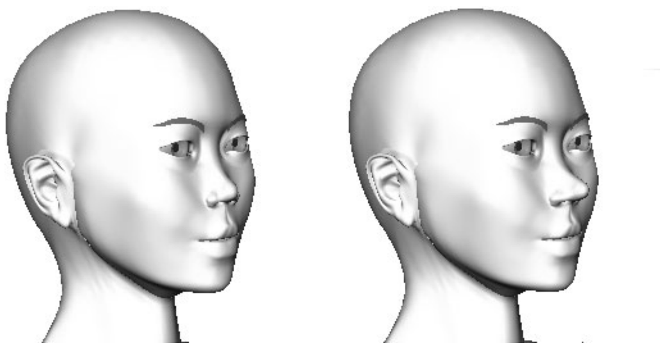
4. Adjust the softMod attributes (for example, Falloff Radius, Falloff Curve) to interactively adjust how the new ear position blends into the surrounding skin.





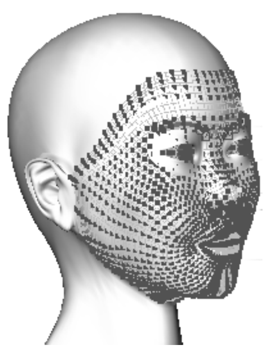

## Example 4: Symmetrically deforming a character's nose

This example shows you how to symmetrically modify the nose on a character's face using the Soft Modification deformer.



Before After

**1** Select a single vertex in the center of the character's nose.

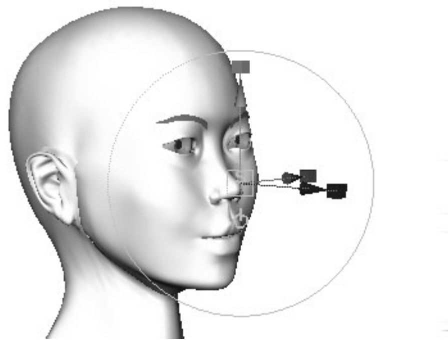


**2** Select Deform > Soft Modification > ❐, and set the following:

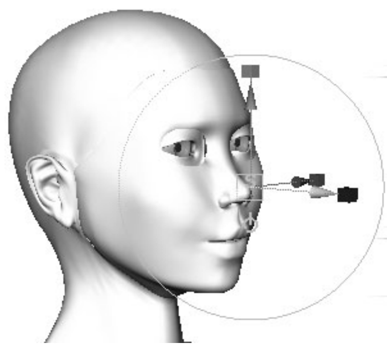
- turn on Preserve History

- turn off Mask Unselected
- turn on Falloff Around Selection

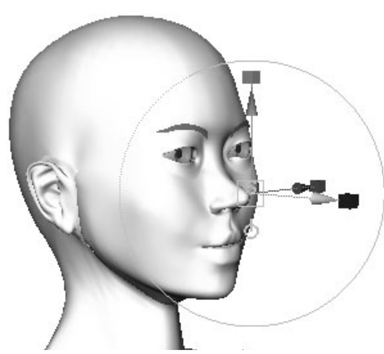
Then click Soft Modification.

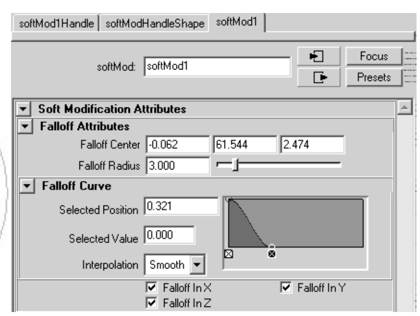


**3** Use the manipulator to create a deformation.



**4** Adjust the softMod attributes (for example, Falloff Radius, Falloff Curve) to interactively adjust the effect of the deformation (for example, to make the new nose shape blend smoothly into the rest of the face).





# Set jiggle deformers

### To create jiggle deformers

1. Select the points or entire object you want to jiggle.
2. Select Deform > Create Jiggle Deformer > ❐.

   The Jiggle Deformer Options window appears.
3. Click the Basic and Advanced tabs and set the creation options.

   See "Deform > Create Jiggle Deformer" on page 387 and "Advanced deformer options" on page 403.
4. Do one of the following:
   - Click Create to create a jiggle deformer. The options window closes.
   - Click Save to save the creation options without creating a jiggle deformer. The options window remains open.
   - Click Reset to reset the options to their defaults. The options window remains open.
   - Click Close to close the window.
5. After you play the animation to check the results, tune the jiggle with the Paint Jiggle Weights Tool. See the next procedure.

### To paint jiggle weights

1. Select the surface with the jiggle deformer you want to paint weights on.
2. Switch to smooth shading mode by selecting Shading > Smooth Shade All or by pressing the default hotkey 5.
3. Select Deform > Paint Jiggle Weights Tool > ❐.

   The Paint Jiggle Weights Tool Settings appear.

| Note | The Paint Jiggle Weights Tool automatically detects jiggle deformers on the surface. |
|---|---|

4. Check that Color Feedback is turned on in the Display section. Color feedback helps you identify the weights on the surface by representing them as grayscale values (smaller values are darker, larger values are lighter).

| Tip | You can use the default hotkey Alt+c (IRIX, Linux, and Windows) or Option+c (Mac OS X) to turn Color Feedback on and off outside the Tools Settings Editor. |
|---|---|

5 Select the jiggle deformer you want to paint.

6 In the Paint Attributes section of the Tool Settings window, click the jiggle*n*.weights button and select the appropriate jiggle weights name from the pop-up menu.

   You can only paint weights on one jiggle deformer at a time. If you select more than one jiggle deformer, you can only paint weights on the active one (the one that provides color feedback).

   If the surface has only one jiggle deformer, you can select the surface alone.

| Tip | If you are painting on a single surface, you can skip step 6 and select the jiggle deformer and weight attribute without opening the Tool Settings window by right-clicking the surface and selecting the appropriate jiggle weights name from the Paint command submenu. |
|---|---|

7 Select a brush, paint operation, and value and define other settings as required.

8 Drag the brush across the deformer where you want to change the weights.

Tip: Using hotkeys, you can quickly pick weight values from one jiggle deformer and paint them on another jiggle deformer or the same one. Do the following:

- Select the jiggle deformer with the weight values you want to pick.
- Hold down the Pick Color Mode hotkey (default hotkey: /), click on the area of the jiggle deformer with the weight you want to pick, then release the hotkey.
- If you are painting the picked weight on the same deformer, drag the brush across the deformer.
- If you are painting the picked weight on another jiggle deformer, select that deformer, then drag the brush across it.

#### To create disk cache for jiggle deformers

1 Select Deform > Create Jiggle Disk Cache > ❐.

2 The Jiggle Disk Cache Options window appears.

3 Click the Basic and Advanced tabs and set the creation options.

See ”Deform > Create Jiggle Disk Cache” on page 388 and ”Advanced deformer options” on page 403.

4 Do one of the following:

- Click Create to create the jiggle cache. The options window closes.

Maya creates the jiggle cache and creates a permanent jiggle cache file for each jiggle deformer in the scene. The file or files are in the current project’s *data* folder, and are named *scene*_jiggle*n*.mcj.

- Click Save to save the creation options without creating the jiggle cache. The options window remains open.
- Click Reset to reset the options to their defaults. The options window remains open.
- Click Close to close the window.

## Set wire deformers

#### To specify wire tool settings

1 Select Deform > Wire Tool > ❐.

The Tool Settings window appears. See ”Deform > Wire Tool” on page 389.

2 Set the tool settings.

3 Do one of the following:

- Click Reset Tool to reset to the default tool settings.
- Click Close to close the Tool Settings window.

**Note** You can change the tool settings when you are using the Wire Tool by selecting Window > Settings/Preferences > Tool Settings.

#### To create a wire deformer without holders

1 Set the Wire Tool’s Holders setting to off (the default is off).

2 Create the curve(s) you want to use as influence wire(s). For best results, place them on or near the deformable object(s).

3 Select Deform > Wire Tool.

The cursor changes to a crosshair icon, and the Wire Tool icon is appears in the Tool Box. You are now ready to use the Wire Tool to create a wire deformer with the Wire Tool's current tool settings.

The helpline displays information to lead you through the process of creating a wire deformer.

4 Select the object(s) you want to deform, and press Enter (IRIX, Linux and Windows) or Return (Mac OS X) key.

5 Select all of the curves you want to use as influence wires.

If the only curves on the object's surface are the curves you want to use as influence wires, drag the cursor over all the curves on the object. The Wire Tool automatically selects only the curves.

6 Press the Enter or Return key.

A wire deformer is created based on the Wire Tool's settings. The curves you selected are now influence wires that you can use to deform the object(s) you selected.

A base wire is created for each influence wire. The base wire(s) are listed in the Outliner. By default, they are not shown in the scene, but they do influence the deformation effect. The wire node calculates the deformation effect based on differences between each influence wire and its base wire.

A deformer set is created. The deformer set includes all the deformable object's points that can be influenced by the wire deformer.

For more information on creating and editing deformation effects, see "Editing wire deformation effects" on page 274.

#### To create deformation effects

1. Move, rotate, or scale the influence wire(s).
2. Edit the wire deformer's channels.

#### To create a wire deformer with a holder

1. Set the Wire Tool's Holders setting to on (the default is off).
2. On or near the deformable object(s), create the curve(s) you want to use as influence wire(s).
3. On or near the deformable object(s), create the curve you want to use as a holder.
4. Select Deform > Wire Tool.

   The cursor changes to a cross-hair, and the Wire Tool icon appears in the Tool Box. You are now ready to use the Wire Tool to create a wire deformer with the Wire Tool's current tool settings.

Note that the Maya helpline displays information to lead you through the process of creating a wire deformer.

5 Select the object(s) you want to deform, and press the Enter (IRIX, Linux, and Windows) or Return (Mac OS X) key.
6 Select a curve that you want to use as an influence wire, and press the Enter or Return key.
7 Select a holder curve and press Enter or Return. An influence wire can only have one holder.
8 Clear the selection list by selecting empty space, and then press Enter or Return.

| Note | You can assign a holder to more than one influence wire. To do so, pick a curve that will be an influence wire, press Enter or Return, then select the curve that will be a holder, and press Enter or Return again. Pick the next curve that will be an influence wire, press Enter or Return, then select the same holder curve, and press Enter or Return again. Continue this process for each influence wire curve that will be assigned the holder curve. |
|---|---|

9 When you are ready to create the wire deformer, clear the selection list by selecting empty space, and then press Enter or Return.

A wire deformer is created based on the Wire Tool's tool settings. The curves you selected are now influence wires that you can use to deform the object(s) you selected.

A base wire is created for each influence wire. The base wire(s) are listed in the Outliner. By default, they are not shown in the scene, but they do influence the deformation effect. The wire node calculates the deformation effect based on differences between each influence wire and its base wire.

A deformer set is created. The deformer set includes all the deformable object's points that can be influenced by the wire deformer.

### To create deformation effects

1 Move, rotate, or scale the influence wire(s).
2 Edit the wire deformer's channels.

For more information on creating and editing deformation effects, see the next section, "Editing wire deformation effects" on page 274.

### To add an influence wire

1 Select the curves you want to add to the deformer.

2. Shift-click on any wire in the deformer to select the deformer to which you want add the curves as influence wires.
3. Select Deform > Edit Wire > Add.

   The selected curves become influence wires for the wire deformer.

#### To remove an influence wire

1. Select the curves that you want to remove as influence wires.
2. Select Deform > Edit Wire > Remove.

   The selected curves are no longer influence wires.

#### To reset influence wires

1. Select the influence wire(s) you want to reset.
2. Select Deform > Edit Wire > Reset.

#### To display the base wire

1. Select the influence wire.
2. Select Deform > Edit Wire > Show Base Wire.

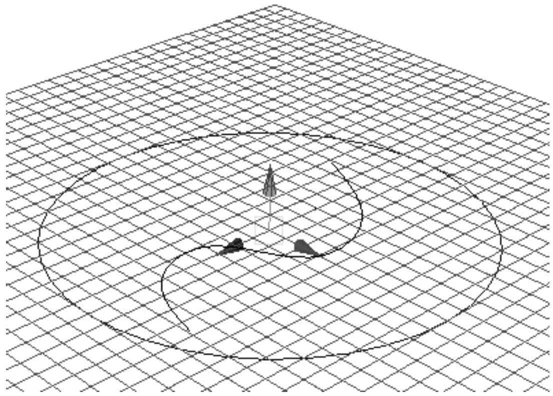
#### To create a wire group

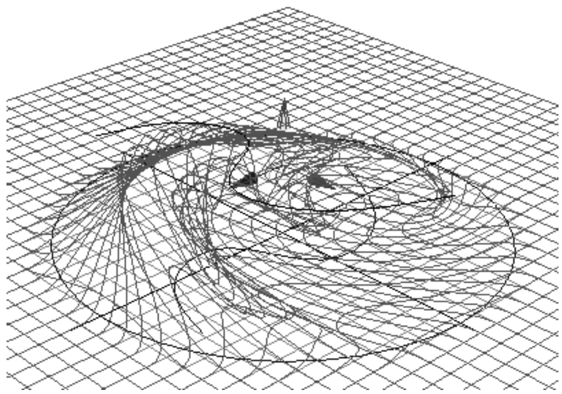
1. Select the influence wire.
2. Select Deform > Edit Wire > Parent Base Wire.

A wires group is created that includes the influence wire and the base wire. The wires group is named after the influence wire and listed in the Outliner.

## Wire deformer quick start

This section shows you how to create a typical wire deformer as quickly as possible.





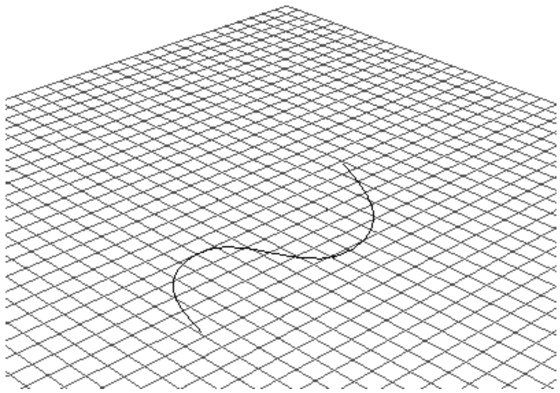
In this example, you will deform a surface with an S-shaped curve, limiting the deformation region with a circle.

### Create a NURBS plane

Create a NURBS plane with Width 40, Length 40, Patches U 40, Patches V 40, and Surface Degree Cubic.

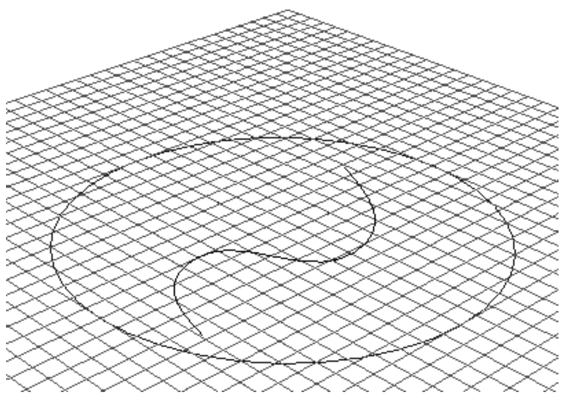
### Create a curve for an influence wire

Draw an S-shaped curve on the center area of the plane. Use the CV Curve Tool or EP Curve Tool.



### Create a circle for limiting the deformation region

- Create a circle that surrounds the S-shaped curve on the plane.



You'll use the circle to limit the deformation region. A curve that limits the deformation region is called a *holder*.

### Create the wire deformer

1 Select Deform > Wire Tool > ❐.

2 In the Tool Settings window, turn Holders on (default is off).

3 Close the Tool Settings window.

In the workspace, note the cursor is now cross-shaped, indicating you are using the Wire Tool.

Be sure you do the next four steps in exactly the following order.

4 Select the plane and press Enter (IRIX, Linux, and Windows) or Return (Mac OS X).

5 Select the S-shaped curve, and press Enter or Return.

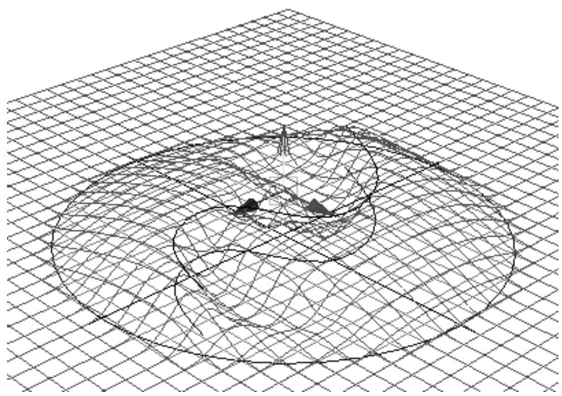
6 Select the circle, and press Enter or Return.
7 Finally, select nothing in the workspace, and press Enter or Return.

Selecting nothing tells Maya you are done selecting holders, and tells it to create the wire deformer.

Maya now creates a wire deformer. You can now manipulate the S-shaped curve to create deformation effects. Note that only the region within the circle can be affected.

### Create deformations

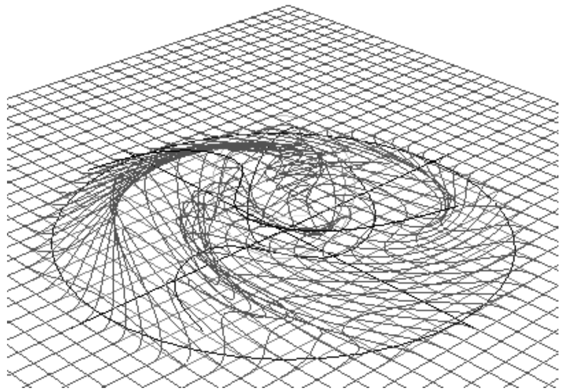
1 Select the S-shaped curve.
2 Move the S-shaped curve up.



The plane deforms upwards as if attracted to the S-curve.

The wire deformer uses the S-shaped curve to influence the shape of the deformation, so this curve is called an influence wire. Meanwhile, because the circle is limiting the deformation region, it’s called a holder.

3 Rotate the S-shaped curve to get a swirling “chakra” deformation effect.



# Set wrinkle deformers

**To create a wrinkle deformer**

1. Select one or more deformable objects. Typically, the deformable object is a NURBS surface.
2. Select Deform > Wrinkle Tool > ❐. See ”Deform > Wrinkle Tool” on page 394.

   The Tool Settings window appears.
3. Set the Wrinkle Options.
4. Do one of the following:
   - Click Reset Tool to reset to default options. The window remains open.
   - Click Close to close the Tool Settings window.

| | |
|---|---|
| **Note** | You can change the tool settings whenever you are using the Wire Tool by selecting Windows > General Editors > Tool Settings. |

   A UV region of the surface is highlighted, allowing you to shape a wire cluster for deforming the surface.
5. Using the middle mouse button, shape the UV region. Scale it using the circle in the middle of each side, rotate it using the corners, and move it using the dot in the middle of the UV region.
6. Press Enter (IRIX, Linux, and Windows) or Return (Mac OS X) when the UV region fits the area of the deformable object.

   The “C” icon is the wrinkle deformer’s cluster deformer handle.

#### To create wrinkle deformation effects

1 Move, rotate, or scale the cluster deformer handle (the “C” icon).

2 Move, rotate, or scale the influence wires.

For more information on creating and editing deformation effects, see ”Editing wrinkle deformation effects” on page 279.

## Set wrap deformers

#### To create objects to use as wrap influence objects

1 Model one or more NURBS surfaces, NURBS curves, or polygonal surfaces that you want to use as wrap influence objects.

Note that the shape and distribution of CVs or vertices can affect the wrap deformation effect. Typically, you would want to have fewer points (CVs, for example) in the influence object(s) than in the objects you want to deform.

2 Position the object or objects that you want to use as wrap influence objects. Place them around the object(s) you want to deform so that they can influence the object(s).

3 If you are going to use more than one object as a wrap influence object, group those objects together now. You must group all those objects together before you create the wrap deformer. However, note that you can also add wrap influence objects after you have created a wrap deformer. See ”Adding and removing wrap influence objects” on page 283.

Note If you want to render a wrap influence object, be sure that you first turn on the object’s Primary Visibility (in the Attribute Editor’s Render Stats section). When you tell Maya to use some object as a wrap influence object, Maya turns off the Primary Visibility attribute because typically you would not want to render a wrap influence object. However, in some situations, you might want to render the wrap influence object. For example, you might use a cloth garment as a wrap influence object, and then wish to render both the action of the garment and the wrap deformer’s effects.

#### To create a wrap deformer

1 Select the object(s) you want to deform.

2 Select the previously created object or group of objects you want to use as wrap influence objects.

For more information on creating wrap influence objects, see ”To create objects to use as wrap influence objects” on page 309.

3 Select Deform > Create Wrap > ❐.

The Create Wrap Deformer Options window appears.

4 Click the Basic and Advanced tabs to set the creation options.

See ”Deform > Create Wrap” on page 395 and ”Advanced deformer options” on page 403.

5 Do one of the following:

- Click Create to create a wrap deformer. The options closes.

Maya creates a wrap deformer node for each object you want to deform. Maya also creates wrap base objects for each wrap influence object. The Outliner lists the wrap base objects, which are hidden by default. Note that if you are using more than one influence object and have therefore grouped them together, the base objects are placed in the same group as the influence objects.

The creation time can vary, depending on the number and resolution of the deformable objects and wrap influence objects.

- Click Save to save creation options without creating a wrap deformer. The options window remains open.
- Click Reset to reset to default creation options. The opens window remains open.
- Click Close to close the window.

**To create wrap deformation effects**

1 Move, rotate, or scale the wrap influence object(s).

2 Move the points of the wrap influence object(s).

3 Edit channels added to the wrap influence objects, and edit the channels of the wrap deformer(s).

## Wrap deformers example

### Deforming high-res sphere with low-res sphere

In this short example, you will use a low-resolution sphere to deform a high-resolution sphere.

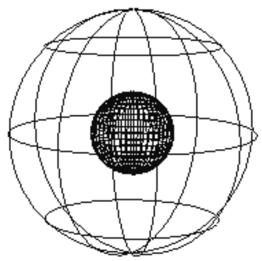
**To create high-res sphere**

Create a NURBS sphere with the default options, except set Sections to 40 and Spans to 20.

### To create low-res sphere

Create a NURBS sphere with the default options, except set Radius to 3.

The low-res sphere surrounds the high-res sphere, whose resolution is five times that of the low-res sphere.



### To create wrap deformer

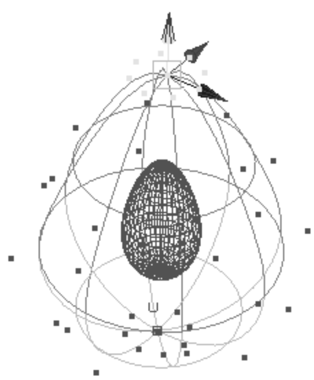
**1** Select the high-res sphere, and then select the low-res sphere.

**2** Select Deform > Create Wrap.

The low-res sphere is now a wrap influence object. To find out more about creating wrap deformers, see "Creating wrap deformers" on page 281.

### To create wrap deformation effects

**1** Select some of the CVs of the low-res sphere and move them.

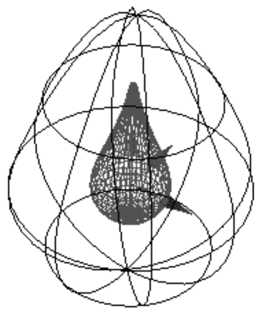


The high-res sphere deforms in response to the changes to the other sphere's CVs.

**2** Switch to object selection mode, keeping the low-res sphere selected.

In the Channel Box, note the sphere's two new channels: Dropoff and Wrap Samples.

**3** Set Dropoff to 20.



The deformation becomes more pronounced.

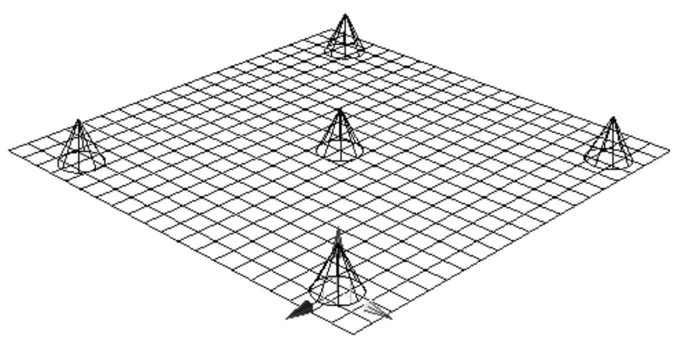
If you'd like to experiment further with Dropoff and Wrap Samples, see "Editing NURBS wrap influence object channels" on page 282.

#### To create a plane

Create a NURBS plane with all the default options, except set Width to 20, Length to 20, U Patches to 20, and V Patches to 20.

#### To create cones

**1** Create five NURBS cones with all default options. Arrange them on the plane as follows:



**2** Group all the cones together.

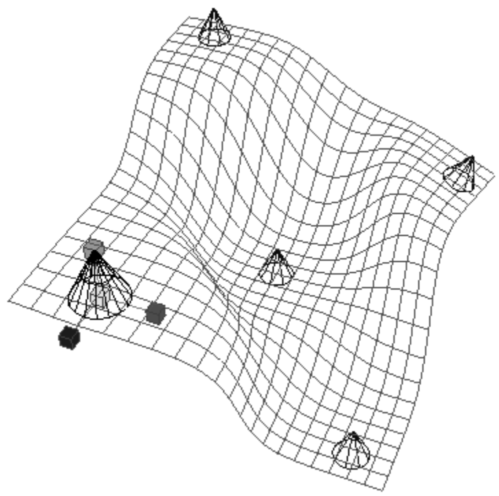
#### To create a wrap deformer

**1** Select the plane, then select the cones group.

**2** Select Deform > Create Wrap.

### To deform plane by moving cones

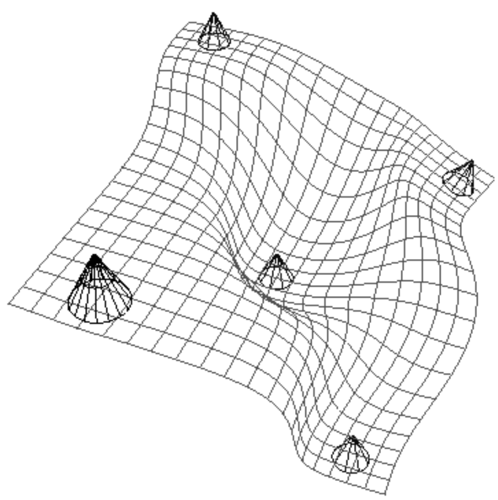
Move, rotate, or scale the cones to deform the plane.



You can create a wide variety of deformation effects just by manipulating the cones.

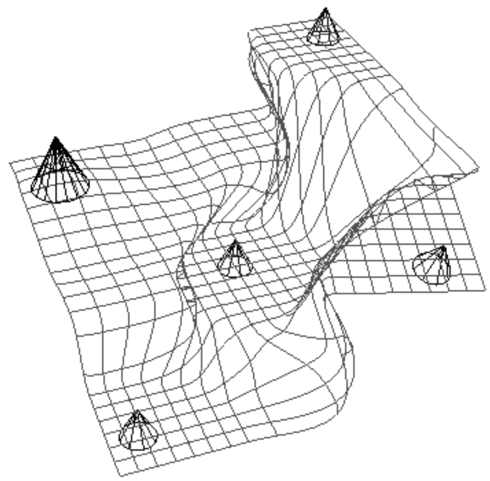
### To edit deformation effects

**1** Experiment with each cone's Dropoff channel.



For more information on the Dropoff and Wrap Samples channels, see "Editing NURBS wrap influence object channels" on page 282.

**2** In the Channel Box, note that the wrap deformer node (wrap1) is listed in the OUTPUTS for the cones and in the INPUTS for the plane. Select the wrap deformer node. Experiment with wrap1's Weight Threshold channel, which can provide sharper or smoother deformation effects.



Weight Threshold channel set to 0.5.

For more information on the wrap deformer channels, see "Wrap deformer nodes" on page 423.

Note that you can also move the plane away from or into the influence of the cones. You could create an animation in which the plane goes through a deformation when it gets close to the cones.

## Set point on curve deformers

### To create a point on curve locator

**1** Create a NURBS curve.

**2** To select a curve point on the curve, right-click the curve and select Curve Point from the marking menu.

**3** Click on the curve at where you would like to create the point on curve locator constraint. The curve point is displayed as a small yellow box.

**4** Drag along the curve to adjust the point's position on the curve.

As you drag, you move the curve point. The curve point's position is defined in terms of the curve's U parameter.

**5** Select Deform > Point On Curve > ❐.

The Point On Curve Options window appears. See "Deform > Point On Curve" on page 403.

6 Set the options you want.

7 Do one of the following:

- Click Create to create a point on curve locator constraint. The options window closes.

  Maya creates a locator at the curve point with the default point on curve options. The curve point is now constrained to the locator.

  For each curve, Maya creates a least squares modifier node (default name: leastSquaresModifier*n*).

- Click Save to save the constraint options. The options window remains open.
- Click Reset to reset to the default constraint options. The options window remains open.
- Click Close to close the Point On Curve Options window.

## Paint deformer set membership

Using the Paint Set Membership Tool, you can modify which of a deformable object's points (for example, CVs or vertices) belong to multiple deformer sets by painting the points you want added to, transferred to, or removed from the set.

### To modify which vertices belong to a set

1 Select a deformed or a skinned object.

2 Select Shading > Smooth Shade All (default hotkey: 5) to switch to smooth shading mode.

3 Select Deform > Paint Set Membership Tool > ❐.

  The Paint Set Membership Tool Settings appears. See "Deform > Paint Set Membership Tool" on page 401.

4 Set the tool settings.

5 Select the set you want to modify as follows:

- In the Tool Settings editor (Deform > Paint Set Membership Tool > ❐), click the SetMembership tab. In the Set Membership section, click the set in the Select Set To Modify box. The selected set name appears in the Set To Modify box.
- Use the Pick Color Mode hotkey (default hotkey: /) to select the set on the surface. Hold down the hotkey, click on the set you want to paint (click anywhere on the colored area), then release the hotkey.

6 Drag over the CVs or vertices you want to add to, transfer to, or remove from the selected set.

## Prune deformer set membership

1. Select the deformable objects whose currently unaffected points you want to prune from the deformation.
2. Select Deform > Prune Membership, and from the cascading menu select the deformer whose set you want to prune.

| | |
|---|---|
| **Note** | Prune Membership only removes points from the deformation if they are currently not affected by the deformer. Therefore, if you try to prune the membership of an object whose points are all controlled by the deformer, you will receive the error message "No components could be pruned." |

# Edit deformation effects

## Edit blend shape deformers

The Blend Shape editor provides you with controls for all of the blend shape deformers in your scene.

### To edit Blend Shape channels with the Channel Box

1. Select a blend shape deformer node (default name: blendShape*n*).

   One quick way to select the blend shape deformer node is to select the object being deformed, and then select the blend shape deformer node in its history from the Channel Box (under INPUTS).

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. In the Channel Box, click on a channel name with the left mouse button.
3. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key will give you finer control, and pressing the Shift key will give you less control.

### To edit Blend Shape attributes with the Attribute Editor

1. Select the blend shape deformer node (default name: blendShape*n*).
2. Open the Attribute Editor by selecting Window > Attribute Editor or use the default shortcut Ctrl+a.
3. Edit the attributes.

See "blendShape" on page 406.

### To delete the target object before creating a blend shape

- Turn the Delete Targets creation option on.
  See "Deform > Create Blend Shape" on page 364.

### To delete the target object after creating blend shape

- Select and delete the object in the workspace or Outliner.

### To key all target sliders

1. In the Blend Shape editor, adjust the sliders to create the desired blend shape.
2. In the Time Slider, click the frame where you want to set keys.
3. In the Blend Shape editor, click the Key All button.
   Maya sets keys for all the target sliders in the blend shape deformer.

### To key the maximum influence of one target slider

1. In the Time Slider, click the frame where you want to set the key.
2. In the Blend Shape editor, set the target slider to 1.
3. Click the Key button below the target slider.
   Maya sets a key for that target slider only, ignoring the possible blending influences of the other target sliders.

### To save a blend shape as a new target

1. Set the target sliders to deform the base object.
2. Select the base.
3. Click Add in the Blend Shape editor.
   Maya creates a new target at the same location as the base. A slider for the target appears in the Blend Shape editor.
   Move the new target away from the base. If in local mode, you can modify the target's shape, for instance, by transforming its CVs or vertices. Use the new target slider to deform the base to the target.

### To select the blend shape node

Click the Select button for the blend shape node in the Blend Shape editor.

### To add a target shape for blend shape deformers

1. Select one or more deformable objects as new target object shape(s), and then a blend shape deformer's base object shape.

2 Select Deform > Edit Blend Shape > Add > ❐.

The Add Blend Shape Target Options window appears. See ”Deform > Edit Blend Shape > Add” on page 366.

3 Set the options.

4 Do one of the following:

- Click Apply if you want to add the selected target object shapes now. The options window closes.
- Click Save if you want to save the options you’ve specified. The options window remains open.
- Click Reset to reset to the default options. The options window remains open.
- Click Close to close the Add Blend Shape Target Options window.

### To remove a target shape

1 Select the target objects you want to remove.

2 Select Deform > Edit Blend Shape > Remove > ❐.

The Remove Blend Shape Target Options window appears. See ”Deform > Edit Blend Shape > Remove” on page 367.

3 Set the options.

4 Do one of the following:

- Click Apply if you want to remove the selected target object shapes now. The options window closes.
- Click Save if you want to save the options you’ve specified. The options window remains open.
- Click Reset to reset to the default options. The options window remains open.
- Click Close to close the Remove Blend Shape Target Options window.

### To swap two target shapes

1 Select two target objects whose order you want to swap.

2 Select Deform > Edit Blend Shape > Swap > ❐.

The Swap Blend Shape Target Options window appears. See ”Deform > Edit Blend Shape > Swap” on page 368.

3 Set the options.

4 Do one of the following:

- Click Apply if you want to swap the selected target object shapes now. The options window closes.

- Click Save if you want to save the options you've specified. The options window remains open.
- Click Reset to reset to the default options. The options window remains open.
- Click Close to close the Swap Blend Shape Target Options window.

#### To delete a blend shape deformer

1. Select the blend shape deformer node.
2. Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

   The selected deformer nodes are all deleted. However, the base object still has the tweak node as an input node, so any tweaks you might have made are preserved.

## Edit lattice deformers

#### To move, rotate, or scale the influence lattice

1. Select the lattice deformer handle node (default name: ffd*n*Lattice).
2. Move (translate), rotate, or scale the handle to change the effect of the deformation.
3. Move or rotate the handle pivot point by pressing the Insert (IRIX, Linux, and Windows) or Home (Mac OS X) key, moving the pivot point, and then pressing the Insert or Home key again.

   Remember that you can access the deformer handle's local axes (Display > Component Display > Local Rotation Axes), it's rotate and scale pivots (Display > Component Display > Rotate Pivots or Scale Pivots) and it's selection handle (Display > Component Display > Selection Handles).

#### To edit by moving, rotating, or scaling lattice points

1. Select the lattice deformer handle node (default name: ffd*n*Lattice).
2. Go into component mode by clicking the Select By Component Type button in the Maya toolbar.
3. Select lattice points.
4. Move (translate), rotate, or scale the points to change the effect of the deformation.

#### To edit lattice channels with the Channel Box

1. Select a lattice deformer node (default name: ffd*n*).

One quick way to select the lattice deformer node is to select the object being deformed, and then select the lattice deformer node in its history from the Channel Box (under INPUTS).

Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).

2. In the Channel Box, the following channels are listed by default:
3. Click on a channel name with the left mouse button.
4. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

### To edit lattice attributes with the Attribute Editor

1. Select the lattice deformer node (default name: ffd*n*).
2. Open the Attribute Editor by selecting Window > Attribute Editor The default shortcut is Ctrl+a.
3. Edit the attributes.

   See "Lattice deformer nodes" on page 407.

### To reset the lattice

1. Select the lattice.
2. Select Deform > Edit Lattice > Reset Lattice.

   See "Deform > Edit Lattice > Reset Lattice" on page 371.

### To reset lattice points after tweaking

1. Select the lattice.
2. Select Deform > Edit Lattice > Remove Lattice Tweaks.

   See "Deform > Edit Lattice > Remove Lattice Tweaks" on page 371.

### To prune lattice deformer set membership

1. Select deformable objects whose currently unaffected points you want to prune from the deformation.
2. Select Deform > Prune Membership > Lattice.

   Maya removes the deformable object's points currently unaffected points from the lattice.

> **Note** Prune Membership only removes points from the deformation if they are currently not affected by the deformer. Therefore, if you try to prune the membership of an object whose points are all controlled by the deformer, you will receive the error message "No components could be pruned."

### To select the lattice shape display

1. Select the lattice deformer.
2. Select Display > Component Display > Lattice Shape.

   The lattice deformer selects between displaying its lattices and the "L" icon.

### To turn on or off lattice shape display

1. Select the lattice deformer.
2. Select Display > Component Display > Lattice Points.

### To show all lattice deformers

- Select Display > Show > Show Deformers > Lattices.

### To hide all lattice deformers

- Select Display > Hide > Hide Deformers > Lattices.

### Toggling lattice shape handle (L icon)

- To help control screen clutter and display performance, you can select between displaying an "L" icon as the lattice deformer handle and displaying the deformer's lattices.

### To group the deformed lattice and the base lattice

1. Select the deformed lattice and base lattice.
2. Select Edit > Group.

   If you have grouped the base lattice and the deformed lattice, a simple way to select the two lattices in the scene (without opening the Outliner) is to select the deformed lattice and press the Up Arrow key to get the group node.

### To parent the lattice to the geometry

You can parent the lattice to the geometry in two ways, depending on when you're parenting:

- After you create the lattice, open the Outliner and drag and drop the lattice onto the geometry using the middle mouse button. An alternate way is to select the lattice, then the geometry, and choose Edit Parent.
- Before you create the lattice, open the Lattice Options window (select Deform > Lattice > ❐) and turn on the Parenting creation option.

#### To change lattice deformer performance settings

1. Select Window > Settings/Preferences > Performance Settings.
2. In the Performance Settings window, note the Deformers section.
3. Click the performance of Lattices to On, Off, or Interactive. (For more information, see the *Basics* guide.)
4. When you're done, click Close.

#### To change lattice resolution settings

1. Select Window > Settings/Preferences > Performance Settings.
2. In the Deformers section, set the Lattice Resolution to Per Node, Global, or Interactive.
3. Click Close.

#### To delete a lattice deformer

1. Select the lattice deformer node.
2. Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

   The deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved.

## Edit cluster deformers

#### To edit cluster channels with the Channel Box

1. Select a cluster deformer node (default name: cluster*n*).

   One quick way to select the cluster deformer node is to select the object being deformed, and then select the cluster deformer node in its history from the Channel Box (under INPUTS). Another way is to select the cluster deformer handle (the C icon).

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. In the Channel Box, the following channels are listed by default:
3. Click on a channel name with the left mouse button.

4 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

#### To edit cluster deformer attributes with the Attribute Editor

1 Select the cluster deformer node (default name: cluster).
2 Open the Attribute Editor by selecting Window > Attribute Editor or by using the default shortcut Ctrl+a.
3 Edit the attributes.

See ”cluster” on page 409.

#### To prune deformer set membership

1 Select deformable objects whose currently unaffected points you want to prune from the deformation.
2 Select Deform > Prune Membership > Cluster.

Maya removes the deformable object’s points currently unaffected points from the cluster deformer set.

**Note** Prune Membership only removes points from the deformation if they are currently not affected by the deformer. Therefore, if you try to prune the membership of an object whose points are all controlled by the deformer, you will receive the error message “No components could be pruned.”

#### To query cluster weights

1 Select the points whose cluster weights you want to edit.
2 Select Window > General Editors > Component Editor.

The *Component Editor* appears.

The Component Editor displays the component data for currently selected components in the workspace.

By default, the Component Editor updates dynamically as you select components in the workspace. Also, as you select components in the Component Editor, the workspace updates dynamically.

3 Click on the Weighted Deformers tab. The Weighted Deformers section lists the weights assigned to CVs, vertices, or lattice points by cluster deformers (default names: cluster*n*).

### To modify a point's cluster weight

**1** In the Component Editor's spreadsheet, click the component data box you want to edit.

Only the component whose box you've selected is now selected in the workspace.

**2** Enter a new value.

### To modify several cluster weights at once

**1** In the workspace, select the points whose weights you want to edit.

**2** In the Component Editor's spreadsheet, drag through the component data boxes you want to edit.

**3** Enter the value you want all the boxes to have.

### To modify an entire row or column (IRIX, Linux, and Mac OS X only)

**1** In the workspace, select the points whose weights you want to edit.

**2** In the Component Editor's spreadsheet, click one of the boxes in the row or column.

**3** Click the row or column heading.

Now all the boxes for the row or column are selected.

**4** Enter a value for all the boxes in the row or column.

### To modify an entire row or column (Windows only)

**1** In the workspace, select the points whose weights you want to edit.

**2** To change all the entries of a row or column, in the Component Editor's spreadsheet, select the row or column heading.

**3** Shift-click any of the boxes in that row or column.

**4** Enter a new value to update the entire row or column.

For more information about the Component Editor, see the *Basics* guide.

### To paint weights on a cluster

**1** Select the surface with the cluster you want to paint weights on.

**2** Select Shading > Smooth Shade All or press the default hotkey 5 to switch to smooth shading mode.

**3** Select Deform > Paint Cluster Weights Tool > ❐.

In the Tool Settings, the Paint Cluster Weights Tool Settings appear. See "Deform > Paint Cluster Weights Tool" on page 397.

| | |
|---|---|
| **Note** | The Paint Cluster Weights Tool automatically detects clusters on the surface and selects one for painting. |

4 In the Display section, Turn on Color Feedback. Color feedback helps you identify the weights on the surface by representing them as grayscale values (smaller values are darker, larger values are lighter).

| | |
|---|---|
| Tip | You can use the default hotkey Alt+c (IRIX, Linux, and Windows) or Option+c (Mac OS X) to turn Color Feedback on and off outside the Tools Settings Editor. |

5 Select the cluster you want to paint. In the Paint Attributes section of the Tool Settings window, click the cluster*n*.weights button and select the appropriate cluster weights name from the pop-up menu.

Note that you can only paint weights on one cluster at a time. If you select more than one cluster, you can only paint weights on the active one (the one that provides color feedback).

If the surface has only one cluster, you can select the surface alone.

The selected cluster turns white, representing a weight value of 1, the default.

| | |
|---|---|
| Tip | If you are painting on a single surface, you can skip step 3 and select the cluster without opening the Tool Settings window by right-clicking the surface and selecting the appropriate cluster weights name from the Paint command submenu. |

6 Select a brush, paint operation, and value and define other settings as required.

7 Drag the brush across the cluster.

> Tip You can quickly pick weight values from one cluster and paint them on another cluster or the same cluster using hotkeys.
>
> - Select the cluster with the weight values you want to pick.
> - Hold down the Pick Color Mode hotkey (default hotkey: /), click on the area of the cluster with the weight you want to pick, then release the hotkey.
> - If you are painting the picked weight on the same cluster, drag the brush across the cluster.
>
> If you are painting the picked weight on another cluster, select that cluster, then drag the brush across it.

**To delete a cluster deformer**

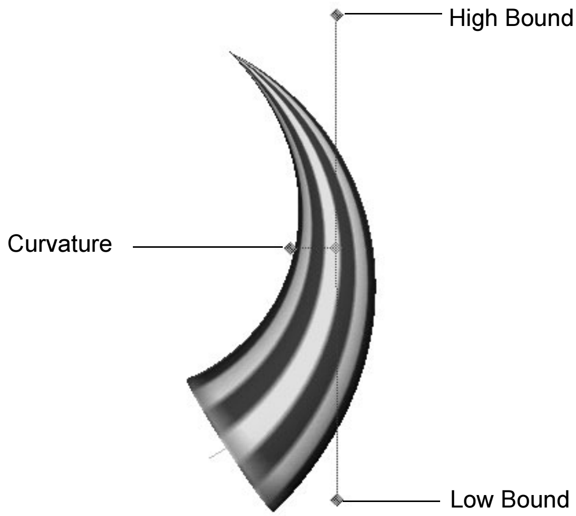
1. Select the cluster deformer node.
2. Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

   The deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved.

## Edit bend nonlinear deformers

**To manipulate bend deformer handles**

1. Select the bend deformer node (default name: bend*n*).
2. Select the Show Manipulator Tool (default shortcut: t key).
3. Note the manipulators on the bend deformer handle. These enable you to edit attributes interactively.




Bend deformer handle manipulators

4 In the scene, select one of the manipulators on the bend deformer handle.

5 Press the middle mouse button and move the mouse to edit.

The Channel Box updates the values you are changing.

### To edit bend deformers by moving, rotating, or scaling the handle

1 Select the bend deformer handle node (default name: bend*n*Handle).

2 Do one of the following:

- Move (translate), rotate, or scale the handle to change the effect of the deformation.
- Move or rotate the handle pivot point by pressing Insert (IRIX, Linux, and Windows) or Home (Mac OS X) key, move the pivot point, and then press the Insert or Home key again.

### To edit bend constraint channels with the Channel Box

1 Select the bend deformer node (default name: bend*n*).

One quick way to select the bend deformer node is to select the object being deformed, and then select the bend deformer node in its history from the Channel Box (under INPUTS).

Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).

2 Click on a channel name with the left mouse button.
3 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl gives you finer control, and pressing the Shift key gives you less control.

#### To edit bend deformer attributes

1 Select the bend deformer node (default name: bend*n*).
2 Open the Attribute Editor by selecting Window > Attribute Editor or use the default shortcut Ctrl+a.
3 Edit the attributes.

See ”Nonlinear deformer nodes” on page 410.

#### To delete a bend deformer

1 Select the bend deformer handle.
2 Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

The bend deformer handle, bend deformer handle shape, and bend deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved. Also, note that the various input nodes that structure the evaluation of the deformation are not deleted.

## Edit flare nonlinear deformers

#### To edit with the flare handle manipulators

1 Select the flare deformer node (default name: flare*n*).
2 Select the Show Manipulator Tool (default shortcut: t key).

The manipulators on the flare deformer handle let you to edit attributes interactively.

3 In the scene, select one of the manipulators on the flare deformer handle.
4 Press the middle mouse button and move the mouse to edit.

The Channel Box updates the values you are changing.

### To edit flare deformers by moving, rotating, or scaling the handle

1. Select the flare deformer handle node (default name: flare*n*Handle).
2. Move (translate), rotate, or scale the handle to change the effect of the deformation.
3. Move or rotate the handle pivot point by pressing the Insert (IRIX, Linux, and Windows) or Home (Mac OS X) key, moving the pivot point, and then pressing the Insert or Home key again.

### To edit flares with the Channel Box

1. Select the flare deformer node (default name: flare*n*).

   One quick way to select the flare deformer node is to select the object being deformed, and then select the flare deformer node in its history from the Channel Box (under INPUTS).

   You can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Click on a channel name with the left mouse button.
3. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you coarser control.

### To edit flare deformers with the Attribute Editor

1. Select the flare deformer node (default name: flare*n*).
2. Open the Attribute Editor by selecting Window > Attribute Editor or use the default shortcut Ctrl+a.
3. Edit the attributes.

   See ”flare” on page 411.

### To delete a flare deformer

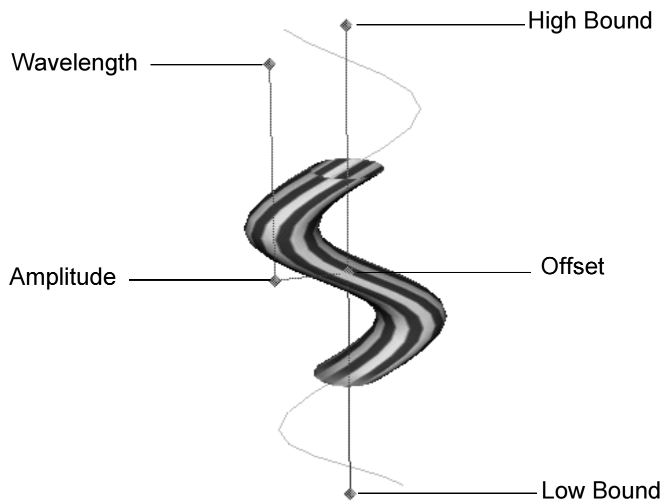
1. Select the flare deformer handle.
2. Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

   The flare deformer handle, flare deformer handle shape, and flare deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved. Also, note that the various input nodes that structure the evaluation of the deformation are not deleted.

# Edit sine nonlinear deformers

**To edit sine deformers by using the handle manipulators**

1. Select the sine deformer node (default name: sine*n*).
2. Select the Show Manipulator Tool (default shortcut: t key).
3. The manipulators on the sine deformer handle let you edit attributes interactively.




Sine deformer handle manipulators

4. In the scene, select one of the manipulators on the sine deformer handle. Press the middle mouse button and move the mouse to edit.

   The Channel Box updates the values you are changing.

**To edit sine deformers by moving, rotating, or scaling the handle**

1. Select the sine deformer handle node (default name: sine*n*Handle).
2. Move (translate), rotate, or scale the handle to change the effect of the deformation.
3. Move or rotate the handle pivot point by pressing the Insert (IRIX, Linux, and Windows) or Home (Mac OS X) key, moving the pivot point, and then pressing the Insert or Home key again.

4 Remember that you can access the deformer handle's local axes (Display > Component Display > Local Rotation Axes), rotate and scale pivots (Display > Component Display > Rotate Pivots or Scale Pivots) and selection handle (Display > Component Display > Selection Handles).

### To edit sine deformers with the Channel Box

1 Select the sine deformer node (default name: sine*n*).

One quick way to select the sine deformer node is to select the object being deformed, and then select the sine deformer node in its history from the Channel Box (under INPUTS).

You can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).

2 Click on a channel name with the left mouse button.

3 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

### To edit sine deformers with the Attribute Editor

1 Select the sine deformer node (default name: sine*n*).

2 Open the Attribute Editor by selecting Window > Attribute Editor or use the default shortcut Ctrl+a.

3 Edit the attributes.

See "sine" on page 412.

### To delete a sine deformer

1 Select the sine deformer handle.

2 Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

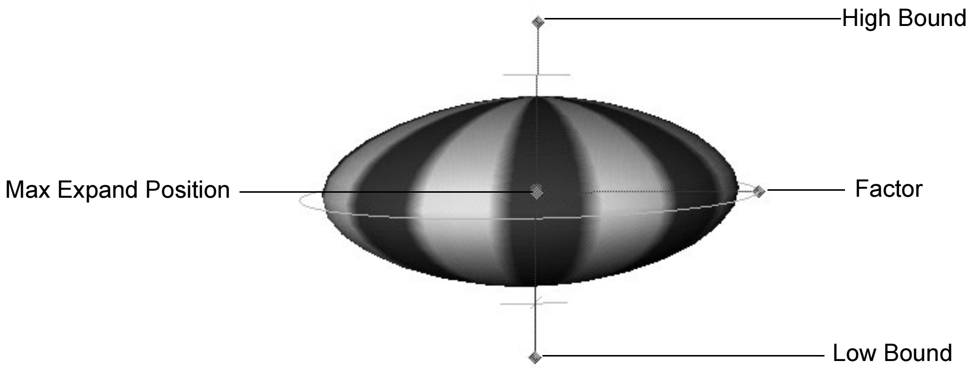
3 The sine deformer handle, sine deformer handle shape, and sine deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved. Also, note that the various input nodes that structure the evaluation of the deformation are not deleted.

# Edit squash nonlinear deformers

| | |
|---|---|
| **Tip** | Adjust Expand, Start Smoothness, and End Smoothness from the Channel Box. |

**To edit squash deformers using handle manipulators**

1. Select the squash deformer node (default name: squash*n*).
2. Select the Show Manipulator Tool (default shortcut: t key).




Squash deformer handle manipulators

3. In the scene, select one of the manipulators on the squash deformer handle.

   The manipulators on the squash deformer handle let you edit attributes.
4. Press the middle mouse button and move the mouse to edit interactively. Note that the Channel Box updates the values you are changing.

**To edit squash deformers by moving, rotating, or scaling handles**

1. Select the squash deformer handle node (default name: squash*n*Handle).
2. Move (translate), rotate, or scale the handle to change the effect of the deformation.
3. Move or rotate the handle pivot point by pressing the Insert (IRIX, Linux, and Windows) or Home (Mac OS X) key, moving the pivot point, and then pressing the Insert or Home key again.

Remember that you can access the deformer handle's local axes (Display > Component Display > Local Rotation Axes), rotate and scale pivots (Display > Component Display > Rotate Pivots or Scale Pivots) and selection handle (Display > Component Display > Selection Handle).

#### To edit squash deformers with the Channel Box

1. Select the squash deformer node (default name: squash*n*).

   One quick way to select the squash deformer node is to select the object being deformed, and then select the squash deformer node in its history from the Channel Box (under INPUTS).

   You can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Click on a channel name with the left mouse button.
3. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key will give you finer control, and pressing the Shift key will give you less control.

#### To delete a squash deformer

1. Select the Squash deformer handle.
2. Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

   The squash deformer handle, squash deformer handle shape, and squash deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved. Also, note that the various input nodes that structure the evaluation of the deformation are not deleted.

#### To edit squash deformers with the Attribute Editor

1. Select the squash deformer node (default name: squash*n*).
2. Open the Attribute Editor by selecting Window > Attribute Editor or use the default shortcut Ctrl+a.
3. Edit the attributes.

   See ”squash” on page 412.

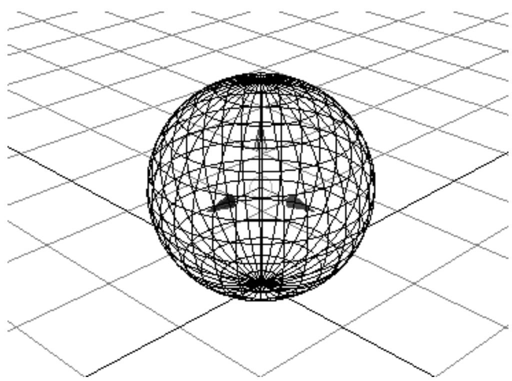
## Squash deformer example

### Squashing a sphere onto the ground

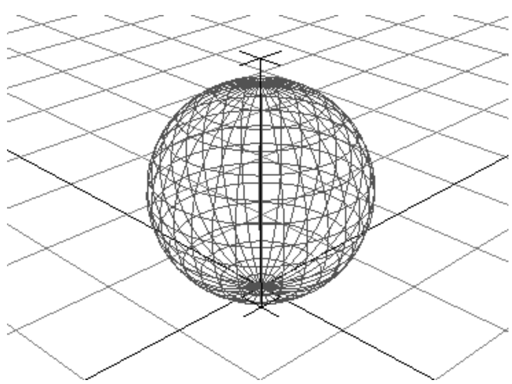
By default, Maya places nonlinear deformer handles at the center of the object to be deformed. For instance, when you create a sphere and create a squash deformer for it, Maya places the squash deformer handle at the center of the sphere. The deformation will be relative to the sphere's center. If you want to squash the sphere against the ground, you can adjust the squash deformer's attributes and move the squash deformer handle so that the deformation will be relative to where the sphere touches the ground. In general, you can make these adjustments so that the squashing effect can occur relative to any location inside or outside of the sphere.

**To set up the sphere and the deformer**

1. Create a primitive NURBS sphere.
2. Press the 3 number key to increase the display resolution.
3. Move the sphere so that it is sitting on the grid in a perspective view.



4. Create a squash deformer for the sphere.

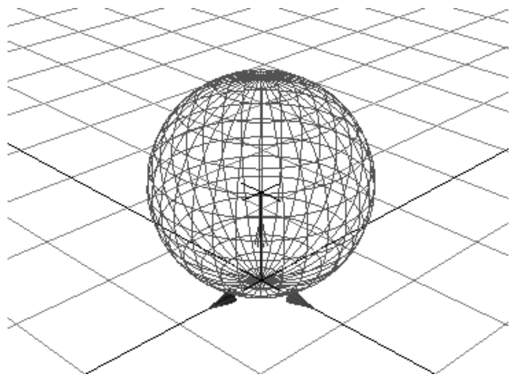


### To edit the deformer

1 Edit the squash deformer by setting squash1's attributes as follows:

| | |
|---|---|
| Low Bound | 0 |
| High Bound | 0.75 |

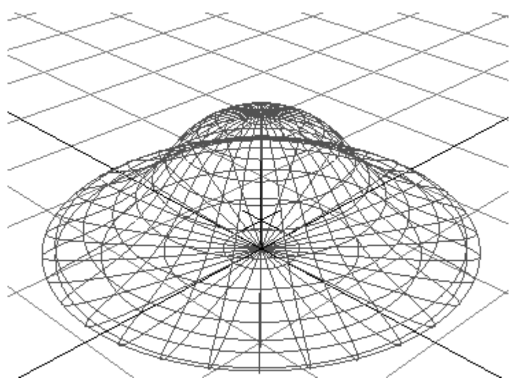
2 Move the deformer handle so the lower boundary is where the sphere is making contact with the ground.



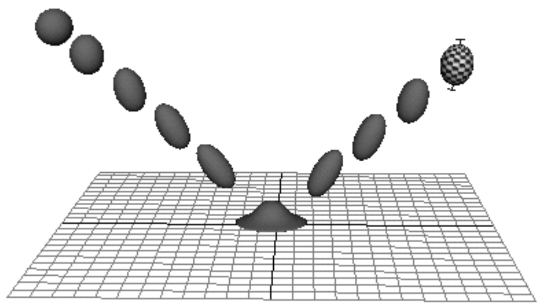
### To squash the sphere against the ground

1 Now, from the Channel Box, change the Factor.

The squash deformation takes place relative to the bottom of the sphere, where the sphere is touching the ground.



## Bouncing ball setup



You can create squash and stretch effects with the squash deformer. This example shows how you can set up a ball for a bouncing ball animation.

### To create NURBS sphere with squash control

1. Do the previous example.
2. Set squash1's Factor attribute back to 0.

### To create deformer for stretch control

1. Create another squash deformer for the sphere to provide stretch control. Use the default creation options.

   The squash1 deformer provides the squashing that occurs when the ball lands on the ground. The deformer you've just created (squash2) will provide the stretching that will occur when the ball is in flight.

### To define the ball

1. Group the sphere (nurbsSphere1) and the deformer handles (squash1Handle and squash2Handle).
2. Rename the group *ball*.
3. Open the Channel Control window (Window > General Editors > Channel Control), and make the following attributes Non Keyable:
   - scaleX, scaleY, scaleZ
   - visibility

   The Channel Box now lists only the following keyable attributes for ball: Translate X, Translate Y, Translate Z, Rotate X, Rotate Y, and Rotate Z.
4. Close the Channel Control window.

   Now you will add two attributes to the ball for squashing and stretching.

5 With the ball selected, select Modify > Add Attribute.
   - Add a keyable attribute called *flyStretch*, with the following Min/Max Values: Minimum 0, Maximum 10, Default 0.
   - Add a keyable attribute called *landSquash*, with the following Min/Max Values: Minimum 0, Maximum 10, Default 0.

6 Close the Add Attribute window.

**To set driver and driven keys for stretching**

1. Open the Set Driven Key window (Animate > Set Driven Key > Set > ❐).
2. Load ball as driver, select flyStretch attribute, and set the attribute to 0.
3. Load squash2 as driven, select factor attribute, and set the attribute to 0.
4. Click Key.
5. Set ball's flyStretch attribute to 10.
6. Set squash2's factor attribute at 0.6.
7. Click Key.

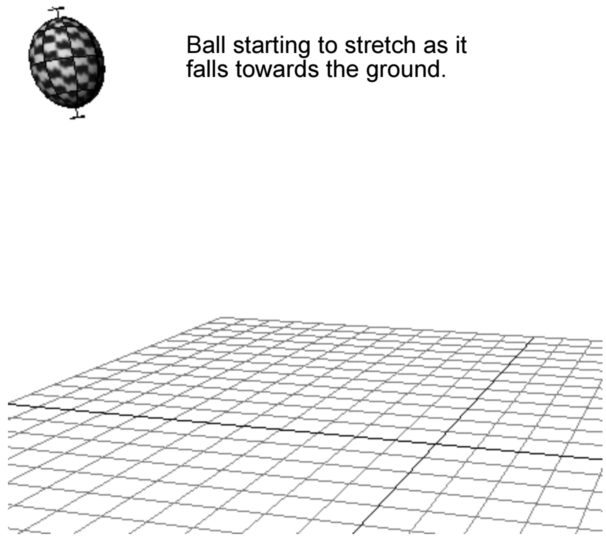
**To set driver and driven keys for squashing**

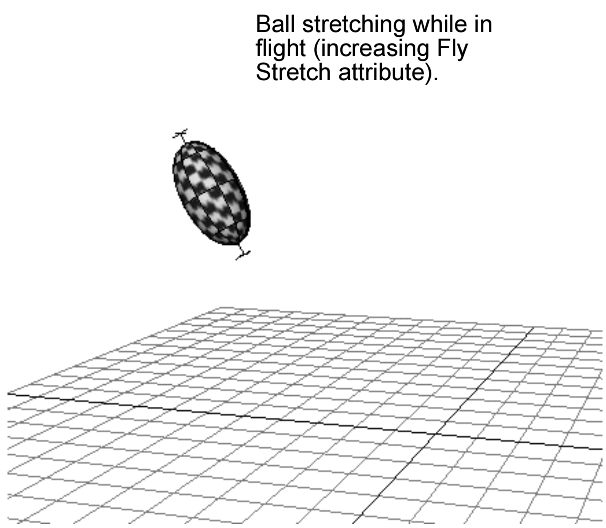
1. Open the Set Driven Key window if it is not already opened.
2. Load ball as driver, select landSquash attribute, and set the attribute to 0.
3. Load squash1 as driven, select factor attribute, and set the attribute to 0.
4. Click Key.
5. Set ball's landSquash attribute to 10.
6. Set squash1's factor attribute at -2.
7. Click Key.
8. Click Close to close the editor.

   Now the ball is ready for a bouncing ball animation with squash and stretch effects.

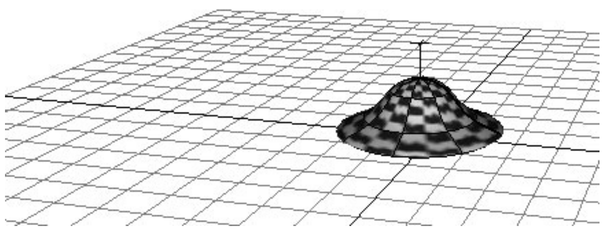
### Animating the ball

You've now set up the ball for animation. Try creating an animation of the ball bouncing. Include stretch effects when the ball is in flight and squash effects when the ball hits the ground. For example, your animation might look something like the following images.

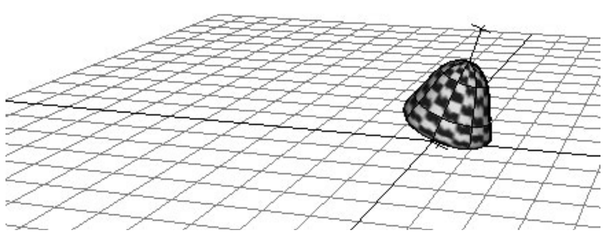

Ball starting to stretch as it falls towards the ground.

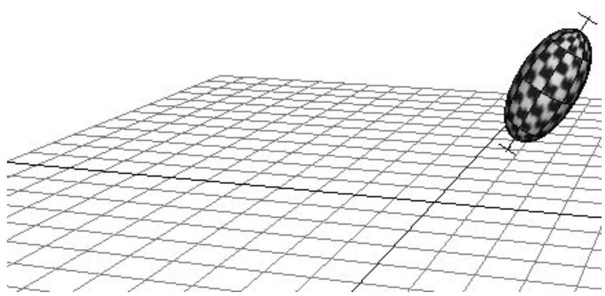

Ball stretching while in flight (increasing Fly Stretch attribute).

Ball lands on the ground. Fly Stretch attribute is 0, and Land Squash attribute is 10.



Ball leaving the ground with reduced squashing and increased stretching.






These images show just a simple example of a bouncing ball. As you develop your animation of a bouncing ball, try to see how much “character” you can give to the ball’s movements.

## Control a Sculpt deformer’s strength using a texture

**To use a two dimensional texture with the Sculpt deformer**

**1** Select the object that you want to deform.

**2** Create a sculpt deformer for that object.

See *“Set sculpt deformers”*.

**3** Select the sculpt deformer and open the Attribute Editor.

**4** Select the sculpt node’s tab, and expand its Sculpt History section.

**5** Click the Add Advanced Sculpt Attributes button.

The Advanced Sculpt Attributes appear in the Sculpt History section.

**6** Turn on Enable Advanced.

**7** Click the map button that is next to the Texture attribute. See ”Texture” on page 417.

The Create Render Node window appears. See ”Create > Create Render Node” on page 92.

**8** Click the 2D texture that you want to use with your sculpt deformer.

The strength of your sculpt deformer is now affected by the texture.

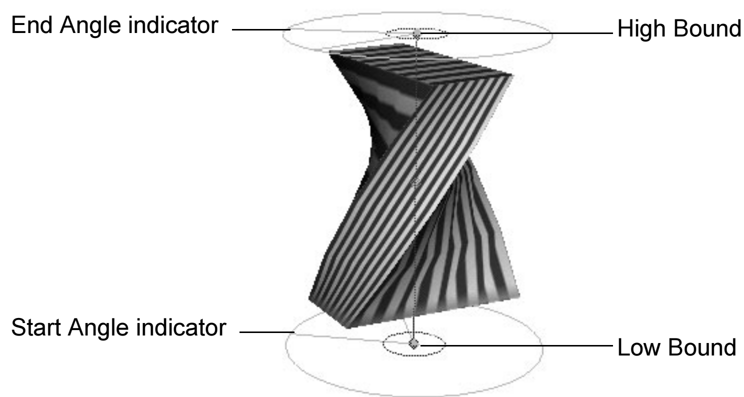
**9** Set the advanced sculpt attributes.

See ”Add Advanced Sculpt Attribute” on page 416.

# Edit twist nonlinear deformers

### To edit twist deformable objects by using handle manipulators

1. Select the twist deformer node (default name: twist*n*).
2. Select the Show Manipulator Tool (default shortcut: t key).




Twist deformer handle manipulators

3. In the scene, select one of the manipulators on the twist deformer handle.

   The manipulators on the twist deformer handle let you edit attributes.
4. Press the middle mouse button and move the mouse to edit interactively. Note that the Channel Box updates the values you are changing.

### To edit twist deformers by moving, rotating, or scaling handle

1. Select the twist deformer handle node (default name: twist*n*Handle).
2. Move (translate), rotate, or scale the handle to change the effect of the deformation.
3. Move or rotate the handle pivot point by pressing the Insert (IRIX, Linux, and Windows) or Home (Mac OS X) key, moving the pivot point, and then pressing the Insert or Home key again.

   Remember that you can access the deformer handle's local axes (Display > Component Display > Local Rotation Axes), rotate and scale pivots (Display > Component Display > Rotate Pivots or Scale Pivots) and selection handle (Display > Component Display > Selection Handle).

#### To edit twist deformers with the Channel Box

1. Select the twist deformer node (default name: twist*n*).

   One quick way to select the twist deformer node is to select the object being deformed, and then select the twist deformer node in its history from the Channel Box (under INPUTS).

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. In the Channel Box, the following channels are listed by default:
3. Click on a channel name with the left mouse button.
4. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

#### To edit twist deformers with the Attribute Editor

1. Select the twist deformer node (default name: twist*n*).
2. Open the Attribute Editor by selecting Windows > Attribute Editor or use the default shortcut Ctrl+a.
3. Edit the attributes.

   See ”twist” on page 413.

#### To delete a twist deformer

1. Select the twist deformer handle.
2. Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

   The twist deformer handle, twist deformer handle shape, and twist deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved. Also, note that the various input nodes that structure the evaluation of the deformation are not deleted.

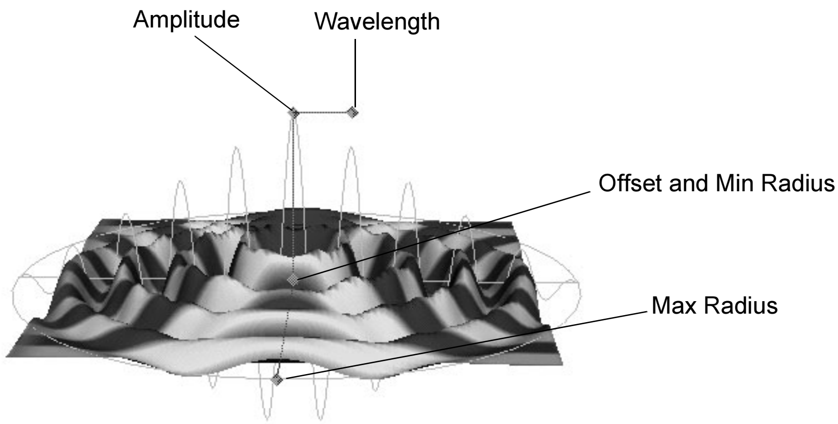
## Edit wave nonlinear deformers

**Tips**

- Offset and Min Radius manipulators are both at the center of the handle by default.
- Adjust Dropoff from the Channel Box.

### To edit wave deformers by using handle manipulators

1. Select the wave deformer node (default name: wave*n*).
2. Select the Show Manipulator Tool (default shortcut: t key).




Wave deformer handle manipulators

3. In the scene, select one of the manipulators on the wave deformer handle.

   The manipulators on the wave deformer handle let you edit attributes.
4. Press the middle mouse button and move the mouse to edit interactively. Note that the Channel Box updates the values you are changing.

   The Offset and Min Radius manipulators are both located at the center of the handle by default.

### To edit by moving, rotating or scaling handle

1. Select the wave deformer handle node (default name: wave*n*Handle).
2. Move (translate), rotate, or scale the handle to change the effect of the deformation.
3. Move or rotate the handle pivot point by pressing the Insert (IRIX, Linux, and Windows) or Home (Mac OS X) key, moving the pivot point, and then pressing the Insert or Home key again.

   Remember that you can access the deformer handle's local axes (Display > Component Display > Local Rotation Axes), rotate and scale pivots (Display > Component Display > Rotate Pivots or Scale Pivots) and selection handle (Display > Component Display > Selection Handle).

#### To edit waved deformers with the Channel Box

1. Select the wave deformer node (default name: wave*n*).

   One quick way to select the wave deformer node is to select the object being deformed, and then select the wave deformer node in its history from the Channel Box (under INPUTS).

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Click on a channel name with the left mouse button.
3. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

#### To edit wave deformers with the Attribute Editor

1. Select the wave deformer node (default name: wave*n*).
2. Open the Attribute Editor by selecting Windows > Attribute Editor or use the default shortcut Ctrl+a.
3. Edit the attributes.

   See ”wave” on page 414.

#### To delete a wave deformer

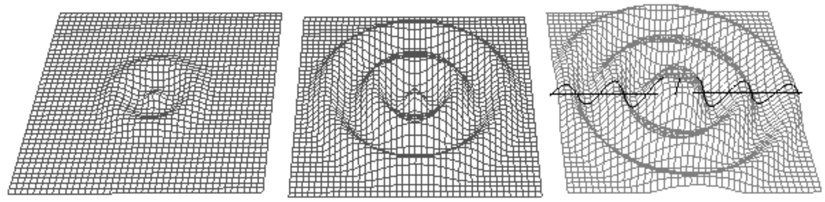
1. Select the wave deformer handle.
2. Select Edit > Delete or us the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

   The wave deformer handle, wave deformer handle shape, and wave deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved. Also, note that the various input nodes that structure the evaluation of the deformation are not deleted.

## Wave deformer example

### Ripple animation

You can create ripple effects by using a wave deformer on a NURBS or polygonal surface. This example shows how you can create a simple ripple effect on a NURBS plane.



### To create the NURBS plane

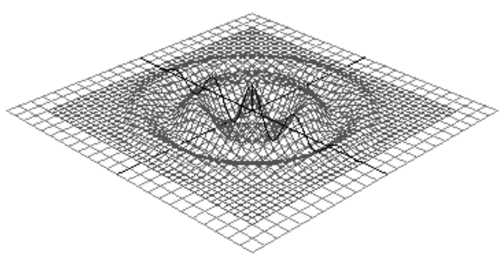
Create a NURBS plane, using the default creation options except set Width to 20, U Patches to 40, and V Patches to 40.

### To create a wave deformer

With the plane selected, create a wave deformer for the plane with the following creation options:

| | |
|---|---|
| Min Radius | 0 |
| Max Radius | 1 |
| Amplitude | 0.2 |
| Wavelength | 0.4 |
| Dropoff | 1 |
| Offset | 0 |

The result follows:

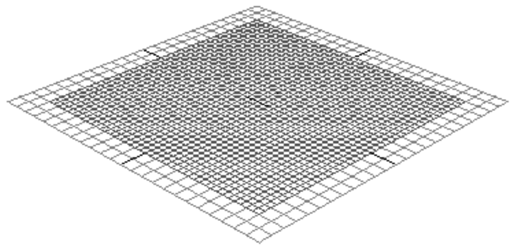


Next, you will set keys at frames 1, 10, and 20.

### To key ripple at frame 1

1 In the Timeline, select frame 1.
2 In the Channel Box, set wave1's attributes as follows:

| | |
|---|---|
| Amplitude | 0.0 |
| Max Radius | 0.1 |



Ripple at frame 1

**3** Set keys for all of wave1's attributes.

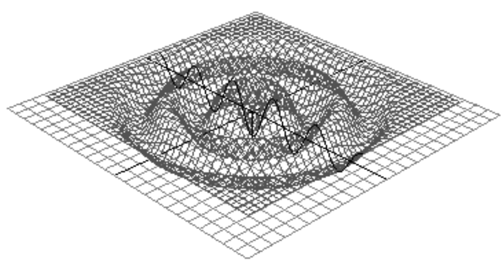
#### To key ripple at frame 10

**1** In the Timeline, select frame 10.

**2** In the Channel Box, set wave1's attributes as follows:

| | |
|---|---|
| Amplitude | -0.2 (negative value allows first wavelet to go down) |
| Dropoff Position | 0.5 |
| Max Radius | 1 |

The result is as follows:



Ripple at frame 10

**3** Set keys for all of wave1's attributes.

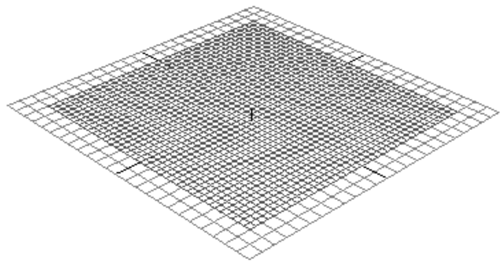
#### To key ripple at frame 20

**1** In the Timeline, select frame 20.

**2** In the Channel Box, set wave1's attributes as follows:

| | |
|---|---|
| Min Radius | 1 |
| Amplitude | -0.1 |

The result is as follows:



Ripple at frame 20

3 Set keys for all of wave1's attributes.

**To see the ripple**

1 Scrub or play the animation.
2 You can create more intricate and complicated rippling effects by continuing to adjust the wave deformer's attributes. You can also apply additional wave or sine deformers to the plane for more complex results.

| | |
|---|---|
| Tip | Instead of animating the Dropoff Position attribute, you could write an expression that drives the drop off of the wave. The expression would be as follows:<br>`wave1.offset = wave1.wavelength * .25;` |

## Edit sculpt deformers

**To manipulate the sculpt deformer sphere**

1 In the workspace or the Outliner, select the sculpt sphere (default name: sculpt*n*Sphere).
2 To create deformation effects, move, rotate, or scale the sculpt sphere.

**To manipulate the sculpt deformer stretch origin locator**

1 In the workspace or the Outliner, select the stretch origin locator (default name: sculpt*n*StretchOrigin).
2 To create deformation effects, move, rotate, or scale the stretch origin locator.

**To edit sculpt deformer channels with the Channel Box**

1 Select a sculpt deformer node (default name: sculpt*n*).

One quick way to select the sculpt deformer node is to select the object being deformed, and then select the sculpt deformer node in its history from the Channel Box (under INPUTS).

Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control.)

2. In the Channel Box, click on a channel name with the left mouse button.
3. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

**To edit sculpt deformer attributes with the Attribute Editor**

1. Select the sculpt deformer node (default name: sculpt*n*).
2. Open the Attribute Editor by selecting Windows > Attribute Editor or use the default shortcut Ctrl+a.
3. Edit the attributes.

   See "Sculpt deformer nodes" on page 415.

**To prune sculpt deformer set membership**

1. Select deformable objects whose currently unaffected points you want to prune from the deformation.
2. Select Deform > Prune Membership > Sculpt.

   Maya removes the deformable object's points currently unaffected points from the sculpt deformer set.

> **Note** Prune Membership only removes points from the deformation if they are currently not affected by the deformer. Therefore, if you try to prune the membership of an object whose points are all controlled by the deformer, you will receive the error message "No components could be pruned."

**To delete a sculpt deformer**

1. Select the sculpt deformer node.
2. Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

The deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved.

## Edit jiggle deformers

### To disable or delete the jiggle cache for all jiggling objects

1. Select Deform > Jiggle Disk Cache Attributes to display the Attribute Editor.
2. In the Control For All Caches section:
   - Click Delete All Caches.
   - From the Enable Status menu, select Disable All.

### To delete the jiggle cache for a particular object

1. Select the object to which you applied the jiggle deformer.
2. Select Window > Attribute Editor. Select the jiggle*n*Cache tab. (Expand the size of the Attribute Editor to see all tabs.)
3. Click Delete Cache.

## Edit wire deformers

You can edit all of a wire deformer's attributes with the Attribute Editor.

**Warning** Don't use freezeGeometry on wire deformers if you plan to move the base wire.

Otherwise, if you have a wire deformer whose freezeGeometry attribute is turned on and you move its base wire relative to the deformed surface (ore vice versa), the deformation's behavior will change the next time the scene is loaded into Maya.

### To edit wire channels with the Channel Box

1. Select a wire deformer node (default name: wire*n*).

   One quick way to select the wire deformer node is to select the object being deformed, and then select the wire deformer node in its history from the Channel Box (under INPUTS).

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).

2 In the Channel Box, click on a channel name with the left mouse button.

3 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key will give you finer control, and pressing the Shift key will give you coarser control.

#### To edit wire attributes with the Attribute Editor

1 Select the wire deformer node (default name: wire*n*).

2 Open the Attribute Editor by selecting Window > Attribute Editor or use the default shortcut Ctrl+a.

3 Set the attributes.

See ”Wire deformer nodes” on page 421.

#### To add a wire dropoff locator

1 To select a curve point on the influence wire, right-click the influence wire curve and select Curve Point from the marking menu.

2 Click the influence wire curve roughly where you would like to put the wire dropoff locator. The curve point is displayed as a small yellow box.

3 Drag along the curve to adjust the point’s position on the curve.

As you drag, you move the curve point. The curve point’s position is defined in terms of the curve’s U parameter.

Now you need to specify the curve point as a wire dropoff locator.

4 Select Deform > Wire Dropoff Locator.

The curve point is now a wire dropoff locator.

5 To add more wire dropoff locators, repeat steps 2 through 5.

#### To move a wire dropoff locator

1 Be sure you are in components selection mode, with the parameter points selection mask on.

2 Select the wire dropoff locator shape node.

3 Select the Move Tool.

4 You can now move the dropoff locator along the influence wire curve.

#### To edit a wire dropoff locator’s channels

1 Select the wire dropoff locator shape node (default name: locatorShape*n*).

**2** In the Channel Box, edit its channels.

Note the wire deformer also includes Locator Envelope and Wire Locator Twist channels for each wire dropoff locator. These channels correspond to the wire deformer's Envelope and Twist attributes. See "Wire deformer nodes" on page 421.

#### To edit a wire dropoff locator's attributes

**1** Select a wire dropoff locator (default name: locatorShape*n*).

**2** Open the Attribute Editor by selecting Window > Attribute Editor or use the default shortcut Ctrl+a.

**3** Edit the attributes.

See "Wire Dropoff Locator nodes" on page 423.

#### To smooth jagged effects

**1** Do one of the following:

- Increase the wire deformer's Dropoff Distance attribute. See "Wire deformer nodes" on page 421.
- Add more points to the object's surface. For example, if the object is a NURBS surface, increase the number of control vertices.

### Add and remove holders

Holders are curves that limit the deformation region. Adding or removing a holder can sometimes lead to unexpected changes in the deformation region. You can remedy these effects by editing and pruning the wire deformer set.

#### To add a holders

You can a holder to a deformer only if it doesn't already have a holder.

**1** Select the curves you want to add to the deformer as holders.

**2** Shift-click on any wire in the deformer to select the deformer to which you want add the curve as a holder.

**3** Select Deform > Edit Wire > Add Holder.

The selected curve becomes a holder for the wire deformer.

#### To remove holders

To remove a holder you must delete the curve. If you need the curve, first duplicate the curve then delete the original one (that serves as a holder).

**1** Select the curve that you want to remove as holder.

**2** Delete the curve.

## Move, rotate, scale holders

Moving, rotating, or scaling holders can change the deformation effect and the deformation region. You can move, rotate, or scale a holder in the same way that you would move, rotate, or scale any object in Maya.

## Edit the shape of holders

Editing the shape of holders can change the deformation effect and the deformation region. You edit the shape of holders in the same way that you edit NURBS curves during modeling.

## Edit wire deformer sets

For more information, see ”Editing deformer set membership” on page 234.

## Prune wire deformer sets

Pruning is useful for quickly limiting the deformation region as you manipulate influence wires.

### To prune deformer set membership

1. Select a wire deformer node.
2. Move the influence wire(s) so that only those points you want to keep in the deformer set are being affected.
3. Select Deform > Prune Membership > Wire.

   The undeformed points are removed from the deformer set.

**Note** Prune Membership only removes points from the deformation if they are currently not affected by the deformer. Therefore, if you try to prune the membership of an object whose points are all controlled by the deformer, you will receive the error message “No components could be pruned.”

### To delete a wire deformer

1. Select the wire deformer node.
2. Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux and Windows) or Delete (Mac OS X) key.

   The deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved.

## Edit wrinkle deformers

#### To move, rotate, or scale Wrinkle influence wires

1. In the Outliner, select the influence wires you want to move, rotate, or scale. The influence wires are listed under the wrinkle deformer's cluster deformer handle.
2. Select Display > Show > Show Selection.

   Now you can manipulate each influence wire in the same way that you would if working with a wire deformer. For example, you can even move the base wires or add wire dropoff locators.

#### To delete a wrinkle deformer

1. Select the wrinkle deformer's cluster deformer node.
2. Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux and Windows) or Delete (Mac OS X) key.

   The deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved.

## Edit wrap deformers

#### To move, rotate, or scale a wrap influence object

1. Select the wrap influence object.
2. Move, rotate, or scale the wrap influence object.

#### To edit by manipulating wrap influence object points

1. Select points (CVs or vertices) of the wrap influence object.
2. Move, rotate, or scale the vertices.

#### To move, rotate, or scale the wrap deformed object

1. Select the deformed object.
2. Move, rotate, or scale the deformed object.

#### To edit wrap deformer channels with the Channel Box

1. Select the NURBS curve or surface acting as a wrap influence object.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).

2 In the Channel Box, click on a channel name with the left mouse button.
3 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

#### To edit wrap deformer channels with the Channel Box

1 Select the polygonal surface acting as a wrap influence object.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control.)
2 In the Channel Box, click on a channel name with the left mouse button.
3 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

#### To edit wrap deformer channels with the Channel Box

1 Select a wrap deformer node (default name: wire*n*).

   One quick way to select the wrap deformer node is to select the object being deformed, and then select the wrap deformer node in its history from the Channel Box (under INPUTS).

   Each object being deformed has its own input wrap deformer node.

   You can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control.)
2 In the Channel Box, click on a channel name with the left mouse button.
3 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

#### To edit wrap deformer attributes with the Attribute Editor

1 Select a wrap deformer node (default name: wrap*n*).

2 Open the Attribute Editor by selecting Windows > Attribute Editor or use the default shortcut Ctrl+a.
3 Edit the attributes.

See "Wrap deformer nodes" on page 423.

#### To add wrap deformer influence objects

If needed, create the objects you want to use as wrap influence objects. See "To add wrap deformer influence objects" on page 355.

1 Select the deformed object(s), or their wrap deformer nodes, to which you want to add the wrap influence object.
2 Now also select the object or group of objects that you want to add as wrap influence objects.
3 Select Deform > Edit Wrap > Add Influence.

See "Deform > Edit Wrap > Add Influence" on page 396.

#### To remove wrap deformer influence objects

1 Select the deformed object(s), or their wrap deformer nodes, from which you want to remove the wrap influence object.
2 Now also select the wrap influence objects whose influence you want to remove.
3 Select Deform > Edit Wrap > Remove Influence.

See "Deform > Edit Wrap > Remove Influence" on page 396.

#### To delete a wrap deformer

1 Select the wrap deformer node.
2 Select Edit > Delete or use the default shortcut Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key.

   The deformer nodes are all deleted. However, the object still has the tweak node as an input node, so any tweaks you might have made are preserved.

## Edit deformer set membership

You can directly edit deformer set membership with the Edit Membership Tool.

# Edit deformer set membership

### To edit deformer set membership with the Edit Membership Tool

1. Select Deform > Edit Membership Tool.
2. Select the deformer you want to edit.
3. Go into component selection mode (click the select by component type icon).

   The members of the deformer set whose joint you selected are displayed in yellow. This set is the currently selected set. Members of other sets are displayed in other colors. Points displayed in dark red are not in a set.
4. To add points to the currently selected set, select them while pressing the Shift key and left mouse button, and then release the mouse button.

   The selected points are now displayed in yellow, indicating they are in the currently selected set.
5. To remove points from the currently selected set, select them while pressing the Ctrl+the left mouse button.

   The selected points are now displayed in dark red, indicating they are currently not in a set.
6. Click the Select Tool to quit the editing mode.

## Change deformation order

# Change deformation order

### To change deformation order

1. In the scene, move the pointer to the object being deformed and press the right mouse button.

   A marking menu is displayed.
2. From the marking menu, select Inputs > Complete List.

   The List of history operations window is displayed for the selected object.

   Move the pointer over the name of the deformer whose order you want to change. Press the middle mouse button, drag over the name of the operation you want the deformer to precede, and release the mouse button.

#### To change point tweaking's deformation order

1. In the scene, move the pointer to the object being deformed and press the right mouse button.

   A marking menu appears.
2. From the marking menu, select Inputs > Complete List.

   The List of history operations window is displayed for the selected object.
3. Move the pointer over the name of the tweak node (default name: tweak*n*) whose order you want to change.
4. Press the middle mouse button, drag over the name of the operation that is where you want point tweaking to take place, and release the mouse button.

### Set the display of deformation objects

## Set the display of deformation objects

#### To show all deformers

Select Display > Show > Show Deformers > All.

You can also show all lattices, sculpt influence objects, cluster handles, nonlinear deformer handles, or wrap influence objects.

#### To hide all deformers

Select Display > Hide > Hide Deformers > All.

You can also hide all lattices, sculpt influence objects, cluster handles, nonlinear deformer handles, or wrap influence objects.

#### To display intermediate deformation object(s)

1. Select the object(s) being deformed.
2. Select Deform > Display Intermediate Objects.

#### To hide intermediate deformation object(s)

1. Select the intermediate object(s) being displayed.
2. Select Deform > Hide Intermediate Objects.
3. Change the deformer.

## Set deformer node performance

# Set deformer node performance

### To change dependency graph node performance

1. Select Window > Settings/Preferences > Performance Settings.
2. In the Performance Settings window, note the Dependency Graph Evaluation section.
3. Turn on one of the Refresh On options.

For more information, search the online help for Performance Settings.

4. When you're done, click Close.

### To change deformer performance settings

1. Select Window > Settings/Preferences > Performance Settings.
2. In the Performance Settings window, note the Deformers section.

   You can turn the performance of the following on, off, or to Interactive: sculpt influence objects (Sculpts), lattice influence objects (Lattices), wire influence objects (Wires), blend shapes, and clusters. You can set Cluster Resolution to Per Node, Global, or Interactive. You can set the Lattice Resolution to Per Node, Global, or Interactive.
3. When you're done, click Close.

### To edit channels with the Channel Box

In some object's deformation chain, when the placement of one (or more) of the deformers is set to Parallel, you can blend the influences of the deformers in the chain in parallel.

1. Select a parallel blender node (default name: parallelBlender*n*).

   One quick way to select the blend shape deformer node is to select the object being deformed, and then select the parallel blender node in its history from the Channel Box (under INPUTS).

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. In the Channel Box, click on a channel name with the left mouse button.
3. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives finer control, and pressing the Shift key gives you less control.

#### To add shear channels

1 Select the deformable object.
2 Select Window > General Editors > Channel Control.
The Channel Control window is displayed. In the Non Keyable pane, note the following attributes: shearXY, shearXZ, and shearYZ.
3 For each attribute, select it, and click the << Move button.
The attributes are added to the Keyable pane. In the Channel Box, the channels for the object now include Shear XY, Shear XZ, and Shear YZ. You can now create shearing effects by using the Channel Box.
4 To close the Channel Control window, click the Close button.

#### To delete an object's history

1 Select the object.
2 Select Edit > Delete By Type > History.

#### To set node behavior

1 Open the node's Attribute Editor.
2 In the Attribute Editor, open Node Behavior.
See "General deformer node attributes" on page 405.
3 Click Caching on or off.
4 Select the Node State as Normal, HasNoEffect, or Blocking.
5 Close the Attribute Editor.

### Edit deformer sets with the Relationship editor

## Edit deformer sets with the Relationship editor

To edit deformer sets with the Relationship Editor, select Window > Relationship Editors > Deformer Sets. For more information about sets and using the *Relationship Editor*, see the *Basics* guide.

### Work with the tweak node

## Work with the tweak node

#### To disable a tweak node

1 Open the tweak node's Attribute Editor.
2 In the Attribute Editor, open Node Behavior.

See ”General deformer node attributes” on page 405.

3 Set Node State to HasNoEffect.

**To enable a tweak node**

1 Open the tweak node’s Attribute Editor.
2 In the Attribute Editor, open Node Behavior.
See ”General deformer node attributes” on page 405.
3 Set Node State to Normal.

# Reference Windows and Editors

## Blend Shape Editor

### Blend Shape Editor

#### Target weight sliders

Each slider provides a way for you to set the target weight quickly. By default, each target slider sets values from the minimum value (by default, 0.000) to the maximum value (by default, 1.000). The current weight is displayed in the target weight box below the target slider.

You can change the orientation of the sliders. For more information, see ”To edit Blend Shape channels with the Channel Box” on page 316.

#### Target weight boxes

Each box displays the current weight for each target. By default, a target weight can range from the minimum value (by default, 0.000) to the maximum value (by default, 1.000). As you change values, the target weight sliders update according to the value you enter in the target weight boxes.

You can enter values from -10.000 to 10.000 to invert or amplify the target’s influence. If you enter a value greater than the current maximum value (by default, 1.000), the maximum value changes to double the value or to 10.000 if double the value is greater than 10.000. If you enter a value less than the current minimum value (by default, 0.000), the minimum value changes to double the value or to -10.000 if double the value is less than -10.000. When you enter values less than the current minimum or greater than the current maximum, the target weight sliders change to reflect the new range of values.

#### Target names

By default, the target name is the name of a target object (for example, nurbsSphere1). If you prefer, enter some other name for the target name. Entering a new target name does not change the name of the target object. Changing the target name is useful if you want to give a more appropriate target name after you've created the blend shape deformer.

#### New button

Click to create a new blend shape deformer. Clicking New is the same as selecting Deform > Create Blend Shape.

#### Add button

Bake the selected base shape and add it as a target.

#### Key All button

Key all weights at their current values.

#### Reset All button

Set all weight values to zero.

#### Select button

Select the blend shape deformer node.

#### Key buttons

Key the current value, or drag and drop on the Timeline to set an exclusive key.

### Blend Shape Attributes

#### Origin

Specifies whether the blend shape is relative to the base object shape's position, rotation, and scale, or is directly specified by you. Select local, world, or user.

| | |
|---|---|
| Local | Blends the base object shape to the target object shape(s) while ignoring differences in position, rotation, and scale between the target shape(s). |
| World | Blends the base object shape to the target object shape(s), taking into account any differences in position, rotation, and scale between the target object shape(s). |

The local and world selections are identical to the Origin creation option’s selections. The user selection, however, is not one of the Origin creation option’s selections.

User — Specifies that two special attributes, baseOrigin and targetOrigin, provide origin information. For more information on these attributes, see the online documentation for the blend shape deformer node (default name: blendShapen) and the blendShape MEL command. You can use the setAttr MEL command to set the values of the baseOrigin and targetOrigin attributes.

### Weight

Target — Specifies the influence value, or weight, of the named target. A value 0.000 specifies that the target has no influence; a value of 1.000 specifies that the target has maximum influence. For each named target, use slider to select values from 0.000 to 1.000.

### Deformer Attributes

Envelope — Specifies the deformation scale factor. Use slider to select values from 0.000 to 1.000. You can also enter values from -2.000 to 2.000. A value of 2.000 would double the overall deformation effect. A negative value would invert the effect. Default is 1.000.

### Node Behavior

See ”General deformer node attributes” on page 405.

### Extra Attributes

No extra attributes by default.

## Channel Control Editor

### Channel Control Editor

Channels are the keyable attributes displayed in the Channel Box. The Channel Box provides a convenient way to edit the channels available for a deformer.

Envelope

Specifies the deformation scale factor. Select values from 0 to 1. You can also enter values from -2 to 2. A value of 2 would double the overall deformation effect. A negative value would invert the effect. Default is 1.

Weight[*n*]

Specifies the influence of one of the deformers in the deformation chain. A value 0 specifies that the target has no influence and a value of 1 specifies that the target has maximum influence.

# Menus

## Animation menu set

### Deform >

## Deform > Edit Membership Tool

You can directly edit deformer set memberships by selecting deformable object points with the Edit Membership Tool.

## Deform > Prune Membership

You can remove unaffected points from a deformer set based on which points the deformer is affecting. Use this to avoid unnecessary calculations for points that are not being affected by the deformation. Prune Membership is available for only Cluster, Lattice, Sculpt, and Wire deformers.

The pruning operation considers only the current position of each component in the undeformed and deformed versions of the geometry affected by the specified deformation. If you have animated attributes of your deformation, the pruning operation is performed based only on the current attribute values. This means that components that are potentially affected at other frames of your animation might get pruned out if they are unaffected at the current frame.

Since a typical blend shape operation has weights of 0.0 for some target shapes at any point in time, this operation is especially dangerous when applied to blend shape deformations. For this reason, there is no menu item provided to prune membership for blend shape deformers. However, you can access prune membership for blend shape deformers through the command line.

# Deform > Create Blend Shape

## Deform > Create Blend Shape > ❐

### Basic tab

#### BlendShape Node

Specifies the name of the blend shape deformer (the blend shape deformer algorithm node). You should rename this node so that its name reminds you of the role of the blend shape deformation (for example, lipSync). If you don't specify a name, Maya provides the default name blendShape*n.*

#### Envelope

Specifies the deformation scale factor. Use the slider to specify values from 0.0000 to 1.0000. Default is 1.0000.

#### Origin

Specifies whether the blend shape will be relative to the base object shape's position, rotation, and scale. Click Local or World. The default is Local.

| | |
|---|---|
| Local | Local will blend the base object shape to the target object shape(s) while ignoring differences in position, rotation, and scale between the base object and the target object(s). For facial animation setup, you would typically want to select Local. In general, Local is useful when you want to have your target object(s) in various separate positions for easy viewing but don't want their positions to affect the deformation. |
| World | World will blend the base object shape to the target object shape(s), taking into account any differences in position, rotation, and scale between the target object shape(s). |

#### Target Shape Options

| | |
|---|---|
| In-Between | Specifies whether the blending will be in series or in parallel.<br><br>If on, the blending will be in series. Shape transitions will occur in the order in which you selected the target shape(s). The effect will be that the blend shape will be able to change from the first target object shape, to the second, and so on, back and forth through the series of target object shapes chained together as "in-between" shapes. |

If off, the blending will occur in parallel. Each target object shape can influence the blending simultaneously in a parallel fashion rather than one after another in a series. Typically, for facial animation setup, you would want In-Between off so that you can have a variety of basic facial expressions that form the basis of all the possible expressions. Because the blending is in parallel, you can control the influence of each basic expression at any moment to get a nearly infinite variety of highly nuanced expressions. Default is off.

| | |
|---|---|
| Check Topology | Specifies whether to check if the base shape and the target shape(s) have the same topology. For example, if using NURBS objects, you could check if all the shapes have the same number of CVs. Click on or off. Default is on. |
| Delete Targets | Specifies whether to delete the target shape(s) after creation. Deleting target shapes can be useful if you don't need to see or manipulate the target shape(s) and you want to improve display performance. However, be sure to save a copy of the target shapes in case you later decide you need to adjust them. Default is off. |

## Advanced tab

### Deformation order

Specifies the placement of the deformer node in the deformable object's history.

| | |
|---|---|
| Default | Typically places the deformer immediately before the current final shape node. |
| Before | Places the deformer immediately before the current final shape node. Default and Before typically provide the same placement. |
| After | Places the deformer immediately after the current final shape node, and creates a new final shape node. |
| Split | Splits the input deformation history into two separate deformation chains, providing two final shapes originating from the same deformable object. |
| Parallel | Creates a final shape that blends the object's current input history in parallel with the new deformer. |

## Edit Blend Shape >

# Deform > Edit Blend Shape > Add

## Deform > Edit Blend Shape > Add > ❐

### Specify Node

If the base object shape you selected is influenced by only one blend shape deformer, you don’t need to turn Specify Node on. If on, you can specify BlendShape Node and Existing Nodes. Specify node is off by default.

### BlendShape Node

Specifies the name of the blend shape deformer to which you want to add the target object shapes. BlendShape Node is available only when Specify Node is on.

### Existing Nodes

Lists all the blend shape deformer nodes in the scene, and indicates the blend shape deformer to which you want to add target object shapes. Existing Nodes is available only when Specify Node is on.

### Add In-Between Target

Specifies whether you want to specify the Target Index and In-Between weight. Typically, you want to specify the Target Index and InBetween weight to control the effect of the target object shapes you are adding.

### Target Index

If the blend shape deformer blends target object shapes in parallel (the In-Between creation option was off when you created the blend shape deformer), you can add the new target object shapes so that they work in series with one of the existing target object shapes. One quick way you can identify the appropriate value for Target Index is by looking at the order of the target sliders in the Blend Shape Editor (Window > Animation Editors > Blend Shape). Note that in the editor, each target object shape has its own target slider. In the editor, going from left to right, the Target Index value for the left-most target slider would be 1, the next 2, and so on.

If the blend shape deformer blends target object shapes in series (the In-Between creation option was on when you created the blend shape deformer), Target Index can only be 1 because there is only one target slider. In this case, you don’t have to specify Target Index, but you do need to specify the In-Between Weight.

### In-Between Weight

Specifies the weight at which the added target object shape will have maximum influence. Use slider to select values from 0 to any value less than 1. Do not select 1 because 1 is the weight at which the existing target object shape has its maximum influence.

### Target Shape Options

Specifies whether to check if the added target object shapes have the same topology as the base object shape and the existing target object shape(s). For example, if using NURBS objects, you could check if all the shapes have the same number of CVs. Click Check Topology to turn it on or off. Check Topology is on by default.

# Deform > Edit Blend Shape > Remove

## Deform > Edit Blend Shape > Remove > ❐

### Specify Node

If the base object shape you selected is influenced by only one blend shape deformer, you don't need to set Specify Node on. If on, you can specify BlendShape Node and Existing Nodes. Default is off.

### BlendShape Node

Specifies the name of the blend shape deformer whose target object shapes you want to remove. (Available if Specify Node is on.)

### Existing Nodes

Lists all the blend shape deformer nodes in the scene, and indicates the blend shape deformer whose target object shapes you want to remove. Only available if Specify Node is on.

### Specify Target

When on, removes the specified target object shape from the current blend shape deformer. See "Target shapes, base shapes, and blend shapes" on page 238.

### Target Name

Lists the name of the target object shape to be removed from the current blend shape deformer. Only available only when Specify Target is on.

# Deform > Edit Blend Shape > Swap

## Deform > Edit Blend Shape > Swap > ❐

### Specify Node

If the base object shape you selected is influenced by only one blend shape deformer, you don’t need to set Specify Node on. If on, you can specify BlendShape Node and Existing Nodes. Default is off.

### BlendShape Node

Specifies the name of the blend shape deformer whose target object shapes you want to swap. Only available if Specify Node is on.

### Existing Nodes

Lists all the blend shape deformer nodes in the scene, and indicates the blend shape deformer whose target object shapes you want to swap. Only available if Specify Node is on.

# Deform > Create Lattice

## Deform > Create Lattice > ❐

### Basic tab

### Divisions

Specifies the structure of the lattice in the lattice’s local STU space. STU space provides a special coordinate system for specifying the structure of lattices.

You can specify the lattice’s structure in terms of S, T, and U divisions. When you specify the divisions, you also indirectly specify the number of lattice points in the lattice, because the lattice points are located where the divisions meet on the lattice’s exterior. The greater the number of divisions, the greater the lattice point resolution.

Though your control over the deformation increases with the number of lattice points, the performance may be affected.

The default settings are: S has 2 divisions, T has 5 divisions, and U has 2 divisions, which provides 20 lattice points.

### Local Mode

Specifies whether each lattice point can influence only the deformable object’s points that are nearby (local), or can influence all the deformable object’s points. Check on or off (default is on). If on, you can specify Local Divisions.

### Local Divisions

Only available if Local Mode is on. Specifies the extent of each lattice point's local influence in terms of the lattice's local STU space. The default settings are: S has 2 divisions, T has 2 divisions, and U has 2 divisions. With the default setting, each lattice point can only influence the deformable object's points that are at most two divisions away (in S, T, or U) from the lattice point.

### Positioning

Specifies whether the lattice is centered around the selected deformable object(s), or positioned at the workspace origin.

Typically you would want the lattice centered around the object(s) so that you can create deformation effects right after you create the deformer. However, you might want the object to be initially free of the lattice's influence, deforming only when it moves into the base lattice's space. For example, you might develop a ghost (the deformable object) that could squeeze through a keyhole-shaped influence lattice and then pop out on the other side, resuming its original shape.

Turn on Positioning to center the lattice and turn off Positioning to put the lattice at the workspace origin. Default is on.

### Grouping

Specifies whether to group the influence lattice and base lattice together. Grouping the influence lattice and base lattice lets you transform (move, rotate, or scale) the two together. Default is off. The influence lattice and base lattice are not grouped by default.

### Parenting

Specifies whether to parent the lattice to the selected deformable object(s) upon deformer creation. Parenting them lets you transform (move, rotate, or scale) them together. Default is off.

### Freeze Mode

Specifies whether to freeze the lattice deformation mapping. If frozen (checked on), components of objects being deformed that are inside the influence lattice remain fixed inside the lattice and affected only by the influence lattice, even if you transform (move, rotate, or scale) the object or the base lattice. For more information, see "Freezing the lattice deformation mapping" on page 246. Default is off.

### Outside Points Transform

Specifies the range of influence that the lattice deformer will have on its target object's points. Lets you transform all an object's points, even when parts of the object are outside of the lattice.

Transforming object points outside of the lattice is useful when you want to maintain a normal lattice deformation for objects that randomly pop outside of the lattice. For example, maintaining lattice deformations when chaining lattices together.

| | |
|---|---|
| Only If Inside Lattice | Only points within the base lattice are deformed. Only If Inside Lattice is on by default. |
| Transform All Points | All the target object's points—inside and outside the lattice—are deformed by the lattice. |
| Transform If Within Falloff | Points within the base lattice and up to the specified falloff distance are deformed by the lattice. For example, if you set the Falloff Distance to 2.0, then points within the base lattice and up to 2 widths of the lattice are deformed. |

#### Falloff Distance

Specifies the distance from the base lattice up to which points are affected by the lattice deformer. The units of falloff distance are measured in lattice widths. For example, a Falloff Distance value of 3.0 sets the falloff distance to 3 lattice widths.

The falloff degrades linearly from the edge of the base lattice to the specified falloff distance. This option is available only when Transform If Within Falloff is on.

### Advanced tab

See "Advanced deformer options" on page 403.

#### Deformation Order

Specifies the placement of the deformer node in the deformable object's history. For more information about deformer placement, see "Deformation order" on page 234.

#### Exclusive

Specifies whether the deformer set is in a partition. Sets in a partition can have no overlapping members. If on, the Exclusive Partition and Existing Partitions options become available. Default is off.

#### Partition To Use

Lists any existing partitions, and a default selection Create New Partition. If you select Create New Partition, you can edit the New Partition Name field to specify the name of a new partition. Only available if Exclusive is on.

#### New Partition Name

Specifies the name of a new partition that will include the deformer set. The suggested partition name is deformPartition, which will be created if it does not already exist. Typically, you might put all your exclusive deformer sets in the partition named deformParition. However, you can create as many partitions as you like, and name them whatever you want. Only available if Exclusive is on.

## Deform > Edit Lattice > Reset Lattice

You can reset the lattice by choosing Deform > Edit Lattice > Reset Lattice. However, you cannot change the resolution of a lattice if the lattice points have been moved from their reset position or the lattice has history.

## Deform > Edit Lattice > Remove Lattice Tweaks

If you want to change the number of divisions on a lattice whose points have been moved, choose Deform > Edit Lattice > Reset Lattice Tweaks, and then change the divisions of the lattice. If you want to change the divisions on a lattice with history, find the input lattice shape and change its divisions. You can find the input lattice by selecting the lattice and looking in the attribute editor tabs for the original lattice shape, which will typically share the same base name as the output lattice appended by "Orig."

## Deform > Create Cluster

### Deform > Create Cluster > ❐

#### Basic tab

#### Mode

Specifies whether the cluster deformation will occur only when the cluster deformer handle itself is transformed (moved, rotated, or scaled). With Relative turned on, only transformations to the cluster deformer handle itself will cause deformation effects. With Relative turned off, transformations to objects parented to the cluster deformer handle can cause deformation effects.

For example, suppose you are using a cluster deformer to smooth deformation effects around the wrist joint of a character's skinned arm. If you create a cluster deformer with Relative turned on, and then parent the cluster deformer handle to a wrist joint, you can rotate the shoulder joint without causing cluster deformation effects around the wrist. But when you move the cluster deformer handle itself, you cause cluster deformation effects around the wrist. Default is on.

#### Envelope

Specifies the deformation scale factor. A value of 0 provides no deformation, a value of 0.5 provides a deformation effect scaled to half of its full effect, and a value of 1 provides the full deformation effect. Use the slider to select values between 0 and 1. Default is 1.

### Advanced tab

For information on these creation options, see ”Advanced deformer options” on page 403.

#### Deformation Order

Specifies the placement of the deformer node in the deformable object’s history. For more information about deformer placement, see ”Deformation order” on page 234.

#### Exclusive

Specifies whether the deformer set is in a partition. Sets in a partition can have no overlapping members. If on, the Exclusive Partition and Existing Partitions options become available. Default is off.

#### Partition To Use

Lists any existing partitions, and a default selection Create New Partition. If you select Create New Partition, you can edit the New Partition Name field to specify the name of a new partition. Only available if Exclusive is on.

#### New Partition Name

Specifies the name of a new partition that will include the deformer set. The suggested partition name is deformPartition, which will be created if it does not already exist. Typically, you might put all your exclusive deformer sets in the partition named deformParition. However, you can create as many partitions as you like, and name them whatever you want. Only available if Exclusive is on.

## Create Nonlinear >

# Deform > Create Nonlinear > Bend

## Deform > Create Nonlinear > Bend > ❐

### Basic tab

#### Low Bound

Specifies lower extent of the bending along the bend deformer’s negative Y axis. Values can be negative numbers or zero. Values can be negative numbers or zero. Use slider to select values from -10.0000 to 0.0000. Default is -1.0000.

#### High Bound

Specifies upper extent of the bending along the bend deformer’s positive Y axis. Values can be positive numbers only (minimum is 0.0000). Use slider to select values from 0.0000 to 10.0000. Default is 1.0000.

#### Curvature

Specifies the amount of bending. Negative values specify the bending towards the bend deformer’s negative X axis. Positive values specify the bending towards the deformer’s positive X axis. Use slider to select values from -4.0000 to 4.0000. Default is 0.0000.

### Advanced tab

See ”Advanced deformer options” on page 403.

#### Deformation Order

Specifies the placement of the deformer node in the deformable object’s history. For more information about deformer placement, see ”Deformation order” on page 234.

#### Exclusive

Specifies whether the deformer set is in a partition. Sets in a partition can have no overlapping members. If on, the Exclusive Partition and Existing Partitions options become available. Default is off.

#### Partition To Use

Lists any existing partitions, and a default selection Create New Partition. If you select Create New Partition, you can edit the New Partition Name field to specify the name of a new partition. Only available if Exclusive is on.

#### New Partition Name

Specifies the name of a new partition that will include the deformer set. The suggested partition name is deformPartition, which will be created if it does not already exist. Typically, you might put all your exclusive deformer sets in the partition named deformParition. However, you can create as many partitions as you like, and name them whatever you want. Only available if Exclusive is on.

# Deform > Create Nonlinear > Flare

## Deform > Create Nonlinear > Flare > ❐

### Basic tab

#### Low Bound

Specifies the lower extent of the flare on the deformer's local negative Y axis. Values can be negative numbers or zero. Use the slider to select values from negative 10.0000 to 0.0000. Default is -1.0000.

#### High Bound

Specifies the upper extent of the flare on the deformer's positive local Y axis. Values can be positive numbers only (minimum value is 0). Use the slider to select values from 0.0000 to 10.0000. Default is 1.0000.

#### Start Flare X

Specifies the amount of flaring from the deformer's X axis at the Low Bound. The flaring progresses along the deformer's local X axis, varying according to the value of Curve. Use the slider to select values from 0.0000 to 10.0000. Default is 1.0000.

#### Start Flare Z

Specifies the amount of flaring from the deformer's Z axis at the Low Bound. The flaring progresses along the deformer's local Z axis to the High Bound, varying according to the value of Curve. Use the slider to select values from 0.0000 to 10.0000. Default is 1.0000.

#### End Flare X

Specifies the amount of flaring from the deformer's X axis at the High Bound. The flaring starts at the Low Bound and progresses along the deformer's local X axis to the High Bound, varying according to the value of Curve. Use the slider to select values from 0.0000 to 10.0000. Default is 1.0000.

### End Flare Z

Specifies the amount of flaring from the deformer's Z axis at the High Bound. The flaring starts at the Low Bound and progresses along the deformer's local Z axis to the High Bound, varying according to the value of Curve. Use the slider to select values from 0.0000 to 10.0000. Default is 1.0000.

### Curve

Specifies the amount of curvature (the profile of the flaring curve) between the Low Bound and the High Bound. A value of 0 specifies no curvature (linear interpolation). Positive values specify outward, bulging curvatures. Negative values specify inward, hourglass-shaped curvatures. Use the slider to select values from 0.0000 to 10.0000. Default is 0.0000.

## Advanced tab

See "Advanced deformer options" on page 403.

### Deformation Order

Specifies the placement of the deformer node in the deformable object's history. For more information about deformer placement, see "Deformation order" on page 234.

### Exclusive

Specifies whether the deformer set is in a partition. Sets in a partition can have no overlapping members. If on, the Exclusive Partition and Existing Partitions options become available. Default is off.

### Partition To Use

Lists any existing partitions, and a default selection Create New Partition. If you select Create New Partition, you can edit the New Partition Name field to specify the name of a new partition. Only available if Exclusive is on.

### New Partition Name

Specifies the name of a new partition that will include the deformer set. The suggested partition name is deformPartition, which will be created if it does not already exist. Typically, you might put all your exclusive deformer sets in the partition named deformParition. However, you can create as many partitions as you like, and name them whatever you want. Only available if Exclusive is on.

# Deform > Create Nonlinear > Sine

## Deform > Create Nonlinear > Sine > ❐

### Basic tab

#### Low Bound

Specifies the extent of the sine wave along the deformer's local negative Y axis. Values can be negative numbers or zero. Use the slider to specify values from negative 10.0000 to 0.0000. Default is -1.0000.

#### High Bound

Specifies the extent of the sine wave along the deformer's local positive Y axis. Values can be positive numbers only (minimum value is 0). Use the slider to specify values from 0.0000 to 10.0000. Default is 1.0000.

#### Amplitude

Specifies the amplitude (maximum wave amount) of the sine wave. Use the slider to specify values from -5.0000 to 5.0000. Default is 0.0000 (no wave).

#### Wavelength

Specifies the frequency of the sine wave along the deformer's local Y-axis. For greater frequency, decrease the wavelength. For lesser frequency, increase the wavelength. Use the slider to specify values from -0.1000 to 10.0000. Default is 2.0000.

#### Dropoff

Specifies how the amplitude decays. Negative values specify a decay towards the center of the deformer handle (minimum is -1.0000), and positive values specify a decay away from the center of the deformer handle (maximum is 1.0000). Use the slider to specify values from -1.0000 to 1.0000. Default is 0.0000 (no decay).

#### Offset

Specifies the location of the sine wave relative to the center of the deformer handle. Changing this value can create a wriggling effect. Use the slider to specify values from -10.0000 to 10.0000. Default is 0.0000.

### Advanced tab

See "Advanced deformer options" on page 403.

#### Deformation Order

Specifies the placement of the deformer node in the deformable object's history. For more information about deformer placement, see "Deformation order" on page 234.

#### Exclusive

Specifies whether the deformer set is in a partition. Sets in a partition can have no overlapping members. Check on or off (default is off). If on, the Exclusive Partition and Existing Partitions options become available.

#### Partition To Use

Lists any existing partitions, and a default selection Create New Partition. If you select Create New Partition, you can edit the New Partition Name field to specify the name of a new partition. Only available if Exclusive is on.

#### New Partition Name

Specifies the name of a new partition that will include the deformer set. The suggested partition name is deformPartition, which will be created if it does not already exist. Typically, you might put all your exclusive deformer sets in the partition named deformParition. However, you can create as many partitions as you like, and name them whatever you want. Only available if Exclusive is on.

# Deform > Create Nonlinear > Squash

## Deform > Create Nonlinear > Squash > ❐

### Basic tab

#### Low Bound

Specifies the lower extent of squashing (or stretching) along the deformer's local negative Y axis. Use the slider to select values from negative 10.0000 to 0.0000. Default is -1.0000.

#### High Bound

Specifies the upper extent of squashing (or stretching) along the deformer's local positive Y axis. Use the slider to select values from 0.0000 to 10.0000. Default is 1.0000.

#### Start Smoothness

Specifies the amount of initial smoothing toward the low bound position (along the deformer's local negative Y axis). Use slider to select values from 0.0000 to 1.0000. Default is 0.0000.

#### End Smoothness

Specifies the amount of final smoothing towards the high bound position (along the deformer's local positive Y axis). Use the slider to select values from 0.0000 to 1.0000. Default is 0.0000.

#### Max Expand Position

Specifies the center of maximum expansion between the high bound position and the low bound position. Values can be between 0.01000 (near the low bound position) and 0.9900 (near the high bound position). Use the slider to select values from 0.0100 to 0.9900. Default is 0.5000.

#### Expand

Specifies the amount of expansion outwards during squashing or inwards during stretching. Use the slider to select values from 0.0000 to 1.7000. Default is 1.0000.

#### Factor

Specifies the amount of squashing or stretching. Increasing negative values specify squashing along deformer's local Y axis. Increasing positive values specify stretching along deformer's local Y axis. Use the slider to select values from -10.0000 to 10.0000. Default is 0.0000 (no squashing or stretching).

### Advanced tab

See "Advanced deformer options" on page 403.

#### Deformation Order

Specifies the placement of the deformer node in the deformable object's history. For more information about deformer placement, see "Deformation order" on page 234.

#### Exclusive

Specifies whether the deformer set is in a partition. Sets in a partition can have no overlapping members. If on, the Exclusive Partition and Existing Partitions options become available. Default is off.

#### Partition To Use

Lists any existing partitions, and a default selection Create New Partition. If you select Create New Partition, you can edit the New Partition Name field to specify the name of a new partition. Only available if Exclusive is on.

#### New Partition Name

Specifies the name of a new partition that will include the deformer set. The suggested partition name is deformPartition, which will be created if it does not already exist. Typically, you might put all your exclusive deformer sets in the partition named deformParition. However, you can create as many partitions as you like, and name them whatever you want. Only available if Exclusive is on.

# Deform > Create Nonlinear > Twist

## Deform > Create Nonlinear > Twist > ❐

### Basic tab

#### Low Bound

Specifies the position of the start angle twisting on the deformer's local Y axis. Values must be negative numbers or zero. Use the slider to select values from -10.0000 to 0.0000. Default is -1.0000.

#### High Bound

Specifies the position of the end angle twisting on the deformer's local Y axis. Values must be positive numbers. Use the slider to select values from 0.0000 to 10.0000. Default is 1.0000.

#### Start Angle

Specifies the degree of twisting at the low bound position on the deformer handle's local negative Y axis. Use the slider to select values from -10.0000 to 10.0000. Default is 0.0000.

#### End Angle

Specifies the degree of twisting at the high bound position on the deformer handle's local positive Y axis. Use the slider to select values from -10.0000 to 10.0000. Default is 0.0000.

### Advanced tab

See "Advanced deformer options" on page 403.

#### Deformation Order

Specifies the placement of the deformer node in the deformable object's history. For more information about deformer placement, see "Deformation order" on page 234.

#### Exclusive

Specifies whether the deformer set is in a partition. Sets in a partition can have no overlapping members. If on, the Exclusive Partition and Existing Partitions options become available. Default is off.

#### Partition To Use

Lists any existing partitions, and a default selection Create New Partition. If you select Create New Partition, you can edit the New Partition Name field to specify the name of a new partition. Only available if Exclusive is on.

#### New Partition Name

Specifies the name of a new partition that will include the deformer set. The suggested partition name is deformPartition, which will be created if it does not already exist. Typically, you might put all your exclusive deformer sets in the partition named deformParitition. However, you can create as many partitions as you like, and name them whatever you want. Only available if Exclusive is on.

# Deform > Create Nonlinear > Wave

## Deform > Create Nonlinear > Wave > ❐

### Basic tab

#### Min Radius

Specifies the minimum radius of the circular sine wave. Use the slider to select values from 0.0000 to 1.0000. Default is 0.0000.

#### Max Radius

Specifies the maximum radius of the circular sine wave. Use the slider to select values from 0.0000 to 10.0000. Default is 1.0000.

#### Amplitude

Specifies the amplitude (maximum wave amount) of the sine wave. Use slider to select values from -5.0000 to 5.0000. Default is 0.0000 (no wave).

#### Wavelength

Specifies the frequency of the sine wave. For greater frequency, decrease the wavelength. For lesser frequency, increase the wavelength. Use the slider to select values from -0.1000 to 10.0000. Default is 1.0000.

#### Dropoff

Specifies how the amplitude decays. Negative values specify a decay towards the center of the deformer handle (minimum is -1.0000), and positive values specify a decay away from the center of the deformer handle (maximum is 1.0000). Use the slider to select values from -1.0000 to 1.0000. Default is 0.0000 (no decay).

#### Offset

Specifies the location of the sine wave relative to the center of the deformer handle. Changing this value can create a rippling effect. Use the slider to select values from -10.0000 to 10.0000. Default is 0.

### Advanced tab

See ”Advanced deformer options” on page 403.

#### Deformation Order

Specifies the placement of the deformer node in the deformable object’s history. For more information about deformer placement, see ”Deformation order” on page 234.

#### Exclusive

Specifies whether the deformer set is in a partition. Sets in a partition can have no overlapping members. If on, the Exclusive Partition and Existing Partitions options become available. Default is off.

#### Partition To Use

Lists any existing partitions, and a default selection Create New Partition. If you select Create New Partition, you can edit the New Partition Name field to specify the name of a new partition. Only available if Exclusive is on.

#### New Partition Name

Specifies the name of a new partition that will include the deformer set. The suggested partition name is deformPartition, which will be created if it does not already exist. Typically, you might put all your exclusive deformer sets in the partition named deformParitition. However, you can create as many partitions as you like, and name them whatever you want. Only available if Exclusive is on.

# Deform > Create Sculpt Deformer

## Deform > Create Sculpt Deformer > ❐

### Basic tab

#### Mode

Specifies the sculpt deformer's mode. Select Flip, Project, or Stretch. Default is Stretch.

| | |
|---|---|
| Flip | With Flip, a deformation occurs when the sculpt tool nears the target object's geometry. Also, Flip places the implicit locator at the center of the sculpt tool. This mode is called Flip because when the center of the sculpt tool passes through the surface, the deformed surface flips to the other side of the sculpt tool. |
| Project | With Project, the sculpt deformer projects the target objects geometry onto the surface of the sculpt tool. The extent to which the projection takes place depends on the sculpt deformer's Dropoff Distance attribute. Also, the Maximum Displacement attribute specifies whether the projection occurs directly on the sculpt tool, inside the sculpt tool, or outside of the sculpt tool. |
| Stretch | With Stretch, when you move the sculpt tool away from the geometry, the affected surface of the geometry stretches or bulges to stay with the sculpt tool. The stretch direction extends from the point marked by the stretch origin locator to the surface of the sculpt tool's shape. |

#### Inside Mode

Specifies how the deformer influences the deformable object's points located inside the sculpt sphere. There are two modes: Ring and Even.

| | |
|---|---|
| Ring | This mode pushes inside points outside of the sculpt sphere, creating a contoured, ring-like effect around the sculpt sphere. |
| Even | This mode spreads the inside points all around the sculpt sphere evenly, creating a smooth, spherical effect. Default is Even. |

#### Max Displacement

Specifies the distance that the sculpt sphere can push a deformable object's points from the sphere's surface. Use the slider to select values from -10.000 to 10.000. Default is 0.100.

#### Dropoff Type

Specifies how the sculpt sphere's range of influence declines or drops off. (The range of influence is specified with the Dropoff Distance.) There are two Dropoff Types: None and Linear.

None specifies no decline, providing a sudden dropoff effect.

Linear specifies a gradual decline, providing a dropoff effect that decreases linearly.

Select None or Linear. (Default is Linear.)

#### Dropoff Distance

Specifies the sculpt sphere's range of influence. (How the range of influence can decline is specified by Dropoff Type.)

#### Positioning

Specifies the placement of the sculpt sphere. Click Positioning on to center sculpt sphere within the deformable object. Click off to place sculpt sphere at the workspace origin. Note that if you are creating a stretch sculpt deformer (Mode is set to Stretch), the stretch origin locator will be placed with the sculpt sphere. Default is on, which centers the sculpt sphere within the deformable object.

#### Grouping

If you are creating a stretch sculpt deformer (Mode is set to Stretch), you can choose whether the stretch origin locator will be put in a group with the sculpt sphere. Click on to group the sculpt sphere with the stretch origin locator. Default is off.

#### Sculpt Tool

Lets you use a custom NURBS object as the sculpt deformer object. See "To use a NURBS surface with the sculpt deformer" on page 290.

### Advanced tab

See "Advanced deformer options" on page 403.

#### Deformation Order

Specifies the placement of the deformer node in the deformable object's history. For more information about deformer placement, see "Deformation order" on page 234.

Exclusive

Specifies whether the deformer set is in a partition. Sets in a partition can have no overlapping members. If on, the Exclusive Partition and Existing Partitions options become available. Default is off.

Partition To Use

Lists any existing partitions, and a default selection Create New Partition. If you select Create New Partition, you can edit the New Partition Name field to specify the name of a new partition. Only available if Exclusive is on.

New Partition Name

Specifies the name of a new partition that will include the deformer set. The suggested partition name is deformPartition, which will be created if it does not already exist. Typically, you might put all your exclusive deformer sets in the partition named deformParitition. However, you can create as many partitions as you like, and name them whatever you want. Only available if Exclusive is on.

## Deform > Soft Modification

The new Soft Modification deformer lets you manipulate groups of vertices at a time to modify the surface of a target geometry.

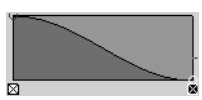
See also ”Soft Modification” on page 268 and ”Use the Soft Modification Tool” on page 291.

### Deform > Soft Modification > ❐

Falloff Radius

The radius that the deformation falls off to zero.

Falloff Curve



Controls the falloff of the modification.

Preserve History

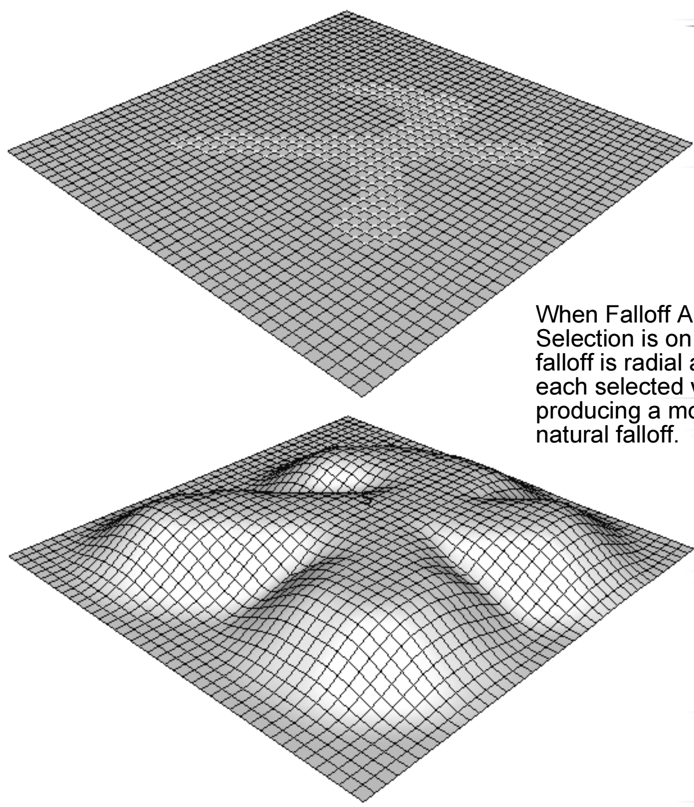
When Preserve History is on, all nodes that the Soft Modification Tool creates are maintained. If you want to animate the effect of the soft modification deformer, you will need to turn on Preserve History to ensure the deformation history is maintained.

When Preserve History is off, the Soft Modification Tool attempts to remove the deformation history and all additional nodes that it creates. If you are using Soft Modification as a modeling tool, and do not want modeling history, you should turn off Preserve History. (In order to generate the correct result, the Soft Modification Tool can only delete deformation history when the underlying shape does not have existing history. If deformer history nodes are still created even when Preserve History is off, try deleting history on the target shape and then try deforming it again. See "Example 2: Modeling without history" on page 293.)

### Mask Unselected

Controls whether all components are deformed (off) or only selected components are deformed (on).

### Falloff Around Selection

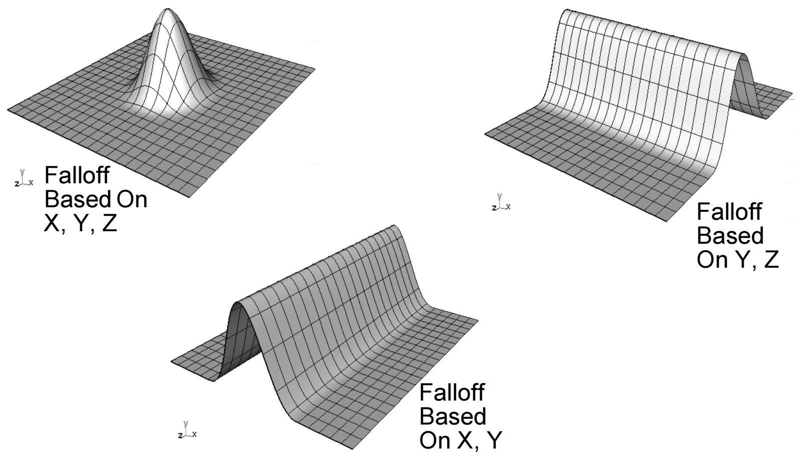



| | |
|---|---|
| on | The falloff is radial around each selected vertex. This can produce more natural falloff, especially when the selected vertices form an unusual shape. |

| | |
|---|---|
| off | The falloff is spherical around the center of the selection. |

Falloff Based On

The direction(s) in which the deformation falloff curve is applied. By turning off falloff along one or more directions, you can create various effects (for example, ramps, edges).




## Notes

- Soft Modification is like a deformer in that it works on clusters.
- You can control the effect of the soft modification by adjusting the Soft Modification Options, the soft modification manipulators, and the softMod node attributes.
- The Soft Modification Tool is similar to the Modify > Transformation Tools > Proportional Modification Tool. However, the Soft Modification Tool provides history and more control (for example, over falloff). The Proportional Modification Tool only works on components. The Soft Modification Tool works on components, objects or multiple objects. The Proportional Modification Tool only works on multiple NURBS surfaces. The Soft Modification Tool works on multiple surfaces of any surface type. The Soft Modification Tool can be used with the Deform > Edit Membership Tool.

# Deform > Create Jiggle Deformer

## Deform > Create Jiggle Deformer > ❐

### Basic tab

#### Stiffness

Sets the rigidity of the jiggle, from 0 to 1. High values diminish elasticity, speed up the jiggle, and cause the points to act as if controlled by tight springs. Low values slow the jiggle and create an effect like spongy springs.

#### Damping

Mutes the springiness of the jiggle. A high value minimizes jiggle. A low value increases springiness.

#### Weight

Scales the jiggle effect up or down on all points, regardless of individual weights. The individual jiggle weights do not change value, only the overall amount of jiggle.

| | |
|---|---|
| Jiggle Only After Object Stops | Jiggle occurs only after a moving object stops, not while it moves. |
| Ignore Transform | Jiggle applies only to animated points, not to the animated transform node of the object. For example, suppose you have animated a kangaroo hopping as it talks. You animated the hopping motion by keying the translate attributes of the transform node, and you animated the moving mouth by keying the points of the mouth. With Ignore Transform on, only the talking mouth jiggles. |

### Advanced tab

The options on the Advanced tab are common to all deformers. See "Advanced deformer options" on page 403.

# Deform > Create Jiggle Disk Cache

## Deform > Create Jiggle Disk Cache > ❐

### Cache Time Range

Specifies which frames are cached. It is easiest to cache the entire frame range of the Time Line. However, you can create smaller cache files to conserve disk space by caching only the frames where objects jiggle.

| | |
|---|---|
| Time Slider | Caches all frames of the Playback Start/End range. |
| Render Globals | Caches the frame range specified in the Image File Output section of the Render Globals window. |

For more information, search the online help for Render Globals.

| | |
|---|---|
| Start/End | Caches the frame range specified in the Start Time and End Time boxes. |
| Start Time<br>End Time | Sets the cached frame range. Available only if you select Custom for the Cache Frame Range. |

### Sampling

Oversampling and undersampling specify how often Maya calculates jitter per frame.

| | |
|---|---|
| Over | A Rate value of 2 or larger might increase the precision of the cached jiggle in scenes where a jiggling object collides with a rigid body quickly and repeatedly. |
| Under | A Rate value of 2 or larger decreases the precision of the cached jiggle, but quickens the caching operation. An Under Sample Rate of 2, for instance, means Maya calculates jitter once every two frames.<br><br>If a jiggling object collides unrealistically with a rigid body, select Over Sample and set a Rate that matches the Over Samples setting available in Solvers > Edit Oversampling or Cache Settings. |

### Rate

Sets an integer value for the Over Sample or Under Sample.

# Deform > Jiggle Disk Cache Attributes

See "Jiggle Disk Cache nodes" on page 420.

# Deform > Wire Tool

To create a wire deformer, use the Wire Tool. The characteristics of the wire deformer you create depend on the Wire Tool's tool settings. By default, the Wire Tool is set to create a wire deformer without holders.

## Deform > Wire Tool > ❒

### Wire Settings

#### Holders

If on, the wire deformer is created with a holder. If off, the wire deformer is created without a holder. Holders are curves that you can use to limit the deformation region. Default is off.

#### Envelope

Specifies the deformation scale factor. Use the slider to select a value between 0.0000 and 2.0000. A value of 0 specifies no deformation effect. Default is 1.0000.

#### Crossing Effect

Specifies the amplitude of the deformation effect where two of the deformer's influence wires cross. Use the slider to select values from 0.0000 to 2.0000. Default is 0.0000, which specifies a smooth, not additive, effect.

#### Local Influence

Specifies the localization of the deformation effect of two or more influence wires. Use the slider to select values from 0.0000 to 2.0000. Default is 0.0000.

#### Dropoff Distance

Specifies the range of influence of each influence wire. Use the slider to select values from 0.0000 to 10.0000. Default is 1.0000.

#### Deformation Order

Specifies the placement of the deformer node in the deformable object's history. Placement selections include: Default, Before, After, Split, or Parallel.

| | |
|---|---|
| Default | Places the deformer immediately before the current final shape node. |
| Before | Places the deformer immediately before the current final shape node. Default and Before typically provide the same placement. |

| | |
|---|---|
| After | Places the deformer as the output of the current final shape node, and creates a new final shape node. |
| Split | Splits the input deformation history into two separate deformation chains, providing two final shapes originating from the same deformable object. |
| Parallel | Creates a final shape that blends the object's current input history in parallel with the new deformer. |

Exclusive

Specifies whether the deformer set will be in a partition. If a deformer set is in a partition, the points in the set cannot be in any other set. The result is that only the deformer you are about to create can influence the points. If on, the Exclusive Partition and Existing Partitions options become available. By default, Exclusive is off.

Exclusive Partition

Specifies the name of the partition. The default name is deformPartition. Only available if Exclusive is on.

Existing Partitions

Specifies an existing partition. The default existing partition is characterPartition. Only available if Exclusive is on.

## Edit Wire >

# Deform > Edit Wire > Add

Lets you add the selected curve to the current wire deformer. See ”To add an influence wire” on page 303.

## Deform > Edit Wire > Add > ❐

Lets you specify the name of the wire deformer to which you want to add the current curve.

Specify Node

Turns on the Wire Node field and Existing Node drop-down menu.

Wire Node

Specifies the name of the wire deformer. Also, if you select the wire deformer's name from the Existing Node drop-down menu, the name of the deformer you select appears in this field.

#### Existing Node

Lists the current wire deformers in your scene. The name of the wire deformer you select from this list appears in the Wire Node field.

## Deform > Edit Wire > Remove

Lets you remove the selected curve from the current wire deformer. See "To remove an influence wire" on page 304.

### Deform > Edit Wire > Remove > ❐

Lets you specify the name of the wire deformer from which you want to remove the current curve.

#### Specify Node

Turns on the Wire Node field and Existing Node drop-down menu.

#### Wire Node

Specifies the name of the wire deformer. Also, if you select the wire deformer's name from the Existing Node drop-down menu, the name of the deformer you select appears in this field.

#### Existing Node

Lists the current wire deformers in your scene. The name of the wire deformer you select from this list appears in the Wire Node field.

## Deform > Edit Wire > Add Holder

Lets you add the selected curve as a holder to the current wire deformer. See "Holders" on page 273 and "Limiting the wire deformation region" on page 277.

### Deform > Edit Wire > Add Holder > ❐

Lets you specify the name of the wire deformer to which you want to add the current curve as a holder.

#### Specify Node

Turns on the Wire Node field and Existing Node drop-down menu.

#### Wire Node

Specifies the name of the wire deformer. Also, if you select the wire deformer's name from the Existing Node drop-down menu, the name of the deformer you select appears in this field.

#### Existing Node

Lists the current wire deformers in your scene. The name of the wire deformer you select from this list appears in the Wire Node field.

## Deform > Edit Wire > Reset

Lets you reset the current wire deformer curve so that it no longer contributes to the deformation. See ”To reset influence wires” on page 304.

### Deform > Edit Wire > Reset > ❒

Lets you specify the name of the wire deformer whose curve you want to reset.

#### Specify Node

Turns on the Wire Node field and Existing Node drop-down menu.

#### Wire Node

Specifies the name of the wire deformer. Also, if you select the wire deformer’s name from the Existing Node drop-down menu, the name of the deformer you select appears in this field.

#### Existing Node

Lists the current wire deformers in your scene. The name of the wire deformer you select from this list appears in the Wire Node field.

## Deform > Edit Wire > Show Base Wire

Displays the base wire of the current wire deformer curve. See ”Influence wires and base wires” on page 273 and ”To display the base wire” on page 304.

## Deform > Edit Wire > Parent Base Wire

Creates a wires group that includes that influence wire and base wire. See ”Creating wires groups that parent influence wires to base wires” on page 276 and ”To create a wire group” on page 304.

### Deform > Edit Wire > Parent Base Wire > ❒

Lets you specify the name of the wire deformer for whose curve you want to generate a group.

#### Specify Node

Turns on the Wire Node field and Existing Node drop-down menu.

#### Wire Node

Specifies the name of the wire deformer. Also, if you select the wire deformer's name from the Existing Node drop-down menu, the name of the deformer you select appears in this field.

#### Existing Node

Lists the current wire deformers in your scene. The name of the wire deformer you select from this list appears in the Wire Node field.

## Deform > Wire Dropoff Locator

Sets the current wire deformer curve point as a wire dropoff locator. See "Using wire dropoff locators for localized deformation effects" on page 276 and "To add a wire dropoff locator" on page 350.

### Deform > Wire Dropoff Locator > ❐

Lets you specify the name of the wire deformer for which you want to create a wire dropoff locator.

#### Wire Node

Specifies the name of the wire deformer. Also, if you select the wire deformer's name from the Existing Node drop-down menu, the name of the deformer you select appears in this field.

#### Existing Node

Lists the current wire deformers in your scene. The name of the wire deformer you select from this list appears in the Wire Node field.

#### Envelope

Specifies the deformation scale factor for the wire dropoff locator. This attribute corresponds with the Locator Envelope channel.

#### Percent

Specifies the local effect that the wire dropoff locator has on the influence wire's dropoff. By default, the influence wire has two implicit locators at each end with a default Percent of 1.000, and the other locators have an effect relative to the Percent of the implicit locators.

# Deform > Wrinkle Tool

## Wrinkle Tool Settings

### Type

Specifies the type of wrinkle deformer. Select Tangential, Radial, or Custom. Default is Radial.

| | |
|---|---|
| Tangential | Combine influence wires that are roughly parallel. A tangential wrinkle deformer can only deform a single NURBS surface. |
| Radial | Combines influence wires that branch from a single point, like spokes on a wheel. A radial wrinkle deformer can only deform a single NURBS surface. |
| Custom | Combines influence wires you created in the fashion that best suits the effect you would like to make. A custom wrinkle deformer can deform any deformable object (NURBS surfaces, polygonal surfaces, lattices and so on). |

### Amount

Specifies the number of parent influence wires in the wrinkle deformer. The total number of influence wires can also include child influence wires specified by Radial Branch Amount and Radial Branch Depth. Use the slider to select value between 0 and 20. Default is 3.

### Thickness

Specifies the surface dropoff. The surface dropoff is the area influenced by each influence wire. Use the slider to select values from 0.0000 to 2.0000. Default is 1.0000.

### Randomness

Specifies how close the wrinkle deformer conforms to the specified Amount, Intensity, Radial Branch Amount, and Radial Branch Depth. Use the slider to select values from 0.0000 to 1.0000. Default is 0.2000.

### Intensity

Specifies the sharpness of the creases created by the influence wires. The minimum intensity (0) specifies smooth creases. The maximum intensity (1) specifies sharp, steep creases. Use slider to select values from 0.0000 to 1.0000. Default is 0.5000.

### Radial Branch Amount

Specifies the number of child influence wires that branch from each parent influence wire. Applies to radial wrinkle deformers only. Use slider to select values from 0 to 10. Default is 2.

### Radial Branch Depth

Specifies the depth of the influence wire hierarchy, which is the number of levels of child influence wires that branch from each parent influence wire. Increasing the Radial Branch Depth exponentially increases the total number of influence wires. Applies to radial wrinkle deformers only. Use slider to select values from 0 to 4. Default is 0.

# Deform > Create Wrap

## Deform > Create Wrap > Create Wrap > ❐

### Weight Threshold

Specifies the influence of the wrap influence objects' shapes based on the proximity of their components to the objects being deformed. Depending on the point density (for example, the number of CVs) of the wrap influence objects, changing the Weight Threshold can change the overall smoothness of the deformation effect. Values range from 0.000 (smooth) to 1.000 (sharp). Use the slider to select values from 0.000 to 1.000. Default is 0.000.

### Limit Influence Area

Click Use Max Distance on to set the Max Distance.

### Max Distance

Specifies the influence area of wrap influence object points. By limiting the influence area with Max Distance, you can limit how much memory Maya requires to perform the deformation. The less memory required, the better the performance. Using Max Distance is especially useful when you are working with high-resolution wrap influence objects.

Max Distance's value is in Maya's working units, which are by default centimeters (Window > Settings/Preferences > Preferences > Settings > Working Units). The default for Max Distance is 0, which provides the default performance of the wrap deformer. The default value of 0 does not specify no influence area. A value of 0 specifies that each point has an *infinite* influence area, and the influence area is constrained by the Weight Threshold attribute. However, a Max

Distance setting such as 0.1 would greatly limit the influence area to within a distance of 0.1 units from each point. Such a setting would require less memory than 0 or a setting greater than 0.1.

| | |
|---|---|
| **Note** | Weight Threshold takes effect within the influence area indicated by Max Distance. |

Maya allocates memory for the wrap deformer when it is created. You can change Max Distance after you create the wrap deformer. That can also improve performance, but for best results try to set the desired value as a creation option. Of course, deciding on the best Max Distance value may require some experimentation. In general, for best performance, you'll want to try minimize Max Distance to some value greater than 0. Note that setting Max Distance to 0 would be better than setting it to a relatively large number (for example, 30), and then lowering the Weight Threshold as desired. This is because a relatively large number will cause more memory allocation than the default setting of 0, even though 0 specifies an infinite influence area. To find the best value, start with 0 or a very small value, and then increase the Max Distance until you reach the value you want.

## Deform > Edit Wrap > Add Influence

Adds the selected object as a new influence for the specified wrap deformation.

## Deform > Edit Wrap > Remove Influence

Removes the selected influence object from the specified wrap deformation.

## Deform > Display Intermediate Objects

Turns on the visibility for the current object's original (prior to deformation) shape. You can use intermediate objects as points-of-reference when deforming objects. For example, you can compare a deformed object and its intermediate object to determine how different the deformed object is from its original shape. See "Displaying and hiding intermediate objects" on page 236.

## Deform > Hide Intermediate Objects

Turns off the visibility for the current object's original (prior to deformation) shape.

## Deform > Paint Cluster Weights Tool

Lets you use the Paint Cluster Weights Tool to set cluster weights simply by painting over the clustered surface. For more information, see:

- ”Cluster deformers” on page 250
- ”Understanding cluster deformers” on page 250
- ”Editing cluster deformation effects” on page 251
- ”Painting cluster weights” on page 252
- ”Painting weights on restricted areas” on page 253
- ”Flooding clusters” on page 253
- ”Mapping weight values to clusters” on page 253

See also *“How Artisan brush tools work”* in the Paint Effects, Artisan, and 3D Paint guide.

### Deform > Paint Cluster Weights Tool > ❐

| Tip | You can define hotkey combinations to change most of the settings without opening the Tool Settings editor. For details on setting hotkey combinations, see *Define Artisan hotkeys* in the Paint Effects, Artisan, and 3D Paint guide. |
|---|---|

Lets you specify the settings for the Paint Cluster Weights Tool in the Tool Settings editor. In the Paint Attributes section there are some attributes unique to the Paint Cluster Weights Tool. These unique attributes are described below along with some of the common attributes, and how they apply to this tool. For descriptions of all other attributes in all other sections, see *“Common Artisan Brush Tool Settings”* in the Paint Effects, Artisan, and 3D Paint guide.

#### Paint Attributes section

These are descriptions of the attributes in the Paint Attributes section.

cluster*n*.weights
jointCluster*n*.weights

Displays the name of the cluster selected to paint and the attribute you are painting (weights). To select another cluster to paint, click this button and select the appropriate cluster weights name. By default, the tool selects the first cluster it detects on the surface (for example, cluster1.weights, or jointCluster3.weights).

### Filter: cluster

Sets a filter so that only cluster nodes display on the menu for the button above this one. You are painting clusters with the Paint Cluster Weights Tool, so you do not need to change this filter.

### Paint Operation

Select which paint operation you want to perform on the selected cluster.

| | |
|---|---|
| Replace | Your brush stroke replaces the vertex weight with the weight set for the brush. |
| Add | Your brush stroke adds the vertex weight to the weight set for the brush. |
| Scale | Your brush stroke scales the vertex weight by the weight factor set for the brush. |
| Smooth | Your brush stroke averages the weights of adjacent vertices to produce a smoother transition between weights. |

### Value

Set the weight value to apply when you perform any of the painting operations.

### Min/Max Value

Set the minimum and maximum possible paint values. By default, you can paint values between 0 and 1. Setting the Min/Max Values you can extend or narrow the range of values.

Negative values are useful for subtracting weight. For example, if you set Min Value to -1, Value to -0.5, and select Add for the operation, you would subtract 0.5 from the weight of vertices you paint.

Positive values are used as multipliers.

> Tip: To help you differentiate paint values when you paint with ranges greater than 0 to 1 (for example, -5 to 5), and to maximize the range of values that display when you paint values with ranges between 0 to 1 (for example, 0.2 to 0.8), set Min Color and Max Color (in the Display section) to correspond with the Min/Max values.

### Clamp

Select whether you want to clamp the values within a specified range, regardless of the Value set when you paint.

| | |
|---|---|
| Lower | Turn this on to clamp the lower value to the Clamp Value specified below. For example, if you clamp Lower and set the lower Clamp Value to 0.5, the values you paint will never be less than 0.5, even if you set the Value to 0.25. |
| Upper | Turn this on to clamp the upper value to the Clamp Value specified below. For example, if you clamp Upper, set the upper Clamp Value to 0.75, and set Value to 1, the values you paint will never be greater than 0.75. |

#### Clamp Values

Set the Lower and Upper values for clamping.

#### Flood

Click Flood to apply the brush settings to all the weights on the selected cluster. The result depends on the brush settings defined when you perform the flood.

#### Vector Index

If you are painting a three channel attribute (RGB or XYZ), select the channel you want to paint. Cluster weight is a single channel attribute, therefore you do not need to change this setting.

## Deform > Paint Jiggle Weights Tool

Lets you use the Paint Jiggle Weights Tool to tune the jiggle of individual points by painting their jiggle weight values. For more information, see "Creating Jiggle deformers" on page 269 and "Set jiggle deformers" on page 299. See also *"How Artisan brush tools work"* in the Paint Effects, Artisan, and 3D Paint guide.

### Deform > Paint Jiggle Weights Tool > ❐

Lets you specify the settings for the Paint Jiggle Weights Tool in the Tool Settings editor. In the Paint Attributes section there are some attributes unique to the Paint Jiggle Weights Tool. These unique attributes are described below along with some of the common attributes, and how they apply to this tool. For descriptions of all other attributes in all other sections, see *"Common Artisan Brush Tool Settings"* in the Paint Effects, Artisan, and 3D Paint guide.

#### Paint Attributes

These are descriptions of the attributes in the Paint Attributes section.

### jiggle*n*.weights

Displays the name of the jiggle node selected to paint and the attribute you are painting (weights). To select another deformer to paint, click this button and select the appropriate jiggle deformer weights name. By default, the tool selects the first jiggle deformer it detects on the surface (for example, jiggle2.weights).

### Filter: jiggle

Sets a filter so that only jiggle deformer nodes display on the menu for the button above this one. You are painting jiggle weights with the Paint Jiggle Weights Tool, so you do not need to change this filter.

### Paint Operation

Select which paint operation you want to perform on the selected jiggle deformer.

| | |
|---|---|
| Replace | Your brush stroke replaces the painted weight with the weight set for the brush. |
| Add | Your brush stroke adds the painted weight to the weight set for the brush. |
| Scale | Your brush stroke scales the painted weight by the weight factor set for the brush. |
| Smooth | Your brush stroke averages the weights of adjacent vertices to produce a smoother transition between weights. |

### Value

Set the weight value to apply when you perform any of the painting operations.

### Min/Max Value

Set the minimum and maximum possible paint values. By default, you can paint values between 0 and 1. Setting the Min/Max Values you can extend or narrow the range of values.

Negative values are useful for subtracting weight. For example, if you set Min Value to -1, Value to -0.5, and select Add for the operation, you would subtract 0.5 from the weight of vertices you paint.

Positive values are used as multipliers.

| Tip | To help you differentiate paint values when you paint with ranges greater than 0 to 1 (for example, -5 to 5), and to maximize the range of values that display when you paint values with ranges between 0 to 1 (for example, 0.2 to 0.8), set Min Color and Max Color (in the Display section) to correspond with the Min/Max values. |
|---|---|

#### Clamp

Select whether you want to clamp the values within a specified range, regardless of the Value set when you paint.

| | |
|---|---|
| Lower | Turn this on to clamp the lower value to the Clamp Value specified below. For example, if you clamp Lower and set the lower Clamp Value to 0.5, the values you paint will never be less than 0.5, even if you set the Value to 0.25. |
| Upper | Turn this on to clamp the upper value to the Clamp Value specified below. For example, if you clamp Upper, set the upper Clamp Value to 0.75, and set Value to 1, the values you paint will never be greater than 0.75. |

#### Clamp Values

Set the Lower and Upper values for clamping.

#### Flood

Click Flood to apply the brush settings to all the weights on the selected jiggle deformer. The result depends on the brush settings defined when you perform the flood.

#### Vector Index

If you are painting a three channel attribute (RGB or XYZ), select the channel you want to paint. Jiggle deformer weight is a single channel attribute, therefore you do not need to change this setting.

## Deform > Paint Set Membership Tool

Lets you use the Paint Set Membership Tool to modify which of a deformable object’s points (for example, CVs or vertices) belong to multiple deformer sets by painting the points you want added to, transferred to, or removed from the set. See ”Paint deformer set membership” on page 315. See also *“How Artisan brush tools work”* in the Paint Effects, Artisan, and 3D Paint guide.

## Deform > Paint Set Membership Tool > ❐

Lets you specify the settings for the Paint Set Membership Tool in the Tool Settings editor. In the Paint Operations section there are some attributes unique to the Paint Set Membership Tool. These unique attributes are described below. For descriptions of all other attributes in all other sections, see *"Common Artisan Brush Tool Settings"* in the Paint Effects, Artisan, and 3D Paint guide.

### Paint Operations

These are descriptions of the attributes in the Paint Operations section.

#### Operation

**Add** If the object includes multiple deformer sets, the Add operation leaves the painted CVs or vertices in the their current sets and adds them to the selected set.

If the object is a rigid skin object, the Add operation does the same thing as the Transfer operation: it removes the painted CVs or vertices from their current set and adds them to the selected set.

**Transfer** If the object includes multiple deformer sets, the Transfer operation removes the painted CVs or vertices from their current sets and adds them to the selected set.

If the object is a rigid skin object, the Transfer operation does the same thing as the Add operation: it removes the painted CVs or vertices from their current set and adds them to the selected set.

**Remove** The Remove operation removes the painted CVs or vertices from the sets they belong to, so the CVs vertices are not influenced by any deformers or joints.

#### Set Membership

**Select Set To Modify** Click the name of the set you want to add to, transfer to, or remove from. This is just one way of selecting the set. There are two others. For details, see "Paint deformer set membership" on page 315.

**Set To Modify** The name of the set you select appears in this box.

# Deform > Point On Curve

## Deform > Point On Curve > ❐

### Keep Original

If on, Point On Curve retains the original curve shape (default name: lsqModCurve*n*). Default is off.

### Point Weight

Specifies how much influence the point on curve locator constraint should have on the curve’s shape relative to other point on curve locator constraints. Set values from 0.1000 (least influence) to 1.0000 (most influence). Default is 1.0000.

# Advanced deformer options

## Deformation Order

| | |
|---|---|
| Default | Maya places the deformer before (as an Input) the deformed shape.<br>Default is the same as Before unless the deformer is going to act on a shape node with no history. In this case, the order will be the same as After.<br>When you create a number of deformers for an object with Default, the result is a deformation chain whose order is the same as the order in which you created the deformers. |
| Before | Maya places the deformer immediately before of the deformable object’s deformed shape. In the object’s history, the deformer will be placed right before the deformed shape. |
| After | Maya places the deformer immediately after the deformable object. You can use After to create an intermediate deformed shape somewhere in the middle of the object’s history. With After, the original shape of the object is not hidden. |
| Split | Maya splits the deformation into two deformation chains. You can use Split to deform an object in two ways at the same time, creating two final shapes that originate from the same original shape. |
| Parallel | Maya places the deformer in parallel with the existing input nodes in the object’s history, and then blends the effects provided by the existing input |

nodes and the deformer. A parallel blender node (default name: parallelBlender*n*) that blends the effects of the existing input nodes and the new deformer is placed before the final shape.

Parallel is useful when you want to blend the influences of several deformers acting on a single object. For example, if you create a bend deformer with Default placement for an object, and then create a Sine deformer with Parallel placement, you can directly control how much influence each deformer has on the object by blending the influence of each deformer. The parallelBlender node provides a weight channel for each deformer. You can edit the channels of the parallel blender node.

Front Of Chain — Front Of Chain is only available for blend shape deformers. Blend shape deformers are typically used to create deformation effects on a skinned character. Front Of Chain assures that blend shape deformation effects occur before the deformation effects provided by the skinning. If the effects occur afterwards, undesirable double transformation effects occur when the skeleton is posed.

Front Of Chain places the deformer in front of all the deformer and skinning nodes in the deformable object's shape history, but not before any of the tweak nodes. Tweak nodes let you point tweak objects being deformed. See "Point tweaking objects" on page 235.

### Exclusive

Specifies whether the deformer set is in a partition. Sets in a partition can have no overlapping members. If on, the Exclusive Partition and Existing Partitions options become available. Default is off.

### Partition To Use

Lists any existing partitions, and a default selection Create New Partition. If you select Create New Partition, you can edit the New Partition Name field to specify the name of a new partition. (Available if Exclusive is on.)

### New Partition Name

Specifies the name of a new partition that will include the deformer set. The suggested partition name is deformPartition, which will be created if it does not already exist. Typically, you might put all your exclusive deformer sets in the partition named deformPartition. However, you can create as many partitions as you like, and name them whatever you want. Only available if Exclusive is on.

# Nodes

## Character nodes

## General deformer node attributes

### Deformer Attributes

#### Envelope

Specifies the deformation scale factor. Select values from 0 to 1. You can also enter values from -2 to 2. A value of 2 would double the overall deformation effect. A negative value would invert the effect. Default is 1.

### Node Behavior

#### Caching

Specifies that Maya store the results of input evaluations, and then provide those results to the node. This saves Maya from having to re-evaluate the input nodes every time the node needs the results. If there are no changes to the input nodes, then this setting can improve display performance with no loss of results. However, note that caching uses more memory than would otherwise be used, which might adversely affect performance. Also, if there are changes to input nodes, more memory is allocated and then freed during each deformation, which might also adversely affect display performance.

#### Node State

| | |
|---|---|
| Normal | Specifies that Maya evaluate and display the deformation. Maya will evaluate the node as usual. This is the default. |
| HasNoEffect | Specifies that Maya prevent the deformation, but display the object. Maya will evaluate the nodes in the node's history, but not the node itself. |
| Blocking | Specifies that Maya prevent the deformation, and not display the object. Maya will not report the results of any evaluations of input nodes to this node. |

| | |
|---|---|
| Waiting-Normal | Automatically set by Maya. Specifies that if the dependency graph evaluation refresh performance setting (Window > Settings/ Preferences > Performance Settings) is set to Demand or Release, the node takes the Normal state when you click Update or release the mouse button. |
| Waiting-HasNoEffect | Automatically set by Maya. Specifies that if the dependency graph evaluation refresh performance setting is set to Demand or Release, the node takes the HasNoEffect state when you click Update or release the mouse button. |
| Waiting-Blocking | Automatically set by Maya. Specifies that if the dependency graph evaluation refresh performance setting is set to Demand or Release, the node will take the Blocking state when you click Update or release the mouse button. |

### Extra Attributes

No extra attributes by default.

## tweak

Stores transformation information for the vertices of the target object. Tweak also ensures the proper transformation of vertices when you deform the target object. When you delete the tweak node, all the transformation information for the vertices that are affected by the deformation is erased. Deleting the tweak node is an easy way of removing deformations from an object.

### **Blend Shape deformer nodes**

## blendShape

### Blend Shape Attributes

#### Origin

Specifies whether the blend shape is relative to the base object shape's position, rotation, and scale, or is directly specified by you.

| | |
|---|---|
| World | Blends the base object shape to the target object shape(s), taking into account any differences in position, rotation, and scale between the target object shape(s). |

**Note** The local and world selections are identical to the Origin creation option's selections. For more information, see "Deform > Create Blend Shape" on page 364. However, the User attribute is not one of the Origin creation option's selections.

| | |
|---|---|
| Local | Blends the base object shape to the target object shape(s) while ignoring differences in position, rotation, and scale between the target shape(s). |
| User | Specifies that two special attributes, baseOrigin and targetOrigin, provide origin information. For more information on these attributes, see the online Node and Attribute Reference and the MEL Command reference. You can use the setAttr MEL command to set the values of the baseOrigin and targetOrigin attributes. |

## Lattice deformer nodes

# ffdLatticeShape

## Lattice History

### S Divisions

Specifies the number of S divisions in the influence lattice's structure. Default is 2.

### T Divisions

Specifies the number of T divisions in the influence lattice's structure. Default is 5.

### U Divisions

Specifies the number of U divisions in the influence lattice's structure. Default is 2.

# ffd

## Freeform Deformation Attributes

### Local

Specifies whether each lattice point can influence only the deformable object's points that are nearby (Local on), or can influence all the deformable object's points (Local off). If on, you can specify Local Influence S, Local Influence T, and Local Influence U. Default is on.

### Local Influence S

Specifies the extent of each lattice point's local influence along the S axis of the lattice's local STU space. Use slider to select values from 2 to 30. The default is 2.

### Local Influence T

Specifies the extent of each lattice point's local influence along the T axis of the lattice's local STU space. Use slider to select values from 2 to 30. The default is 2.

### Local Influence U

Specifies the extent of each lattice point's local influence along the U axis of the lattice's local STU space. Use slider to select values from 2 to 30. The default is 2.

### Partial Resolution

Specifies whether Maya calculates the deformation with full resolution or partial resolution. Default is full.

If you don't need to see the deformation at full resolution, you can improve display performance by reducing the accuracy of the deformation. To do so, select partial. With partial selected, use the Partial Resolution slider to specify the deformation calculation's accuracy. A tolerance of 0 means you want full accuracy; the maximum value of 0.1 decreases the accuracy significantly. Select values from to 0.000 to 0.100. Default is 0.010.

If you have set the accuracy to partial, set the accuracy of each deformer back to full before you render the scene if you want to render the deformation with full accuracy.

### Freeze Geometry

Specifies whether to freeze the lattice deformation mapping. If frozen (checked on), components (for example, CVs) of objects being deformed that are inside the influence lattice become fixed inside the lattice and affected only by the influence lattice, even if you transform (move, rotate, or scale) the object or the base lattice. For more information, see "Freezing the lattice deformation mapping" on page 246. Default is off.

## Cluster deformer nodes

# cluster

## Cluster Attributes

### Relative

Specifies whether the cluster deformation occurs only when the cluster deformer handle itself is transformed (moved, rotated, or scaled). With Relative on, only transformations to the cluster deformer handle itself cause deformation effects. Transformations to any objects parented to the handle do not cause deformation effects. With Relative off, transformations to objects parented to the cluster deformer handle can cause deformation effects. The Relative attribute was initially set by the Mode creation option when you created the cluster deformer. See "Mode" on page 371.

### Partial Resolution

Specifies whether Maya provides the complete deformation, or only an approximation of the deformation. Selections include full and partial. Full specifies the complete deformation. Partial specifies an approximation of the deformation, which can improve Maya's display performance. With partial, Maya rounds down the cluster weights based on the Percent Resolution. Default is full.

### Percent Resolution

Specifies the increment percentage by which the cluster deformation resolution is rounded down. Maya uses the increment percentage to round off the cluster weights to the next lowest increment. For example, with a Percent Resolution of 5.00, a cluster weight of .94 would be rounded down to .90. A cluster weight of .46 would be rounded down to .45. Default is 5.00. Only available only if Partial Resolution is set to partial.

### Angle Interpolation

Specifies the interpolation direction. Use this attribute to correct undesirable discontinuities in the deformation effect when you change cluster weights even by a small amount. The discontinuities occur when the cluster deformer uses an inappropriate interpolation direction to guide the deformation effect. To change the interpolation direction, you can set Angle Interpolation to closest, positive, or negative. By default, Angle Interpolation is closest, which provides the usual rigid skinning deformation effects. The default setting is fine for most situations, but when you encounter discontinuities you can adjust the deformation effect by selecting a positive or negative interpolation.

## clusterHandle

This node represents the handle through which you can select the cluster in the scene and manipulate its transformations. This node also stores the transformation information for the cluster's deformation.

### Origin

Displays where (X,Y,Z co-ordinates) the “C” icon appears in the deformed object’s local space.

### Weighted Node

This attribute is connected to the weighted node (the one to which the cluster percentages are applied), if one was specified. Otherwise, the transform node above this node is the one to which the percentages are applied.

**Nonlinear deformer nodes**

## bend

### Nonlinear Deformer Attributes

| | |
|---|---|
| Curvature | Specifies the amount of bending. Negative values specify the bending towards the bend deformer’s negative X axis and positive values specify the bending towards the deformer’s positive X axis. Use the slider to select values from -4.0000 to 4.0000. Default is 0.0000, which specifies no bending. |
| Low Bound | Specifies the lower extent of the bending along the bend deformer’s negative Y axis. Values can be negative numbers or zero. Values can be negative numbers or zero. Use the slider to select values from -10.0000 to 0.0000. Default is -1.0000. |
| High Bound | Specifies the upper extent of the bending along the bend deformer’s positive Y axis. Values can be positive numbers only (minimum is 0.0000). Use the slider to select values from 0.0000 to 10.0000. Default is 1.0000. |

# flare

## Nonlinear Deformer Attributes

| | |
|---|---|
| Low Bound | Specifies lower extent of the flare on the deformer's local negative Y axis. Values can be negative numbers or zero. Use the slider to select values from negative 10.000 to 0.000. Default is -1.000. |
| High Bound | Specifies upper extent of the flare on the deformer's positive local Y axis. Values can be positive numbers only (minimum value is 0). Use the slider to select values from 0.000 to 10.000. Default is 1.000. |
| Start Flare X | Specifies the amount of flaring from the deformer's X axis at the Low Bound. The flaring progresses along the deformer's local X axis, varying according to the value of Curve. Use the slider to select values from 0.000 to 10.000. Default is 1.000. |
| Start Flare Z | Specifies the amount of flaring from the deformer's Z axis at the Low Bound. The flaring progresses along the deformer's local Z axis to the High Bound, varying according to the value of Curve. Use the slider to select values from 0.000 to 10.000. Default is 1.000. |
| End Flare X | Specifies the amount of flaring from the deformer's X axis at the High Bound. The flaring starts at the Low Bound and progresses along the deformer's local X axis to the High Bound, varying according to the value of Curve. Use the slider to select values from 0.000 to 10.000. Default is 1.000. |
| End Flare Z | Specifies the amount of flaring from the deformer's Z axis at the High Bound. The flaring starts at the Low Bound and progresses along the deformer's local Z axis to the High Bound, varying according to the value of Curve. Use the slider to select values from 0.000 to 10.000. Default is 1.000. |
| Curve | Specifies the amount of curvature (the profile of the flaring curve) between the Low Bound and the High Bound. A value of 0 specifies no curvature (linear interpolation). Positive values specify outward, bulging curvature and negative values specify inward, hourglass-shaped curvatures. Use the slider to select values from -3.000 to 3.000. Default is 0.000. |

# sine

## Nonlinear Deformer Attributes

| | |
|---|---|
| Low Bound | Specifies the extent of the sine wave along the deformer's local negative Y axis. Values can be negative numbers or zero. Use the slider to select values from negative 10.000 to 0.000. Default is -1.000. |
| High Bound | Specifies the extent of the sine wave along the deformer's local positive Y axis. Values can be positive numbers only (minimum value is 0.000). Use the slider to select values from 0.000 to 10.000. Default is 1.000. |
| Amplitude | Specifies the amplitude (maximum wave amount) of the sine wave. Use the slider to select values from -5.000 to 5.000. Default is 0.000 (no wave). |
| Wavelength | Specifies the frequency of the sine wave along the deformer's local Y axis. For greater frequency decrease the wavelength, and for lesser frequency increase the wavelength. Use the slider to select values from 0.100 to 10.000. Default is 2.000. |
| Offset | Specifies the location of the sine wave relative to the center of the deformer handle. Changing this value can create a wriggling effect. Use the slider to select values from -10.000 to 10.000. Default is 0.000. |
| Dropoff | Specifies how the amplitude decays. Negative values specify a decay towards the center of the deformer handle (minimum is -1.000), and positive values specify a decay away from the center of the deformer handle (maximum is 1.000). Use the slider to select values from -1.000 to 1.000. Default is 0.000 (no decay). |

# squash

## Nonlinear Deformer Attributes

| | |
|---|---|
| Low Bound | Specifies the lower extent of squashing (or stretching) along the deformer's local negative Y axis. Use the slider to select values from -10.000 to 0.000. Default is -1.000. |

| | |
|---|---|
| High Bound | Specifies the upper extent of squashing (or stretching) along the deformer's local positive Y axis. Use the slider to select values from 0.000 to 10.000. Default is 1.000. |
| Factor | Specifies the amount of squashing or stretching. Increasing negative values specify squashing along deformer's local Y axis and increasing positive values specify stretching along deformer's local Y axis. Use the slider to select values from -10.000 to 10.000. Default is 0.000 (no squashing or stretching). |
| Expand | Specifies the amount of expansion outwards during squashing or inwards during stretching. Minimum value is 0 and maximum value is 10. Use the slider to select values from 0.000 to 10.000. Default is 1.000. |
| Max Expand Position | Specifies the center of maximum expansion between the high bound position and the low bound position. Values can be between 0.010 (near the low bound position) to 0.990 (near the high bound position). Use the slider to select values from 0.010 to 0.990. Default is 0.5. This attribute corresponds with the Max Expand Position channel. |
| Start Smoothness | Specifies the amount of initial smoothing towards the low bound position (along the deformer's local negative Y axis). Use the slider to select values from 0.000 to 1.000. Values can be from 0.000 to 1.000. Default is 0.000. |
| End Smoothness | Specifies the amount of final smoothing towards the high bound position (along the deformer's local positive Y axis). Use the slider to select values from 0.000 to 1.000. Default is 0.000. |

# twist

## Nonlinear Deformer Attributes

| | |
|---|---|
| Low Bound | Specifies the position of the start angle twisting on the deformer's local Y axis. Use the slider to select values from -10.000 to 0.000. Default is -1.000. |
| High Bound | Specifies the position of the end angle twisting on the deformer's local Y axis. Use the slider to select values from 0.000 to 10.000. Default is 1.000. |

| | |
|---|---|
| Start Angle | Specifies the degree of twisting at the low bound position on the deformer handle's local negative Y axis. Use the slider to select values from -859.437 to 859.437. Default is 0.000. |
| End Angle | Specifies the degree of twisting at the high bound position on the deformer handle's local positive Y axis. Use the slider to select values from -859.437 to 859.437. Default is 0.000. |

# wave

## Nonlinear Deformer Attributes

| | |
|---|---|
| Amplitude | Specifies the amplitude (maximum wave amount) of the sine wave. Use the slider to select values from -5.000 to 5.000. Default is 0.000 (no wave). |
| Wavelength | Specifies the frequency of the sine wave. For greater frequency decrease the wavelength, and for lesser frequency increase the wavelength. Use the slider to select values from 0.100 to 10.000. Default is 1.000. |
| Offset | Specifies the location of the sine wave relative to the center of the deformer handle. Changing this value can create a rippling effect. Use the slider to select values from -10.000 to 10.000. Default is 0.000. |
| Dropoff | Specifies how the amplitude decays. Negative values specify a decay towards the center of the deformer handle (minimum is -1.000), and positive values specify a decay away from the center of the deformer handle (maximum is 1.000). Use the slider to select values from -1.000 and 1.000. Default is 0.000 (no decay). |
| Min Radius | Specifies the minimum radius of the circular sine wave. Minimum value is 0.000 and Maximum value is 1.000. Use the slider to select values from 0.000 to 10.000. Default is 0.000. |
| Max Radius | Specifies the maximum radius of the circular sine wave. Minimum value is 0.000 and Maximum value is 1.000. Use the slider to select values from 0.000 to 10.000. Default is 1.000. |

## Sculpt deformer nodes

# sculpt

## Sculpt History

### Mode

Specifies the sculpt deformer's mode.

| | |
|---|---|
| Flip | With Flip, a deformation occurs when the sculpt tool nears the target object's geometry. Also, Flip places the implicit locator at the center of the sculpt tool. This mode is called Flip because when the center of the sculpt tool passes through the surface, the deformed surface flips to the other side of the sculpt tool. |
| Project | With Project, the sculpt deformer projects the target objects geometry onto the surface of the sculpt tool. The extent to which the projection takes place depends on the sculpt deformer's Dropoff Distance attribute. Also, the Maximum Displacement attribute specifies whether the projection occurs directly on the sculpt tool, inside the sculpt tool, or outside of the sculpt tool. |
| Stretch | With Stretch, when you move the sculpt tool away from the geometry, the affected surface of the geometry stretches or bulges to stay with the sculpt tool. The stretch direction extends from the point marked by the stretch origin locator to the surface of the sculpt tool's shape. |

### Inside Mode

Specifies how the deformer influences the deformable object's points located inside the sculpt sphere.

| | |
|---|---|
| Ring | This mode pushes inside points outside of the sculpt sphere, creating a contoured, ring-like effect around the sculpt sphere. |
| Even | This mode spreads the inside points all around the sculpt sphere evenly, creating a smooth, spherical effect. Default is Even. |

### Max Displacement

Specifies the distance that the sculpt sphere can push a deformable object's points from the sphere's surface. Use the slider to select values from -10.000 to 10.000. Default is 0.100. Max Displacement uses the Working Units that you set in the Maya Settings and Preferences.

### Dropoff Type

Specifies how the sculpt sphere's range of influence declines or drops off. The range of influence is specified with the Dropoff Distance.

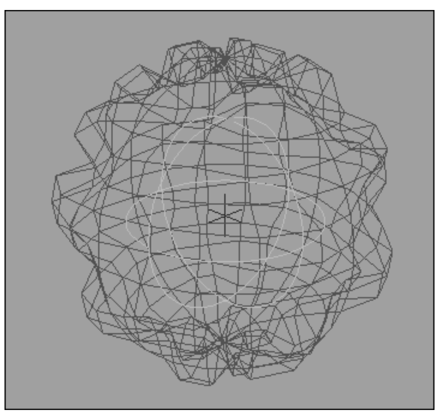
| | |
|---|---|
| None | Specifies no decline, providing a sudden dropoff effect. |
| Linear | Specifies a gradual decline, providing a dropoff effect that decreases linearly. |

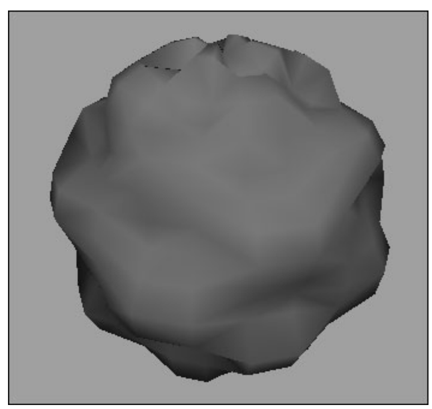
### Dropoff Distance

Specifies the sculpt sphere's range of influence. Note that Dropoff Type specifies how the range of influence can decline. The effect can depend on the deformer's Mode attribute setting. For instance, if Mode is set to Project, see "Project mode" on page 265.

## Add Advanced Sculpt Attribute

With the Advanced Sculpt Attributes you can use 2 dimensional textures to control the strength of the deformation field that surrounds the Sculpt object. This lets you deform geometry in a non-uniform manner.





### Enable Advanced

When on, the Advanced Sculpt Deformer Attributes are active. When off, the Advanced Sculpt Deformer Attributes are inactive.

### Texture

Lets you set a 2D texture to drive the strength of the sculpt deformer. The texture coordinates are assigned to the sculpt object in the same manner as a similarly oriented NURBS object.

### Texture Multiplier

Sets the strength of the 2D texture's influence on the deformation.

### Non Uniform

When on, Maya normalizes the sculpt deformation. Turn on this attribute if you want to compensate for any non-uniform scaling that may have been applied to the target object or the sculpt deformer.

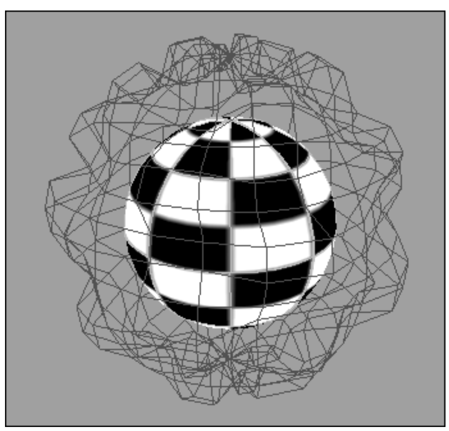
### Paint Resolution

If you want to use Paint Sculpt Map to paint deformation strengths for the sculpt deformer, then you can use these fields to set the resolution values for the file texture.

To change the resolution, you must first disconnect the existing file texture, change the resolution setting, and then re-click the Paint Sculpt Map button.

### Visualize Texture

When on, the objects influenced by the deformer are displayed in wireframe, and the deformer itself is displayed in textured mode. This lets you see the texture that is being used, and is useful when painting deformation textures.



#### Paint Sculpt Map

Opens the 3D paint tool for you to paint deformation strengths over the surface of the deformer. Also, Paint Sculpt Map creates a new file texture and connects it to the sculpt deformer’s Texture attribute. If a non-file texture is connected to the deformer, clicking this button will disconnect it.

For more information, search the online help for 3D Paint Tool.

## Soft modification nodes

### softMod

See softMod in Help > Node and Attribute Reference.

## Jiggle deformer nodes

### jiggle

#### Jiggle Attributes

Enable

| | |
|---|---|
| Enable | The jiggle deformer is on. Enable is the default setting. |
| Disable | The jiggle deformer is off. |
| Enable Only After Object Stops | Sets that jiggling occurs only after a moving object stops, not while it moves. |

Stiffness

See ”Stiffness” on page 387.

Damping

See ”Damping” on page 387.

Jiggle Weight

See ”Weight” on page 387.

Force Along Normal

Sets how much jiggling occurs in directions normal to the surface.

Force On Tangent

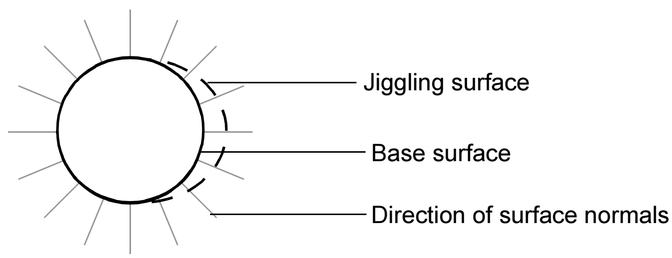
Sets how much jiggling occurs in directions tangent to the surface.

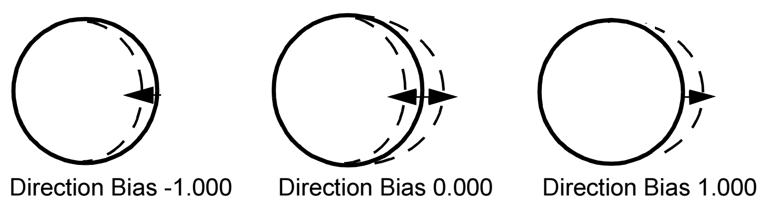
### Motion Multiplier

If you select Enable Only After Object Stops (see “Enable”), the Motion Multiplier scales how much the object jiggles after it stops moving.

### Direction Bias

Sets the direction of a surface’s jiggle relative to the direction of its normals. A Direction Bias of 1.000 causes a jiggle deformed surface to move only in the direction of its normals and a Direction Bias of -1.000 causes a jiggle deformed surface to move only in the opposite direction of its normals. A Direction Bias of 0.000 causes the jiggle deformed surface to move in both directions. The default Direction Bias value is 0.000.




You can use the Direction Bias attribute to prevent any unwanted collapsing of a jiggle deformed surface. For example, if you are animating the jiggly stomach of a sumo wrestler, you can set the Direction Bias attribute’s value to 1.000 so that the surface of the character’s stomach will only jiggle outward.

### Ignore Transform

See ”Ignore Transform” on page 387.

### Disk Cache Attributes

#### Disk Cache

If you have created jiggle cache for the jiggle animation controlled by this jiggle node, this field displays the name of the jiggle cache node. You can click the arrow button to the right of the name to view the jiggle cache settings. See "To create disk cache for jiggle deformers" on page 301.

| | |
|---|---|
| Warning | To avoid unexpected deformations, do not change the number of a deformable object's points after you create a jiggle deformer. |

## Jiggle Disk Cache nodes

# jiggleCache

### Jiggle Cache Attributes

#### Cache Name

Name of the cache of the current jiggle deformer.

#### Start Time

The playback time at which the jiggle cache data begins.

#### End Time

The playback time at which the jiggle cache data ends.

#### Sampling Type

| | |
|---|---|
| Over | Samples the jiggle more than once a frame. For example, an Over value of 3 causes Maya to sample the jiggle 3 times a frame. |
| Under | Samples the jiggle less than once a frame. For example, an Under value of 3 causes Maya to sample the jiggle every 3 frames. |

#### Sampling Rate

Specifies the number of samples per frame.

#### Delete Cache

Deletes the cache of the current jiggle deformer.

## Caching Node

### Disk Cache

Displays the name of the current jiggle deformer.

# Wire deformer nodes

## wire

## Parameters

**Rotation** Specifies an effect that varies between shearing and tangency. Values range from 0 and 1, with 0 indicating maximum shearing and 1 maximum tangency. Use the slider to select values from 0.000 to 1.000. Default is 1.000.

**Crossing Effect** Specifies the amplitude of the deformation effect when two influence wires cross. Values can vary from 0 (smooth effect) to 1 (additive effect). Default is 0. Use the slider to select values from 0.000 to 1.000. Default is 0.000.

**Local Influence** Specifies the localization of the deformation effect of two or more influence wires. Values range from 0 and 1, with 1 specifying greatest localization. Use the slider to select values from 0.000 to 1.000. Default is 0.000.

**Tension** Specifies the influence wire's attraction strength. Values specify how strongly the influence wire can attract the deformable object's points away from the base wire. The attraction strength is therefore expresses a tension between the influence wire and the base wire. Use the slider to select values from -10.000 to 10.000. Default is 1.000.

**Freeze Geometry** Specifies whether to freeze the wire deformation effect. If frozen (turned on), components (for example, CVs) of objects being deformed that are under the influence of the influence wire become fixed and affected only by the influence wire, even if you transform (move, rotate, or scale) the object or the base wire. The reason you would want to freeze geometry is to improve performance. Note that you should not move geometry objects relative to the base wire with freeze geometry checked on. Default is off.

## Scale

Specifies the scale of influence for each influence wire. Default is 1.00.

## Dropoff Distance

Specifies the range of influence for each influence wire. Default is 1.00.

## Locators

If you have created any wire dropoff locators, this section lists the attributes for each wire dropoff locator on each influence wire.

### curveShape*n*->locator*n*

Identifies the influence wire shape (default name: curveShape*n*) and the wire dropoff locators (default name: locator*n*) on that wire. The locators are numbered in the order that they were created, starting with 1.

### Param[*n*]

Specifies the location of the wire dropoff locator on the influence wire curve. The value of *n* indicates which wire dropoff locator, numbered in the order created, starting with 0. The value is in terms of the curve's U parameter.

### Percent[*n*]

Specifies the local effect the locator has on the influence wire's dropoff. The value of *n* represents a specific wire dropoff locator, numbered in the order created, starting from 0. By default, the influence wire has two implicit locators at each end with a Percent of 1.000. Other locators have an effect relative to the Percent of those locators. Use the slider to select values from 0.000 to 1.000. Default is 1.000.

### Twist[*n*]

Specifies localized twisting effects around a wire dropoff locator. The value of *n* represents a specific wire dropoff locator, numbered in the order created, starting with 0. As you increase or decrease the value, twisting of the region of the surface influenced by the wire dropoff locator occurs. In character setup, changing Twist can provide subtle effects for lip and eyebrow action. Default is 0, which specifies no twisting effects. This attribute corresponds with the Wire Dropoff Twist channel.

Envelope[*n*]

Specifies the deformation scale factor for a wire dropoff locator. The value of *n* represents a specific wire dropoff locator, numbered in the order created, starting with 0. Default is 1. This attribute corresponds with the Locator Envelope channel.

## Wire Dropoff Locator nodes

# dropoffLocator

## Dropoff Locator Attributes

Along with the Percent, Param, and Local Position attributes listed here, two other attributes are available to control the deformation effects of a wire dropoff locator: the Twist attribute and the Envelope attribute. These attributes are available as general deformer node attributes. See "General deformer node attributes" on page 405.

Percent

Specifies the local effect the locator has on the influence wire's dropoff. By default, the influence wire has two implicit locators at each end with a Percent of 1.000. Other locators have an effect relative to the Percent of those locators. Use the slider to select values from 0.000 to 1.000. Default is 1.000. Note that this attribute is also available as an attribute of the wire deformer.

Param

Specifies the location of the wire dropoff locator on the influence wire curve. The value is in terms of the curve's U parameter. Note that this attribute is also available as an attribute of the wire deformer.

Local Position

Specifies the local position of the wire dropoff locator on the influence wire curve.

## Wrap deformer nodes

# wrap

## Wrap Attributes

Weight Threshold

Specifies the influence of the wrap influence objects' shapes based on the proximity of their components to the objects being deformed. Depending on the point density (for example, the number of CVs) of the wrap influence objects, changing the Weight Threshold can change the overall smoothness of the deformation effect. Values range from 0.000 (sharp) to 1.000 (smooth). Use the slider to select values from 0.000 to 1.000. Default is 0.000.

Max Distance

Specifies the influence area of wrap influence object points. By limiting the influence area with Max Distance, you can limit how much memory Maya requires to perform the deformation. The less memory required, the better the performance. Using Max Distance is especially useful when you are working with high-resolution wrap influence objects.

### Point On Curve deformer nodes

## leastSquaresModifier

### Least Squares Modifier Attributes

Input NURBS Object

Displays the NURBS curve the least squares modifier node is affecting (for example, curveShape*n*Original).

### Point Constraints

pointConstraint[*n*]

Identifies a point on curve locator constraint. The value of *n* corresponds with the default locator name minus 1. For example, pointConstraint[0] corresponds with locator1.

| | |
|---|---|
| U | Specifies the location of the point on curve locator constraint on the NURBS curve in terms of the curve’s U parameter. |
| Weight | Specifies how much influence the point on curve locator constraint has on the curve’s shape relative to other point on curve locator constraints. Use slider to select values from 0.100 to 1.000. Default is 1.000. The Weight attribute is initially specified by the Point Weight point on curve option. |

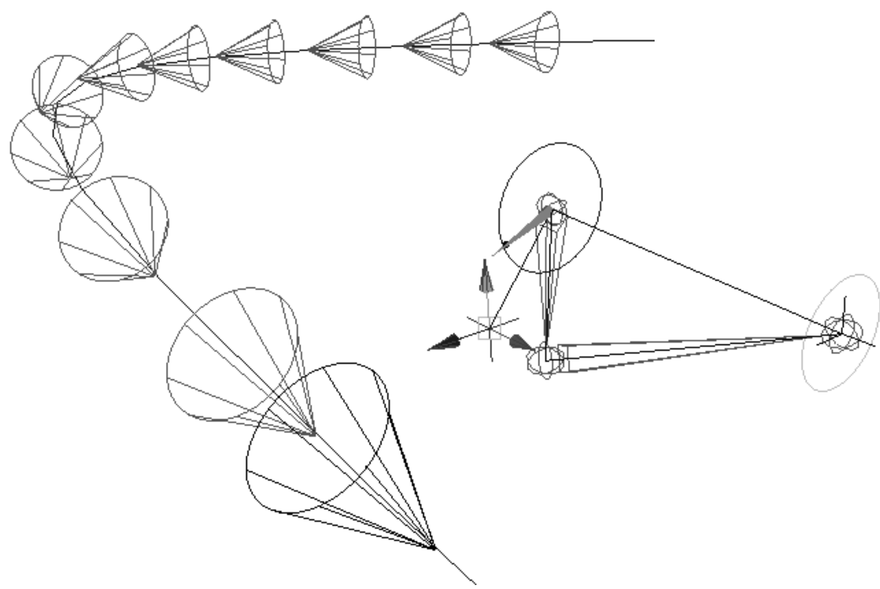
# 5 Constraints

## About

### Introducing Constraints

## Introducing Constraints

Constraints enable you to constrain the position, orientation, or scale of an object to other objects. Further, with constraints you can impose specific limits on objects and automate animation processes.



For example, if you want to quickly animate a sled sliding down a bumpy hill, you might first use a geometry constraint to constrain the sled to the surface. You could then use a normal constraint to make the sled sit flat on the surface. After you create these constraints, you key the sled's positions at the top and bottom of the hill. The animation is then complete.

Maya includes eight types of constraints for character setup and animation:

Note that other software packages use the term "animation controllers" to refer to what Maya calls constraints.

## Constraint node behavior

You don’t need to know about constraint node behavior in order to use constraints effectively. If you are new to constraints, you can skip this section. However, familiarity with constraint node behavior can provide you with more control over constraint manipulation and performance.

For each object in your scene, if there has been any change to its node or any of the nodes in its history (its input or output nodes), Maya will evaluate the nodes and update the display based on the node’s node behavior attributes. The node behavior attributes for constraint nodes can affect how constraint effects are evaluated and displayed.

## Constrained and target objects

A constrained object is an object whose position, orientation, and so on is driven by the nearest surface location, the direction of a target vector, or the position of one or more target objects. The position of one or more target objects is called the target point. The orientation of one or more target objects is called the target orientation.

## Target point

The target point is the position of the target object’s rotate pivot. If there is more than one target object, the target point is the average position of the rotate pivots of all the target objects. If you are using more than one target object, you can vary the influence of each target object on the calculation of the target point. The target point can be a weighted average of the positions of the target objects, with some target objects having more influence than others. The influence of target objects on the weighted average is specified by target object weights. You can change the target point by moving each target object’s rotate pivot.

## Target orientation

The target orientation is the orientation (Rotate X, Y, and Z attributes) of the target object. If there is more than one target object, the target orientation is the average orientation of all the target objects. However, if you are using more than one target object, you can vary the influence of each target object on the calculation of the target orientation. The target orientation can be a weighted average of the orientations of the target objects, with some target objects having more influence than others. The influence of target objects on the weighted average is specified by target object weights.

### Target scale

The target scale is the scaling (Scale X, Y, and Z attributes) of the target object. If there is more than one target object, the target scale is the average scaling of all the target objects. However, if you are using more than one target object, you can vary the influence of each target object on the calculation of the target scale. The target scale can be a weighted average of the scales of the target objects, with some target objects having more influence than others. The influence of target objects on the weighted average is specified by target object weights.

### Target vector

The target vector, or weighted average vector, represents the normal vector at the position of the constrained object. Maya calculates the target vector as a weighted average of the nearby normal vectors on the surface or mesh.

### Target object weights

For each target object, you can specify a target object weight that controls that object's influence in the calculation of the target point, orientation, scale, vector and so on. The resulting weighted average drives the constrained object's position, orientation, and so on. When a target object's weight is 0, the target has no influence over the constrained object. When the target object's weight is 1, the target has full influence over the constrained object.

**Note**

- Constraint weights are useful only when there are multiple target objects.
- When there is only one target object, any weight that is greater than 0 is interpreted as 1. Therefore, the single target is interpreted as having 100% influence over the constrained object.

### Constrained object's position

The constrained object's position is driven by the target point. However, you can offset the constrained object's position from the target point. Offsetting the constrained object's position from the target point can be useful in situations where you don't want the local axis of the constrained object to coincide exactly with the target point. For example, if you want to constrain a ball to a joint in a character's hand so that the hand holds the ball, you'll need to offset the ball from the joint. By offsetting, you can have the ball in the palm of the hand rather than centered inside the hand.

### Constrained object's orientation

The constrained object's orientation is driven by the target orientation.

### Constrained object's scaling

The constrained object's scaling is driven by the target scale.

### Workflow summary

Creating constraints can be as simple as selecting the objects you want to constrain with, selecting the object you want to constrain, and then selecting the appropriate constraint from the Constrain menu. Using constraints can become more complicated as you seek to go beyond the default options for constraints.

Some constraints lock the some of the channels of constrained objects. For example, the aim constraint locks the orientation channels (Rotate X, Y, and Z) of the object it constrains. Which channels get locked dictates how you can you use more than one constraint on an object. For a given object, you can use either an aim constraint, normal constraint, or tangent constraint because each of these constraints locks the orientation channels of a constrained object.

Attributes locked by constraints can also preclude the use of expressions on those attributes. If the locked attributes are on joints, those locked attributes can prevent the skeleton from returning to its bind pose.

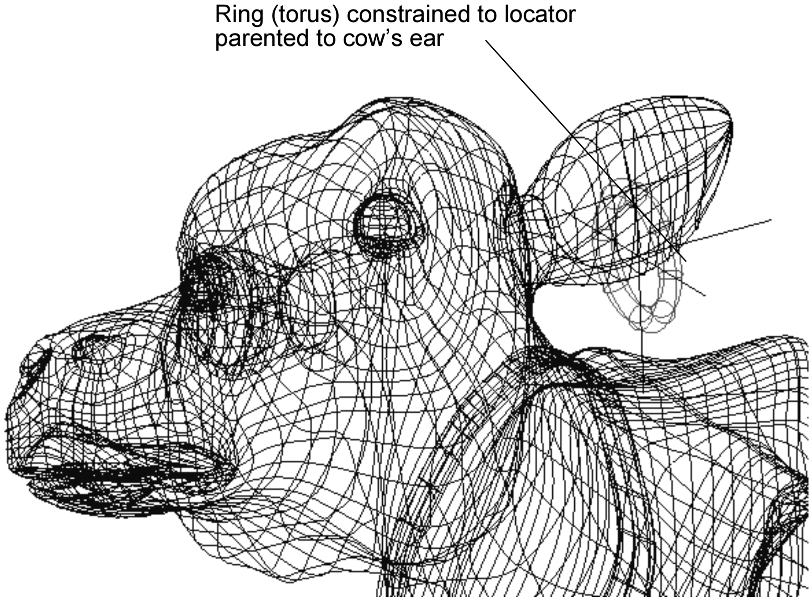
**Point constraints**

## Point constraints

A point constraint causes an object to move to and follow the position of an object, or the average position of several objects. This is useful for having an object match the motion of other objects.

You can also use a point constraint to animate one object to follow a series of objects.

See ”Create point constraints” on page 446 and ”Constrain > Point” on page 488.

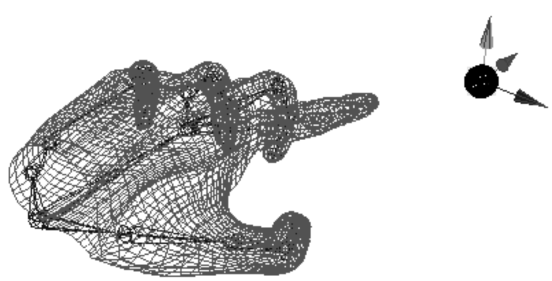



## Aim constraints

# Aim constraints

An aim constraint constrains an object's orientation so that the object aims at other objects. Typical uses of the aim constraint include aiming a light or camera at an object or group of objects. In character setup, a typical use of an aim constraint is to set up a locator that controls eyeball movement.

See "Create aim constraints" on page 448 and "Constrain > Aim" on page 489.



## Constrained object's orientation for aim constraints

The constrained object's orientation is controlled by three vectors: the aim vector, the up vector, and the world up vector. These vectors are not displayed in the workspace, but you can infer their effect on the constrained object's orientation.

You do not need to understand the details of how these vectors work in order to use constraints effectively. If you are new to constraints, you can skip the rest of this section. However, if you want to exercise a high degree of control over an aim constraint, you'll need to work with these vectors. Also, familiarity with these vectors can help you to understand how a constrained object can suddenly roll.

### Aim vector

The aim vector constrains the constrained object so that it always points at the target point. The aim vector starts at the constrained object's pivot point and always points at the target point.

How the object rotates to point at the target point depends on how the aim vector is defined relative to the object's local space. For instance, by default, the aim vector is defined so that it points in the same direction as the local rotation positive X-axis. Consequently, by default, a constrained object's local rotation positive X-axis will point at the target point.

By itself, the aim vector does not completely constrain the object, because the aim vector does not control how the object might rotate about the aim vector. The orientation of the object about the aim vector is controlled by the up vector and the world up vector.

### Up vector and world up vector

The up vector controls the orientation of the constrained object about the aim vector. Like the aim vector, the up vector is defined relative to the constrained object's local space. By default, the up vector tries to point in the same direction as the world up vector, which is defined relative to the scene's world space. The up vector orients the constrained object about the aim vector by trying to align itself as closely as possible with the world up vector.

When you move the target object(s), the constrained object's aim vector moves to point at the target point, and orients the constrained object accordingly. Simultaneously, the constrained object orients itself about the aim vector as directed by the up vector.

For instance, by default, the up vector is defined so that it points in the same direction as the local rotation positive Y-axis. A constrained object's local positive X-axis will point at the target point, as directed by the default aim vector. Simultaneously, the object's local positive Y-axis will try to point in the same direction as the world up vector, as directed by the object's up vector. The aim vector and up vector work together to constrain the orientation of the constrained object.

By default, the up vector tries to stay as closely aligned with the world up vector as possible. However, you can control the role of the world up vector in a variety of ways. For example, instead of defining the world up vector relative to the workspace's world space (the default), you can define it relative to some other object's local space. Such an object is called a world up object.

### Rolling effects

In certain situations, the constrained object can rapidly rotate about its aim vector. To understand why this happens, you need to understand how aim vectors, up vectors, and world up vectors work. If you are new to constraints, you can skip this section. For more information, see the previous section, "Constrained object's orientation for aim constraints" on page 430.

As the aim vector approaches pointing in the same direction or the opposite direction of the up vector, the constrained object rotates more rapidly about the aim vector. When the aim vector points in exactly the same direction, or in exactly the opposite direction, the constrained object can suddenly rotate by 180 degrees about the aim vector.

These rapid rotations provide rolling effects that you might want to prevent. You can prevent rolling effects by moving or animating the world up vector. For more information, see "Control motion history dependence effects for aim constraints" on page 466.

### Motion history dependence effects

Motion history dependence refers to how an object can provide different motion effects in situations that are identical except that the object has been previously manipulated or animated.

For instance, when you animate an object and run the animation in a loop, if the object ends up moving in slightly different ways at the same frame in each loop, the object is motion history dependent. At a certain frame, the object may be oriented differently depending on its previous orientations. In contrast, if the object moves in exactly the same way during each loop, then the object is motion history independent.

Motion history dependence effects can be a problem if you want predictable motion effects. However, if you are seeking some unpredictable motion effects, you might want to take advantage of an object's motion history dependence.

In certain situations, a constrained object's orientation can become motion history dependent. To understand why this happens, you need to be familiar with aim vectors and up vectors (see "Constrained object's orientation for aim constraints" on page 430).

A constrained object can become motion history dependent if you define the aim vector and the up vector to point in the same direction. For example, you might do this if you define the aim vector relative to the constrained object's local Y-axis, but do not change the default direction of the up vector, which is also relative to the object's local Y-axis. For more information, see "Control motion history dependence effects for aim constraints" on page 466.

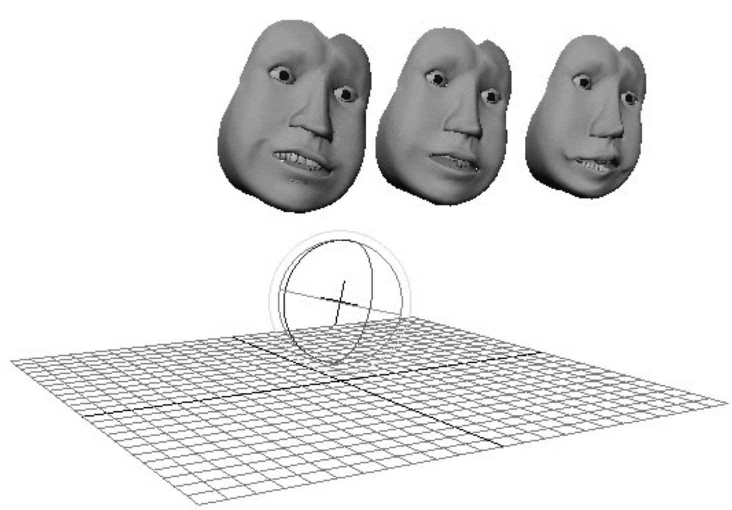
A constrained object can also become motion history dependent if you set the constraint's World Up Type attribute to None. For more information, see "Edit point constraint attributes" on page 461.

### Orient constraints

## Orient constraints

An orient constraint matches the orientation of one object to one or more other objects. This constraint is useful for making several objects orient simultaneously. For example, you can make a group of characters all look in the same direction at the same time by animating one character's head and then constraining all the other character's heads to the head you've just animated.

See "Create orient constraints" on page 449 and "Constrain > Orient" on page 490.



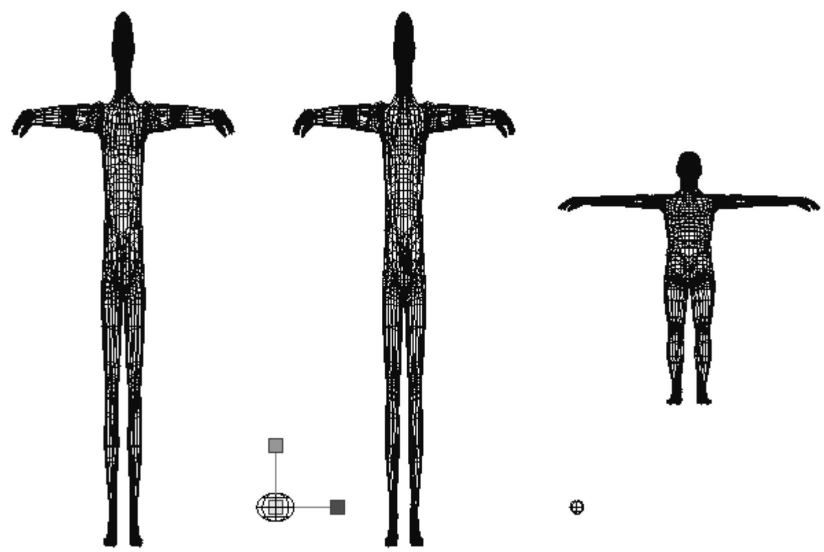
### Scale constraints

## Scale constraints

A scale constraint matches the scaling of one object to one or more other objects. This constraint is useful for making several objects scale simultaneously. For example, you can make a group of characters all look in the same direction at the same time by animating one character's head and then constraining all the other character's heads to the head you've just animated.

A constrained object is an object whose scaling is driven by the scaling of one or more target objects. The scaling of one or more target objects is called the target scale.

See "Create scale constraints" on page 450 and "Constrain > Scale" on page 491.



### Parent constraints

## Parent constraints

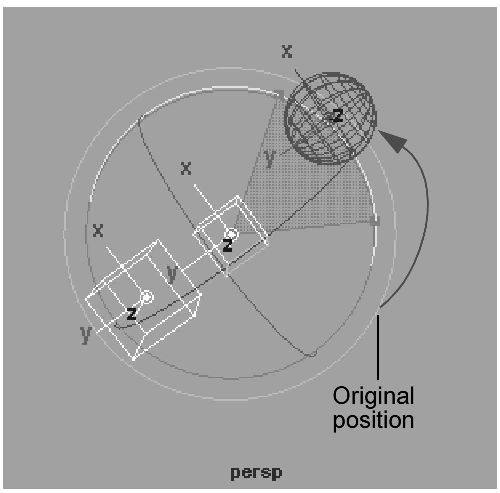
With a parent constraint, you can relate the position—translation and rotation—of one object to another, so that they behave as if part of a parent-child relationship that has multiple target parents. An object's movement can also be constrained by the average position of multiple objects.

When a parent constraint is applied to an object, the *constrained* object does not become part of the *constraining* object's hierarchy or group, but remains independent and behaves as if it is the child of its targets. The constraining object is also known as the *target* object.

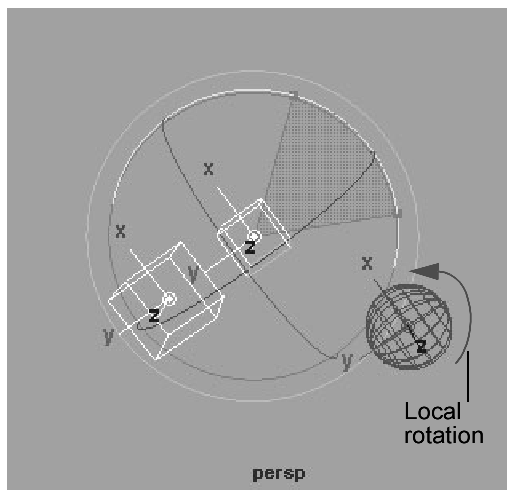
For more information, search the online help for Target objects.

See ”Create a parent constraint” on page 450 and ”Constrain > Parent” on page 491.

An object with a parent constraint does not behave the same as an object with a point and orient constraint. When a Parent constraint is used, rotating the target objects affects the constrained object's rotation along the *world* axis. When a Point and Orient constraint are used, rotating the target objects affects the constrained object's rotation along its *local* axis. This is shown in the following figure.




Parent Constraint



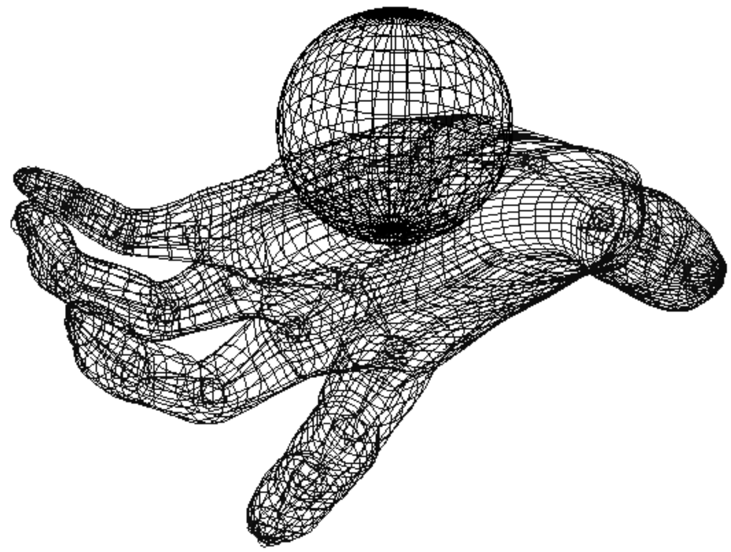

Point and Orient Constraint

**Geometry constraints**

## Geometry constraints

A geometry constraint restricts an object to a NURBS surface, NURBS curve, or polygonal surface (mesh). If you also want the constrained object to orient itself to the surface of the target object(s), use a normal constraint. For more information on normal constraints, see ”Normal constraints” on page 436.

See ”Create a geometry constraint” on page 454 and ”Constrain > Geometry” on page 492.



## Target point for Geometry constraints

The target point is the position of the target object's nearest surface location. If there is more than one target object, the target point is the average position of the nearest surface locations of all the target objects. If you are using more than one target object, you can vary the influence of each target object on the calculation of the target point. The target point can be a weighted average of the nearest surface locations of the target objects, with some target objects having more influence than others. The influence of target objects on the weighted average is specified by target object weights.

## Geometry constraint motion history dependence

Motion history dependence refers to how an object can provide different motion effects in situations that are identical except that the object has been previously manipulated or animated.

For instance, when you animate an object and run the animation in a loop, if the object ends up moving in slightly different ways at the same frame in each loop, the object is motion history dependent. At a certain frame, the object may be oriented differently depending on its previous orientations. In contrast, if the object moves in exactly the same way during each loop, then the object is motion history independent.

Objects constrained by geometry constraints are motion history dependent. That means that the end result of a constrained object's animation depends on where the object started.

## Using a point constraint with a geometry constraint

You can create a point constraint for an object that is already constrained by a geometry constraint. The constrained object's position will be the point on the target object's surface that is closest to the point constraint's target point.

For more information on point constraints, see "Point constraints" on page 428.

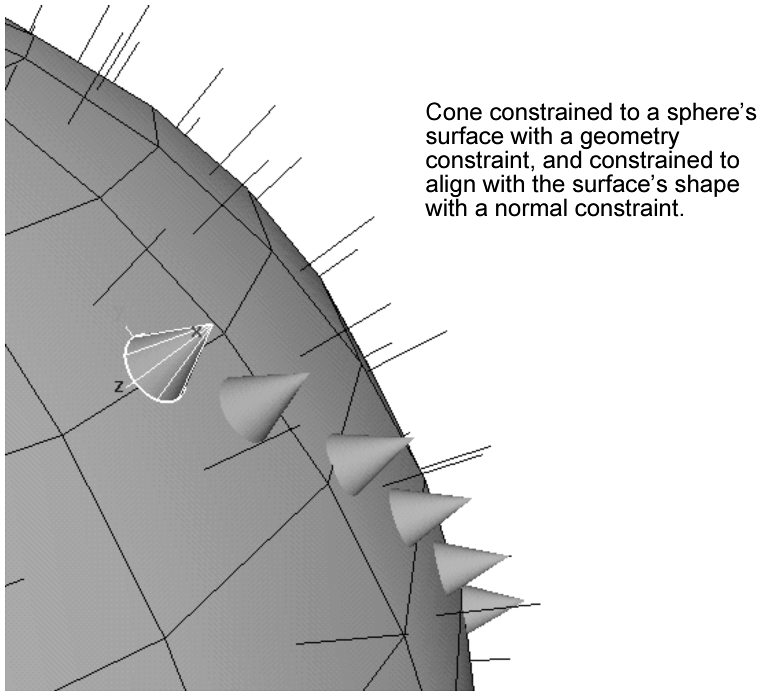
### Normal constraints

## Normal constraints

A normal constraint constrains an object's orientation so that it aligns with the normal vectors of a NURBS surface or polygonal surface (mesh). Normal constraints are useful for having an object travel across a surface that has a unique, complex shape. Without normal constraints, moving or animating the object across the surface could be tedious and time-consuming. For example, you might want to have a tear falling down along character's face. Instead of animating the tear directly, you could constrain it to the face's surface.

See "Create normal constraints" on page 455 and "Constrain > Normal" on page 492.

Typically, you use normal constraints with geometry constraints. For more information on geometry constraints, see "Geometry constraints" on page 434.




## Constrained object's orientation for normal constraints

The constrained object's orientation is controlled by three vectors: the aim vector, the up vector, and the world up vector. These vectors are not displayed in the workspace, but you can infer their effect on the constrained object's orientation.

You do not need to understand the details of how these vectors work in order to use constraints effectively. If you are new to constraints, you can skip the rest of this section. However, if you want to exercise a high degree of control over a normal constraint, you'll need to work with these vectors. Also, familiarity with these vectors can help you to understand how a constrained object can suddenly roll.

### Aim vector

The aim vector constrains the constrained object so that it always aligns with the target vector. The aim vector starts at the constrained object's pivot point and then aligns with the target vector.

How the object rotates to align with the target vector depends on how the aim vector is defined relative to the object's local space. For instance, by default, the aim vector is defined so that it points in the same direction as the local rotation positive X-axis. Consequently, by default, a constrained object's local rotation positive X-axis will align with the target vector.

By itself, the aim vector does not completely constrain the object, because the aim vector does not control how the object might rotate about the aim vector. The orientation of the object about the aim vector is controlled by the up vector and the world up vector.

### Up vector and world up vector

The up vector controls the orientation of the constrained object about the aim vector. Like the aim vector, the up vector is defined relative to the constrained object's local space. By default, the up vector tries to point in the same direction as the world up vector, which is defined relative to the scene's world space. The up vector orients the constrained object about the aim vector by trying to align itself as closely as possible with the world up vector.

When you move the target object(s), the constrained object's aim vector moves to align with the target vector, and orients the constrained object accordingly. Simultaneously, the constrained object orients itself about the aim vector as directed by the up vector.

For instance, by default, the up vector is defined so that it points in the same direction as the local rotation positive Y-axis. A constrained object's local positive X-axis will align with the target vector as directed by the default aim vector. Simultaneously, the object's local positive Y-axis will try to point in the same direction as the world up vector, as directed by the object's up vector. The aim vector and up vector work together to constrain the orientation of the constrained object.

By default, the up vector tries to stay as closely aligned with the world up vector as possible. However, you can control the role of the world up vector in a variety of ways. For example, instead of defining the world up vector relative to the workspace's world space (the default), you can define it relative to some other object's local space. Such an object is called a world up object.

## Rolling effects

In certain situations, the constrained object can rapidly rotate about its aim vector. To understand why this happens, you need to understand how aim vectors, up vectors, and world up vectors work. If you are new to constraints, you can skip this section. For more information, see the previous section, "Constrained object's orientation for aim constraints" on page 430.

As the aim vector approaches pointing in the same direction or the opposite direction of the up vector, the constrained object rotates more rapidly about the aim vector. When the aim vector points in exactly the same direction, or in exactly the opposite direction, the constrained object can suddenly rotate by 180 degrees about the aim vector.

These rapid rotations provide rolling effects that you might want to prevent. You can prevent rolling effects by moving or animating the world up vector. For more information, see "Control motion history dependence effects for aim constraints" on page 466.

## Motion history dependence effects

Motion history dependence refers to how an object can provide different motion effects in situations that are identical except that the object has been previously manipulated or animated.

For instance, when you animate an object and run the animation in a loop, if the object ends up moving in slightly different ways at the same frame in each loop, the object is motion history dependent. At a certain frame, the object may be oriented differently depending on its previous orientations. In contrast, if the object moves in exactly the same way during each loop, then the object is motion history independent.

Motion history dependence effects can be a problem if you want predictable motion effects. However, if you are seeking some unpredictable motion effects, you might want to take advantage of an object's motion history dependence.

In certain situations, a constrained object's orientation can become motion history dependent. To understand why this happens, you need to be familiar with aim vectors and up vectors (see "Constrained object's orientation for aim constraints" on page 430).

A constrained object can become motion history dependent if you define the aim vector and the up vector to point in the same direction. For example, you might do this if you define the aim vector relative to the constrained object's local Y-axis, but do not change the default direction of the up vector, which is also relative to the object's local Y-axis. For more information, see "Control motion history dependence effects for aim constraints" on page 466.

A constrained object can also become motion history dependent if you set the constraint's World Up Type attribute to None. For more information, see "Edit point constraint attributes" on page 461.

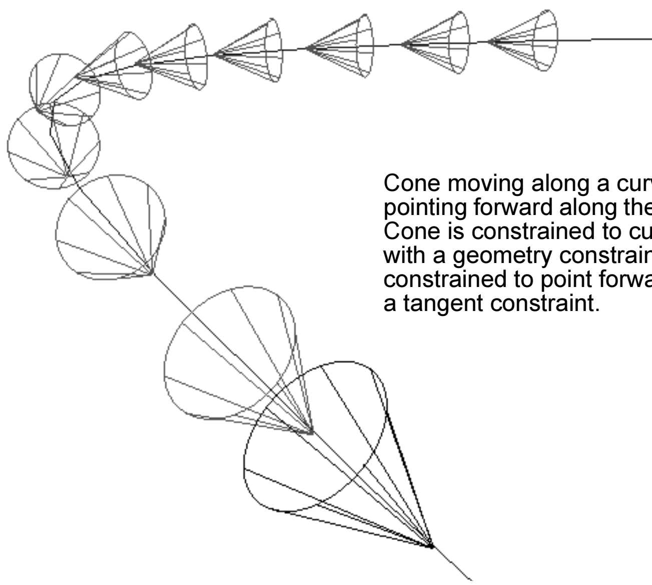
### Tangent constraints

## Tangent constraints

Tangent constraints constrain an object's orientation so that as an object moves along a curve, the object always points in the direction a curve. The curve provides the path of the object's motion, and the object orients itself to point along the curve. Tangent constraints are useful for having an object follow a curve's direction, such as a roller coaster car following the tracks.

See "Create tangent constraints" on page 456 and "Constrain > Tangent" on page 493.

Typically, you use tangent constraints with geometry constraints. For more information on geometry constraints, see "Geometry constraints" on page 434.



Cone moving along a curve and pointing forward along the curve. Cone is constrained to curve with a geometry constraint, and constrained to point forward with a tangent constraint.

## Target vector for tangent curves

The target vector, or weighted average vector, represents the tangent vector along the curve at the position of the constrained object. Maya calculates the target vector as a weighted average of the curve's nearby tangents (that is, the curve's tangent vectors).

## Constrained object's orientation for tangent curves

The constrained object's orientation is controlled by three vectors: the aim vector, the up vector, and the world up vector. These vectors are not displayed in the workspace, but you can infer their effect on the constrained object's orientation.

You do not need to understand the details of how these vectors work in order to use constraints effectively. If you are new to constraints, you can skip the rest of this section. However, if you want to exercise a high degree of control over a tangent constraint, you'll need to work with these vectors. Also, familiarity with these vectors can help you to understand how a constrained object can suddenly roll.

### Aim vector

The aim vector constrains the constrained object so that it always aligns with the target vector. The aim vector starts at the constrained object's pivot point and then aligns with the target vector.

How the object rotates to align with the target vector depends on how the aim vector is defined relative to the object's local space. For instance, by default, the aim vector is defined so that it points in the same direction as the local rotation positive X-axis. Consequently, by default, a constrained object's local rotation positive X-axis will align with the target vector.

By itself, the aim vector does not completely constrain the object, because the aim vector does not control how the object might rotate about the aim vector. The orientation of the object about the aim vector is controlled by the up vector and the world up vector.

### Up vector and world up vector

The up vector controls the orientation of the constrained object about the aim vector. Like the aim vector, the up vector is defined relative to the constrained object's local space. By default, the up vector tries to point in the same direction as the world up vector, which is defined relative to the scene's world space. The up vector orients the constrained object about the aim vector by trying to align itself as closely as possible with the world up vector.

When you move the target object(s), the constrained object's aim vector moves to align with the target vector, and orients the constrained object accordingly. Simultaneously, the constrained object orients itself about the aim vector as directed by the up vector.

For instance, by default, the up vector is defined so that it points in the same direction as the local rotation positive Y-axis. A constrained object's local positive X-axis will align with the target vector, as directed by the default aim vector. Simultaneously, the object's local positive Y-axis will try to point in the same direction as the world up vector, as directed by the object's up vector. The aim vector and up vector work together to constrain the orientation of the constrained object.

By default, the up vector tries to stay as closely aligned with the world up vector as possible. However, you can control the role of the world up vector in a variety of ways. For example, instead of defining the world up vector relative to the workspace's world space (the default), you can define it relative to some other object's local space. Such an object is called a world up object.

### Rolling effects

In certain situations, the constrained object can rapidly rotate about its aim vector. To understand why this happens, you need to understand how aim vectors, up vectors, and world up vectors work. If you are new to constraints, you can skip this section. For more information, see the previous section, ”Constrained object’s orientation for aim constraints” on page 430.

As the aim vector approaches pointing in the same direction or the opposite direction of the up vector, the constrained object rotates more rapidly about the aim vector. When the aim vector points in exactly the same direction, or in exactly the opposite direction, the constrained object can suddenly rotate by 180 degrees about the aim vector.

These rapid rotations provide rolling effects that you might want to prevent. You can prevent rolling effects by moving or animating the world up vector. For more information, see ”Control motion history dependence effects for aim constraints” on page 466.

## Motion history dependence effects

Motion history dependence refers to how an object can provide different motion effects in situations that are identical except that the object has been previously manipulated or animated.

For instance, when you animate an object and run the animation in a loop, if the object ends up moving in slightly different ways at the same frame in each loop, the object is motion history dependent. At a certain frame, the object may be oriented differently depending on its previous orientations. In contrast, if the object moves in exactly the same way during each loop, then the object is motion history independent.

Motion history dependence effects can be a problem if you want predictable motion effects. However, if you are seeking some unpredictable motion effects, you might want to take advantage of an object’s motion history dependence.

In certain situations, a constrained object’s orientation can become motion history dependent. To understand why this happens, you need to be familiar with aim vectors and up vectors (see ”Constrained object’s orientation for aim constraints” on page 430).

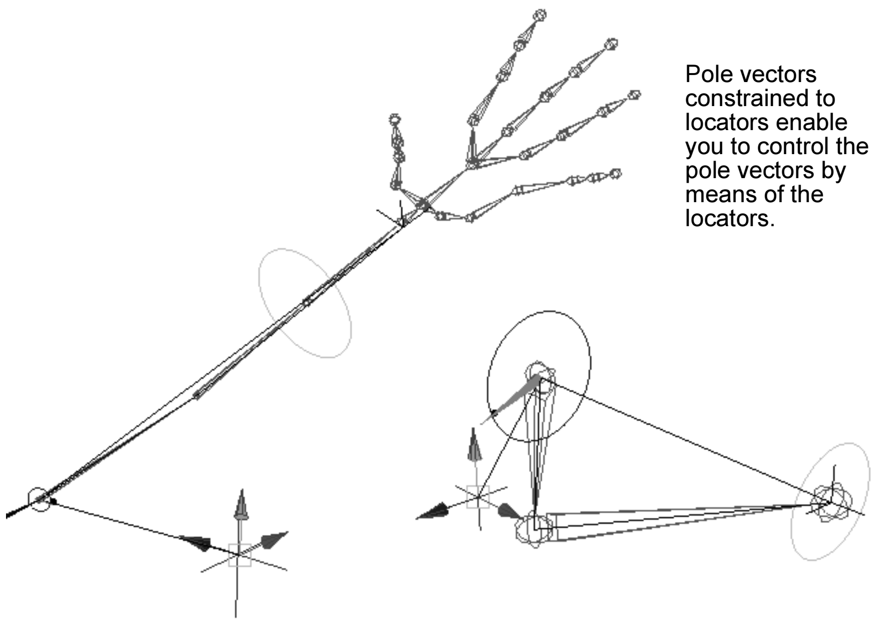
A constrained object can become motion history dependent if you define the aim vector and the up vector to point in the same direction. For example, you might do this if you define the aim vector relative to the constrained object’s local Y-axis, but do not change the default direction of the up vector, which is also relative to the object’s local Y-axis. For more information, see ”Control motion history dependence effects for aim constraints” on page 466.

A constrained object can also become motion history dependent if you set the constraint's World Up Type attribute to None. For more information, see "Edit point constraint attributes" on page 461.

### Pole Vector constraints

## Pole Vector constraints

A pole vector constraint causes the end of a pole vector to move to and follow the position of an object, or the average position of several objects.




In character setup, the pole vectors of IK rotate plane handles for arm joint chains are often constrained to locators placed behind the character.

In general, you will want to constrain a pole vector so that the joint chain does not unexpectedly flip when you manipulate the IK rotate plane handle. Because flipping can occur when the handle vector approaches or intersects the pole vector, you should constrain the pole vector so that the handle vector is unlikely to cross it.

### Target objects for pole vector constraints

A constrained pole vector is a pole vector whose position is driven by the position of one or more target objects. The position of one or more target objects is called the target point.

### Constrained pole vector's end position

The constrained pole vector's end position is driven by the target point. However, you can offset the constrained pole vector's end position from the target point. Offsetting the constrained pole vector's end position from the target point can be useful in situations where you don't want the local axis of the constrained pole vector to coincide exactly with the target point.

Keep in mind that the IK chain controlled by the pole vector's IK rotate plane handle rotates when the target point moves. If you move the target object, the movement of the pole vector will cause the IK chain to rotate.

**Animation-Constraint blending**

## Animation-Constraint blending

You can blend keyframe animation and constraints on the same object. See "Animate and constrain an object" on page 458 and "Animation-Constraint blending workflow" on page 459.

When both keyframe animation and constraints are applied to the same object, a pairBlend node is automatically generated. See "pairBlend" on page 508. All the object's animation and constraint channels are linked to the object through this pairBlend node.

You can modify the weight of the animation-constraint blend to generate various effects. This Weight attribute is on the pairBlend node. See "Blend Weights in the Channel Box" on page 511.

**Note**

- Since animation-constraint blending lets you transform the constrained channels of a driven object, you must lock the constrained channels that you want strictly influenced by the target objects.
- After transforming a constrained object, you can retain its new translation, rotation, and/or scaling by reapplying Maintain Offset from the Modify Constrained Axis Options window. This prevents the constrained object from snapping back to its original translation, rotation, and/or scaling when its target objects are transformed.

### Animation-Constraint blending examples

- A man runs to a parked convertible car, jumps over the door and into the driver's seat, and then drives away. In this example, both the character and car models are animated, and various parts of the man are point constrained to the car's door, seat, and steering wheel.
- A basketball player dribbles the ball, picks it up, and throws it to another player. The player that catches the ball pauses and then shoots for the hoop. In this example, the ball and all the players hands are animated, and the ball is parent constrained to the various player's hands.

## How do I? Edit least squares modifier attributes

The least squares modifier node controls how closely curve points stick to the locators. If you are using many point on curve locator constraints on a curve, or the shape of the curve is particularly complex, the curve might not always be able to stick to all of the point on curve locator constraints at the same time.

**To edit least squares modifier attributes with Attribute Editor**

1. Select a locator that acting as a point on curve locator constraint.
2. Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a). From the Attribute Editor, go to the least squares modifier node (default name: leastSquaresModifier*n*), which is an output of the locator. (In the Channel Box, the node is listed under the locator's OUTPUTS.)

3 The following sections make available attributes: Least Squares Modifier Attributes, Point Constraints, Node Behavior, and Extra Attributes.

# Set constraints

## Set point constraints

## Set point constraint options

#### To set point constraint options

1 If you also want to create a point constraint now, select one or more objects. The last object selected will be the constrained object.

2 Select Constrain > Point > ❒.

The Point Constraint Options window is displayed.

3 Set the constraint options

- Click Add to create a point constraint.
- Click Save to save the constraint options.
- Click Reset to reset to the default constraint options.
- Click Close to close the Point Constraint Options window.

## Create point constraints

When creating point constraints, you can first set creation options and then create a point constraint, or you can immediately create a constraint with the current creation options.

#### To create a point constraint

1 Select one or more target objects, followed by the object you want to constrain to them.

2 Select Constrain > Point.

A point constraint is created with the current constraint options.

The constrained object's position attributes (Translate X, Y, and Z) are now locked. Their values are now provided by the target point.

# Add target objects for point constraints

After you've created a point constraint, you can add more target objects for additional control over the constrained object's position. Adding more target objects changes the target point, which changes the constrained object's position. Adding more target objects is similar to creating point constraints.

### To add target objects

1. Select one or more objects you want to add as target objects, followed by the constrained object.
2. Select Constrain > Point > ❐.

   The Point Constraint Options window is displayed.
3. Click Add to add the selected objects as target objects.

   The constrained object's position changes (if Maintain Offset is off), indicating that it is now constrained by the objects you've just added as target objects.

# Animate target object weights

An interesting technique you can use with point constraints is to animate the target object weights specified by the *targetObject* W*n* channels. You can vary the weights from 0 to any value, so that as an animation progresses different target objects can take turns driving a constrained object's motion.

# Offset constrained object's position

The constrained object's position is driven by the target point, but you can offset the constrained object's position from the target point. To do so, edit the Offset and Offset Polarity attributes with the Attribute Editor.

### To offset constrained object's position

Edit the Offset and Offset Polarity attributes as described in "Edit point constraint attributes" on page 461.

By default, these attributes are not displayed as channels in the Channel Box. If you'd like to control them from the Channel Box, you can add them by using the Channel Control editor (select Window > General Editors > Channel Control.).

## Set aim constraints

# Set aim constraint options

**To set constraint options**

1 If you want to create an aim constraint now, select one or more objects. The last object selected will be the constrained object.

2 Select Constrain > Aim > ❐.

The Aim Constraint Options window appears.

3 Set the constraint options.

- Click Add to create an aim constraint.
- Click Save to save the constraint options.
- Click Reset to reset to the default constraint options.
- Click Close to close the Aim Constraint Options window.

# Create aim constraints

When creating aim constraints, you can first set creation options and then create an aim constraint, or you can immediately create a constraint with the current creation options.

The default constraint options work well for constraining objects so that they aim along their local rotation positive X-axis.

**To create an aim constraint**

1 Select one or more target objects, followed by the object you want to constrain to them.

2 Select Constrain > Aim.

An aim constraint is created with the current constraint options.

The constrained object’s orientation attributes (Rotate X, Y, and Z) are now locked. Their values are now provided by how the constraint object’s aim vector points at the target point.

## Set orient constraint

# Set orient constraint options

**To set orient constraint options**

1 If you also want to create an orient constraint now, select one or more objects. The last object selected will be the constrained object.

2 Select Constrain > Orient > ❐.

The Orient Constraint Options window appears.

3 Set the constraint options.

4 Do the following:

- Click Add to create an orient constraint.
- Click Save to save the constraint options.
- Click Reset to reset to the default constraint options.
- Click Close to close the Orient Constraint Options window.

## Create orient constraints

When creating orient constraints, you can first set creation options and then create an orient constraint, or you can immediately create a constraint with the current creation options.

#### To create an orient constraint

1 Select one or more target objects, followed by the object you want to constrain to them.

2 Select Constrain > Orient.

An orient constraint is created with the current constraint options.

The constrained object's orientation attributes (Rotate X, Y, and Z) are now locked. Their values are now provided by the target orientation.

## Add orient constraint target objects

After you've created an orient constraint, you can add more target objects for additional control over the constrained object's position. Adding more target objects is similar to creating orient constraints.

#### To add target objects

1 Select one or more objects you want to add as target objects, followed by the constrained object.

2 Select Constrain > Orient > ❐.

The Orient Constraint Options window is displayed.

3 Click Add to add the selected objects as target objects.

The constrained object's position changes (if Maintain Offset is off), indicating that it is now constrained by the objects you've just added as target objects.

## Set scale constraints

# Set scale constraint options

### To set scale constraint options

1. If you also want to create a scale constraint now, select one or more objects. The last object selected will be the constrained object.
2. Select Constrain > Scale > ❐.

   The Scale Constraint Options window is displayed.
3. Set the constraint options.
4. Do one of the following:
   - Click Add to create a scale constraint.
   - Click Save to save the constraint options.
   - Click Reset to reset to the default constraint options.
   - Click Close to close the Scale Constraint Options window.

# Create scale constraints

When creating scale constraints, you can first set creation options and then create a scale constraint, or you can immediately create a constraint with the current creation options.

### To create a scale constraint

1. Select one or more target objects, followed by the object you want to constrain to them.
2. Select Constrain > Scale.

   A scale constraint is created with the current constraint options.

   The constrained object's scale attributes (Scale X, Y, and Z) are now locked. Their values are now provided by the target's scaling.

## Set parent constraints

# Create a parent constraint

Before creating your first parent constraint in a Maya session, we recommend that you set the Parent constraint options. See the "Constrain > Parent" on page 491.

After you've created a parent constraint, you can add more target objects for additional control over the constrained object's position and rotation. Adding more target objects changes the target point, which in turn changes the constrained object's position.

For more information, search the online help for Target Point.

### To create a parent constraint

1. Select one or more target objects, followed by the object you want to constrain.
2. Select Constrain > Parent > ❐.

   The Parent Constraint Options window appears.
3. Set the constraint options or select Edit > Reset Settings.

   For information on these options, see "Constrain > Parent" on page 491.
4. To finish setting the constraint, do one of the following:
   - Click Add to create a parent constraint. The Parent Constraint Options window closes. Select Edit > Save Settings to save the constraint options.
   - Click Apply to create a parent constraint. The Parent Constraint Options window remains open. By clicking apply, you can also apply the current constraint options settings to other objects. Select Edit > Save Settings to save the constraint options.
   - Click Close to close the Parent Constraint Options window without setting or creating the constraint.

   A parent constraint is created with the current constraint options.

   The constrained object's position attributes (Translate X, Y, and Z) and rotation attributes (Rotate X, Y, and Z) are now influenced by the target point.

For more information, search the online help for Target Point.

### To add a target object to a parent constraint

1. Select one or more objects to add as target objects, followed by the constrained object.
2. Select Constrain > Parent.

   The constrained object's position and rotation change, indicating that it is now constrained by the objects you have added as target objects.

| Note | If you turn on Maintain Offset in the Parent Constraint Options, the constrained object's position and rotation will not change when you add a target to the constraint. |
|---|---|

**Use parent constraints**

# Parent constraint workflow

## Scenario

A character walks, looks around, stops, and then removes his hat from his head and places it on a table nearby.

1. Select the character's head, and then the hat.
2. Parent constrain the hat to the head, making sure that Maintain Offset is on and the weight is set to 1 in the constraint options.

   For the procedure on creating parent constraints see "To create a parent constraint" on page 451.

   Maintain Offset preserves the original, relative translation and rotation of the constrained object.

   The Weight attribute sets the amount of influence that the target's position and orientation has on the constrained object. For information on parent constraint weighting see "Weight" on page 492.

   A parent constraint is created. Now when the character moves, the hat respects the position and orientation of the character's head in the scene.
3. Select the hat, and then select the head's weight in the Channel Box.

| Note | When the hat is selected, the parent constraint and all its target weights appear in the Channel Box. |
|---|---|

4. Right-click the head's weight and select Key Selected from the menu that appears.

   This anchors the parent constraint's weighting, and the state of the hat, at the current time in the character's animation.
5. Animate the character so that it appears to walk, look around, and then stop.

   The parent constraint forces the hat to follow the head's movements as it nods up and down and rotates from side to side.

6 Animate the character’s arm and hand so that it reaches up and takes hold of the hat’s brim.

7 Select the hand, and then the hat.

8 Parent constrain the hat to the hand, making sure that Maintain Offset is on and the weight is set to 0 in the constraint options.

9 Key the weight of the head and the hand. See steps 3 and 4.

10 Advance one frame along the timeline.

11 Set the weight of the head to 0 and the weight of the hand to 1.

The hand now fully controls the translation and rotation of the hat.

12 Key the head and hand weights. See steps 3 and 4.

This anchors the weighting of the head and hand at the current time in the character’s animation.

13 Transform the arm and hand to the position where it appears to be placing the hat on the table.

14 Set the hat’s current position as its rest position.

Setting the hat’s rest position prevents it from popping back to its original position when the hand’s weight is set back to 0. For more information on the rest position, see ”Constrain > Set Rest Position” on page 494.

15 Set the weight of the hand to 0 and key the weight.

16 Animate the hands so that they return to their original positions hanging at the characters sides.

Since the hands no longer have any influence over the hat’s position or rotation, the hat remains on the table as the hands move back to the character’s sides.

Note If you want to ensure that their is no interpolation between weights (weights switch between 1 and 0 with no blend), set the tangent types of the weights to Stepped.

## Set geometry constraints

When creating geometry constraints, you can first set creation options and then create a geometry constraint, or you can immediately create a constraint with the current creation options.

# Set geometry constraint options

**To set geometry constraint options**

1. If you also want to create a geometry constraint now, select one or more objects. The last object selected will be the constrained object.
2. Select Constrain > Geometry > ❐.

   The Geometry Constraint Options window is displayed.
3. Set the constraint options.
4. Do one of the following:
   - Click Add to create a geometry constraint.
   - Click Save to save the constraint options.
   - Click Reset to reset to the default constraint options.
   - Click Close to close the Geometry Constraint Options window.

# Create a geometry constraint

**To create a geometry constraint**

1. Select one or more target objects, followed by the object you want to constrain to them.
2. Select Constrain > Geometry.

   A geometry constraint is created with the current constraint options.

# Animate target object weights for geometry constraints

An interesting technique you can use with geometry constraints is to animate the target object weights specified by the *targetObject* W*n* channels. You can vary the weights from 0 to any value, so that as an animation progresses different target objects can take turns driving a constrained object's position.

If all the target objects have the same weight (the default), the target point is taken from the first target object you selected when you created the constraint. For instance, if all the weights are 1, the target object whose weight is specified by the constraint's *targetObject* W0 channel provides the target point.

You should animate the target weights so that only one target has the highest weight at any given frame.

Note that objects constrained by geometry constraints are motion history dependent.

# Animate the constrained object for geometry constraints

In contrast with the point constraint, the geometry constraint allows you to set keys on the constrained object's position (Translate X, Y, and Z channels). The constrained object's position will be the point on the target object's surface that is closest to the keyed position.

### Set normal constraints

## Set normal constraint options

#### To set normal constraint options

1. If you want to create a normal constraint now, select one or more objects. The last object selected will be the constrained object.
2. Select Constrain > Normal > ❐.

   The Normal Constraint Options window is displayed.
3. Set the constraint options.
4. Do one of the following:
   - Click Add to create a normal constraint.
   - Click Save to save the constraint options.
   - Click Reset to reset to the default constraint options.
   - Click Close to close the Normal Constraint Options window.

## Create normal constraints

When creating normal constraints, you can first set creation options and then create a normal constraint, or you can immediately create a constraint with the current creation options.

Typically, you would want to first create a geometry constraint to constrain the object to some surface, and then create a normal constraint to constrain the object's orientation so that it aligns with the normal vectors of the surface.

The default constraint options work well for constraining objects so that they aim along their local rotation positive X-axis.

#### To create a normal constraint

1. Select one or more target objects, followed by the object you want to constrain to them.

   Note that typically you would want to first create a geometry constraint, and then create a normal constraint.

**2** Select Constrain > Normal.

A normal constraint is created with the current constraint options.

The constrained object's orientation attributes (Rotate X, Y, and Z) are now locked. Their values are now provided by how the constraint object's aim vector aligns with the target vector.

### Set tangent constraints

## Set tangent constraint options

#### To set tangent constraint options

**1** If you want to create a tangent constraint now, select one or more objects. The last object selected will be the constrained object.

**2** Select Constrain > Tangent > ❐.

The Tangent Constraint Options window is displayed.

**3** Set the constraint options.

- Click Add to create tangent constraint.
- Click Save to save the constraint options.
- Click Reset to reset to the default constraint options.
- Click Close to close the Tangent Constraint Options window.

## Create tangent constraints

When creating tangent constraints, you can first set creation options and then create a tangent constraint, or you can immediately create a constraint with the current creation options.

Typically, you would want to first create a geometry constraint to constrain the object to some curve, and then create a tangent constraint to constrain the object's orientation so that it aligns with the tangents of the curve.

The default constraint options work well for constraining objects so that they aim along their local rotation positive X-axis.

#### To create a tangent constraint

**1** Select one or more target objects, followed by the object you want to constrain to them.

Note that typically you would want to first create a geometry constraint, and then create a tangent constraint.

**2** Select Constrain > Tangent.

A tangent constraint is created with the current constraint options.

The constrained object's orientation attributes (Rotate X, Y, and Z) are now locked. Their values are now provided by how the constraint object's aim vector aligns with the target vector.

### Set pole vector constraints

## Set pole vector constraint options

#### To set constraint options

1. If you also want to create a pole vector constraint now, select one or more objects, followed by the IK rotate plane handle whose pole vector you want to constrain.
2. Select Constrain > Pole Vector > ❐.

   The Pole Vector Constraint Options window is displayed.
3. Set the constraint options.
4. See the following:
   - Click Add to create a pole vector constraint.
   - Click Save to save the constraint options.
   - Click Reset to reset to the default constraint options.
   - Click Close to close the Pole Vector Constraint Options window.

## Create a pole vector constraint

When creating pole vector constraints, you can first set creation options and then create a pole vector constraint, or you can immediately create a constraint with the current creation options.

#### To create a pole vector constraint

1. Select one or more target objects, followed by the IK rotate plane handle whose pole vector you want to constrain to them.
2. Select Constrain > Pole Vector.

   A pole vector constraint is created with the current constraint options.

   The pole vector's position attributes (Pole Vector X, Y, and Z) are now locked. Their values are now provided by the target point.

## Set the rest position

# Set the rest position

### To define a constrained object's default position

1. Select the constrained object for which you want to define the rest position.
2. Set the constrained object's target weights to 0.
3. Move the constrained object to a new position or leave it at its current position.
4. Select Constrain > Set Rest Position.

   The current position of the constrained object is now its default location in world space when its target weights are 0.

## Apply animation and constraints to the same object

# Animate and constrain an object

**Tips**

- Since animation-constraint blending lets you transform the constrained channels of a driven object, you must lock the constrained channels that you want strictly influenced by the target objects.
- After transforming a constrained object, you can retain its new translation, rotation, and/or scaling by reapplying Maintain Offset from the Modify Constrained Axis Options window. This prevents the constrained object from snapping back to its original translation, rotation, and/or scaling when its target objects are transformed.

### To blend animation and constraints on the same object

1. Animate and constrain the same object.

   When an object is driven by both animation and constraints, a pairBlend node is automatically created as an intermediary between the animation channels, constraint channels, and their target object. See "pairBlend" on page 508.
2. Edit and key the pairBlend *Weight* attribute to animate the amount of influence the constraint and keyframe animation have on their target object. See "Blend Weights in the Channel Box" on page 511.

# Animation-Constraint blending workflow

See also "Animate and constrain an object" on page 458.

## Scenario

A basketball player character passes a ball to his teammate.

1. Select the first player's hands, and then the basketball.
2. Parent constrain the ball to the hands, making sure that Maintain Offset is on and the weight is set to 1 in the constraint options.

   For the procedure on creating parent constraints, see "To create a parent constraint" on page 451.

   Maintain Offset preserves the original, relative translation and rotation of the constrained object.

   The Weight attribute sets the amount of influence that the target's position and orientation has on the constrained object. For information on parent constraint weighting see "Weight" on page 492.
3. In the Channel Box, set keys for the weights of the first player's hands.

| Note | When the basketball is selected, the parent constraint and all its target weights appear in the Channel Box. |
|---|---|

   This anchors the parent constraint's weighting, and the position and orientation of the basketball, at the current time in the character's animation.
4. Animate the arms of the first player so that they appear to be throwing the ball.
5. At the point in the first character's animation where he is to let go of the ball, set a key on the ball and on the weights of the hands.

   Since the basketball now has both a constraint and keyframe animation, a new *blend* weight attribute appears in the Channel Box.
6. Change the blend weight to 1, and key the blend weight.
7. Advance one frame along the timeline.
8. Set the weight of the first player's hands to 0, and set the blend weight to 0.

   When the blend weight is 0, the keyframe animation has full control over the position and orientation of the ball.
9. Key the hands' weights and the blend weight.

   This anchors the animation-keyframe blend weight and the first player's weight at the current time in the character's animation.

**10** Keyframe the ball as it is *thrown* from the first player to the second player.

**11** When the ball reaches the position where the second player catches it, parent constrain the ball to the hands of the second player, making sure that Maintain Offset is on and the weight is 0 in the constraint options.

**12** Key the weights of the second player's hands and that of the blend weight.

**13** Advance one frame along the timeline.

**14** Set the blend weight to 1 and the second player's hands' weights to 1.

When the blend weight is 1, the parent constraint has full control over the position and orientation of the ball.

**15** Key the blend weight and the weights of the second player's hands again.

This anchors the weighting of the ball's animation and that of the parent constraint at the current time in the characters' animation.

**16** Animate the second player.

| Note | If you want to ensure that the transitions between the blend weights are clean (no interpolation), set the tangent types of the weights to Stepped. |
|---|---|

# Edit constraints

## Turn all constraint nodes on or off

You can quickly disable (or again enable) all the constraint nodes in a scene. Disabling all constraint nodes sets the Node State attribute of all constraint nodes to HasNoEffect. Enabling all constraint nodes sets the Node State attribute to Normal. For more information on the Node State attribute, see "Constraint node behavior" on page 426.

### To disable all constraint nodes

- Turn off Modify > Enable Nodes > Constraints.

### To enable all constraint nodes

- Turn on Modify > Enable Nodes > Constraints.

## Edit point constraints

# Edit point constraint node behavior

### To set node behavior with Attribute Editor

1. Open the node's Attribute Editor.
2. In the Attribute Editor, open Node Behavior.
3. Click Caching on or off.
4. Select the Node State as Normal, HasNoEffect, or Blocking. (The Waiting-Normal, Waiting-HasNoEffect, and Waiting-Blocking states are for Maya's internal use.)
5. Close the Attribute Editor.

### To set Node State channel with Channel Box

When editing constraint channels with the Channel Box, you can set the Node State to Normal, HasNoEffect, Blocking, Waiting-Normal, Waiting-HasNoEffect, or Waiting-Blocking.

# Edit point constraint attributes

### To edit attributes with Attribute Editor

1. Select the point constraint node.
2. Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).

   The following sections make available attributes: Transform Attributes, Point Constraint Attributes, Pivots, Limit Information, Display, Node Behavior, and Extra Attributes.

# Edit point constraint channels

### To edit channels with the Channel Box

1. In the scene, select the constrained object.

   The point constraint node is in the constrained object's history, listed and automatically selected in the Channel Box under SHAPES.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Set the channels.
3. Click on a channel name with the left mouse button.

4 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

5 Do one of the following:

- Click Select to select the node you are now editing as the currently selected object in your scene.
- Click Load Attributes to load the attribute values of the currently selected node.
- Click Close to close the Attribute Editor.

## Remove target objects for point constraints

After you've created a point constraint, you can remove any of the target objects so that the objects no longer constrain the constrained object.

Note that when you remove a target object, you also remove any animation curves attached to the constraint object for that target object.

For this procedure see "To modify which axes of a constrained object are affected by the constraint" on page 486.

## Change target object weights for point constraints

A target object's weight specifies how much the position of the constrained object can be influenced by a target object. The weights are attributes of the point constraint. For each target object, an attribute named *targetObject* W*n* is included that specifies the weight of each target object. By default, the weights are set to 1, which gives each target object an equal influence over the constrained object's position. However, you can change the weights so that some target objects can have more (or less) influence than others. You can change target object weights with the Channel Box or the Attribute Editor.

### To change target object weights with Channel Box

Edit the *targetObject* W*n* channels as described in "Edit point constraint channels" on page 461.

### To change target object weights with Attribute Editor

Edit the *targetObject* W*n* attributes as described in "Edit point constraint attributes" on page 461.

## Delete point constraints

To delete a point constraint, delete the point constraint node.

#### To delete a point constraint

1. Select the point constraint node only. You can select the point constraint's selection handle if displayed, or use the Hypergraph to select the point constraint node.
2. Select Edit > Delete (default shortcut: Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key).

### Edit aim constraints

## Edit aim constraint channels

#### To edit channels with the Channel Box

1. In the scene, select the constrained object.

   The aim constraint node is in the constrained object's history, listed and automatically selected in the Channel Box under SHAPES.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Set the channels.
3. Click on a channel name with the left mouse button.
4. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

## Edit aim constraint attributes

#### To edit attributes with Attribute Editor

1. Select the aim constraint node.
2. Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).
3. The following sections make available attributes: Transform Attributes, Aim Constraint Attributes, Pivots, Limit Information, Display, Node Behavior, and Extra Attributes.
4. Do one of the following:

- Click Select to select the node you are now editing as the currently selected object in your scene.
- Click Load Attributes to load the attribute values of the currently selected node.
- Click Close to close the Attribute Editor.

## Add target objects for aim constraints

After you've created an aim constraint, you can add more target objects for additional control over the constrained object's orientation. Adding more target objects is similar to creating aim constraints.

#### To add target objects

1. Select one or more objects you want to add as target objects, followed by the constrained object.
2. SelectConstrain > Aim > ❐.

The Aim Constraint Options window appears.

3. Be sure that Add Targets is selected as the Constraint Operation.
4. Click Add to add the selected objects as target objects.

   The constrained object's position changes (if Maintain Offset is off), indicating that it is now constrained by the objects you've just added as target objects.

## Remove target objects for aim constraints

After you've created an aim constraint, you can remove any of the target objects so that the objects no longer constrain the constrained object. Removing target objects is similar to adding target objects.

Note that when you remove a target object, you also remove any animation curves attached to the constraint object for that target object.

For this procedure, see "To modify which axes of a constrained object are affected by the constraint" on page 486.

## Change target object weights for aim constraints

A target object's weight specifies how much the orientation of the constrained object can be influenced by a target object. The weights are attributes of the aim constraint. For each target object, an attribute named *targetObject* W*n* is included that specifies the weight of each target object. By default, the weights are set to 1, which gives each target object an equal influence over the constrained object's orientation. However, you can change the weights so that some target objects can have more (or less) influence than others. You can change target object weights with the Channel Box or the Attribute Editor.

#### To change target object weights for aim constraints with the Channel Box

Edit the *targetObject* W*n* channels as described in "Edit point constraint channels" on page 461.

#### To change target object weights for aim constraints with the Attribute Editor

Edit the *targetObject* W*n* attributes as described in "Edit point constraint attributes" on page 461.

## Prevent rolling effects for aim constraints

In certain situations, a constrained object can rapidly roll about its aim vector. Rolling effects can happen when the aim vector approaches or points in the same direction or in the opposite direction as the up vector. For more information, see "Rolling effects" on page 431.

You can avoid rolling effects by keeping the target point clear of the world up vector's direction. For example, if the world up vector points in the direction of the scene's world space Y-axis (the default), you would try to avoid having the positive or negative Y-axis intersect the target point. You could move the target object(s) as needed, or perhaps change the target object weights so that the target point does not get to close to the Y-axis.

However, if your animation makes such avoidances impossible, you can prevent rolling by changing or animating the world up vector.

#### To change world up vector for aim constraints with the Attribute Editor

Edit the World Up Vector attribute as described in "Edit aim constraint attributes" on page 463. Note that you can also use the Channel Box to edit the World Up Vector.

#### To animate world up vector for aim constraints with the Channel Box

You can set keys on the World Up Vector attribute by using the Channel Box. To select the World Up Vector attribute, see "Editing aim constraint channels with Channel Box" on page 25. To set keys, after you select the attribute press the right mouse button and select Key Selected.

## Control motion history dependence effects for aim constraints

In certain situations, a constrained object can become motion history dependent. For more information, see "Motion history dependence effects" on page 431.

You can control motion history dependence by making sure that the aim vector and the up vector do not point in the same direction. In they are pointing in the same direction, the best way to prevent motion history dependence is to change the up vector's direction. You could also change the aim vector, but it's likely that you choose the aim vector so that the object aims in a particular way.

Additionally, if the aim constraint's World Up Type is set to None, the constrained object can be motion history dependent.

#### To change up vector or aim vector direction with Attribute Editor

Check the Aim Vector and Up Vector attributes as described in "Edit aim constraint attributes" on page 463. If they are the same, edit one of the them so that they do not both point in the same direction.

#### To change World Up Type attribute with Attribute Editor

Check the World Up Type attribute as described in "Edit aim constraint attributes" on page 463. If set to None, the constrained object can be motion history dependent.

## Delete aim constraints

To delete an aim constraint, delete the aim constraint node.

#### To delete an aim constraint

1. Select the aim constraint node only. (Select the aim constraint's selection handle if displayed, or use the Hypergraph to select the aim constraint node.)
2. Select Edit > Delete (default shortcut: Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key).

## Examples

This section includes two examples of implementing aim constraints.

## Aiming a sphere at a sphere

**To setup the two spheres**

1. Create a NURBS sphere.
2. Move the sphere some distance away from the center of the scene.
3. Create another NURBS sphere. Leave it at the scene's origin.
4. Display the sphere's local rotation axis (Display > Component Display > Local Rotation Axes).

**To create an aim constraint**

1. Select the you moved sphere, and then select the sphere at the origin.
2. If you are sure that the constraint options have their default settings, select Constrain > Aim. (To be sure that you are using the defaults, select Constrain > Aim ❐. Click Reset, and then click Add.)

   Now you have constrained the sphere at the origin to aim at the other sphere.

**To use the aim constraint**

Select the sphere you moved (nurbsSphere1), and select the Move Tool. As you move the sphere, the other sphere (nurbsSphere2) will rotate accordingly.

Note how nurbsSphere2's local rotation X-axis always points at the nurbsSphere1. Also, note how nurbsSphere2's local rotation Y-axis always tries to point as closely as possible in the same direction as the scene's Y-axis.

By default, the aim vector causes nurbsSphere2's local rotation X-axis to point at nurbsSphere1. Also, by default, the up vector causes nurbsSphere2's Y-axis to align itself as closely as possible with the scene's Y-axis.

## Aim a cone at a sphere

**To create sphere and cone**

1. Create a NURBS sphere.
2. Move the sphere some distance away from the scene's origin.
3. Create a NURBS cone.

**To create an aim constraint**

1. Select the sphere, and then select the cone.
2. Select Constrain > Aim > ❐.

By default, the aim vector will direct the cone to point at the sphere along its local rotation positive X-axis. However, the cone narrows along its local positive Y-axis. You could change the orientation of the cone's local rotation axis, or you could set the aim vector to direct the cone to point along its local positive Y-axis.

For now, orient the aim vector to point along the cone's local positive Y-axis.

3 Set Aim Vector to 0.0, 1.0, 0.0. (The default is 1.0, 0.0, 0.0.)

   The aim vector will now point along the cone's local positive Y-axis instead of the X-axis.

   By default, the up vector points along the cone's local positive Y-axis. If the aim vector and the up vector point in the same direction, the constrained object will be motion history dependent.To prevent this, you can change the up vector's direction.

4 Set Up Vector to 0.0, 0.0, 1.0.

   The up vector will now point along the cone's local positive Z-axis.

   For convenience, set the world up vector to point in the same direction relative to world space as the up vector does relative to the cone's local space.

5 Set World Up Vector to 0.0, 0.0, 1.0.

6 Click Add.

7 Now you have constrained the cone to aim at the sphere.

#### To use the constraint

Select the sphere, and select the Move Tool. As you move the sphere, the cone will always point at the sphere.

### Edit orient constraints

## Edit orient constraint channels

#### To edit orient constraint channels with the Channel Box

1 In the scene, select the constrained object.

   The orient constraint node is in the constrained object's history, listed and automatically selected in the Channel Box under SHAPES.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).

2 Set the channels.

3 Click on a channel name with the left mouse button.

4 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

## Edit orient constraint attributes

#### To edit orient constraint attributes with the Attribute Editor

1. Select the orient constraint node.
2. Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).
3. The following sections make available attributes: Transform Attributes, Orient Constraint Attributes, Pivots, Limit Information, Display, Node Behavior, and Extra Attributes.
   - Click Select to select the node you are now editing as the currently selected object in your scene.
   - Click Load Attributes to load the attribute values of the currently selected node.
   - Click Close to close the Attribute Editor.

## Remove orient constraint target objects

After you've created an orient constraint, you can remove any of the target objects so that the objects no longer constrain the constrained object. Removing target objects is similar to adding target objects.

Note that when you remove a target object, you also remove any animation curves attached to the constraint object for that target object.

For this procedure, see "To remove a target object from a constraint" on page 487.

## Change orient constraint target object weights

A target object's weight specifies how much the orientation of the constrained object can be influenced by a target object. The weights are attributes of the orient constraint. For each target object, an attribute named *targetObject* W*n* is included that specifies the weight of each target object. By default, the weights are set to 1, which gives each target object an equal influence over the constrained object's orientation. However, you can change the weights so that some target objects can have more (or less) influence than others. You can change target object weights with the Channel Box or the Attribute Editor.

#### To change orient constraint target object weights with the Channel Box

Edit the *targetObject* W*n* channels as described in ”Edit orient constraint channels” on page 468.

#### To change orient constraint target object weights with the Attribute Editor

Edit the *targetObject* W*n* attributes as described in ”Edit orient constraint attributes” on page 469.

## Animate target object weights for orient constraints

An interesting technique you can use with orient constraints is to animate the target object weights specified by the *targetObject* W*n* channels. You can vary the weights from 0 to any value, so that as an animation progresses different target objects can take turns driving a constrained object’s orientation.

## Delete orient constraints

To delete an orient constraint, delete the orient constraint node.

#### To delete an orient constraint

1. Select the orient constraint node only. (Select the orient constraint’s selection handle if displayed, or use the Hypergraph to select the orient constraint node.)
2. Select Edit > Delete (default shortcut: Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key).

### Edit scale constraints

## Edit scale constraint channels

#### To edit scale constraint channels with the Channel Box

1. In the scene, select the constrained object.

   The scale constraint node is in the constrained object’s history, listed and automatically selected in the Channel Box under SHAPES.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).

2 In the Channel Box, the following channels are listed for the scale constraint:
3 Click on a channel name with the left mouse button.
4 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

## Edit scale constraint attributes

**To edit scale constraint attributes with the Attribute Editor**

1 Select the scale constraint node.
2 Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).
3 The following sections make available attributes: Transform Attributes, Scale Constraint Attributes, Pivots, Limit Information, Display, Node Behavior, and Extra Attributes.
4 Do one of the following:
   - Click Select to select the node you are now editing as the currently selected object in your scene.
   - Click Load Attributes to load the attribute values of the currently selected node.
   - Click Close to close the Attribute Editor.

## Add target objects for scale constraints

After you've created a scale constraint, you can add more target objects for additional control over the constrained object's position. Adding more target objects is similar to creating scale constraints.

**To add target objects for scale constraints**

1 Select one or more objects you want to add as target objects, followed by the constrained object.
2 Select Constrain > Scale >❐.
   The Scale Constraint Options window is displayed.
3 Be sure that Add Targets is selected as the Constraint Operation.
4 Click Add to add the selected objects as target objects.

The constrained object’s scaling changes (if Maintain Offset is off), indicating that it is now constrained by the objects you’ve just added as target objects.

## Remove target objects for scale constraints

After you’ve created a scale constraint, you can remove any of the target objects so that the objects no longer constrain the constrained object. Removing target objects is similar to adding target objects.

Note that when you remove a target object, you also remove any animation curves attached to the constraint object for that target object.

For this procedure, see ”To remove a target object from a constraint” on page 487.

## Change target object weights for scale constraints

A target object’s weight specifies how much the scaling of the constrained object can be influenced by a target object. The weights are attributes of the scale constraint. For each target object, an attribute named *targetObject* W*n* is included that specifies the weight of each target object. By default, the weights are set to 1, which gives each target object an equal influence over the constrained object’s scaling. However, you can change the weights so that some target objects can have more (or less) influence than others. You can change target object weights with the Channel Box or Attribute Editor.

#### To change target object weights for scale constraints with the Channel Box

Edit the *targetObject* W*n* channels as described in ”Edit scale constraint channels” on page 470.

#### To change target object weights for scale constraints with the Attribute Editor

Edit the *targetObject* W*n* attributes as described in ”Edit scale constraint attributes” on page 471.

## Animate target object weights for scale constraints

You can animate the target object weights specified by the *targetObject* W*n* channels. You can vary the weights from 0 to any value, so that as an animation progresses, different target objects can take turns driving a constrained object's scale.

## Delete scale constraints

To delete a scale constraint, delete the scale constraint node.

**To delete a scale constraint**

1. Select the scale constraint node only. (Select the scale constraint's selection handle if displayed, or use the Hypergraph to select the scale constraint node.)
2. Select Edit > Delete (default shortcut: Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key).

### Edit parent constraints

## Edit a parent constraint

You can edit a parent constraint's attributes in the Channel Box and Attribute Editor. When you select a constrained object, its constrained axes appear blue in the Channel Box and Attribute Editor. This means that any changes you make to the Parent constraint affects only those axes.

## Offset parent constraints

There are no offset values or fields present in the settings for Parent constraints. Unlike other constraints, you cannot offset Parent constrained objects from the Channel Box or options window. However, you can use the Maintain Offset option to set the offset of your constrained object or you can set a single key on the object to change its offset position.

**To offset a Parent Constrained object using Maintain Offset**

1. Select the constrained object.
2. Translate and rotate the object to where you want it positioned in your scene.
3. Select the target objects, followed by the constrained object.

For more information, search the online help for Targets.

4. Select Constrain > Parent Constraint > ❐.

The Parent Constraint Options window appears. See ”Constrain > Parent” on page 491.

5. Set the Parent constraint options to reflect the values of the constrained object’s channels and settings in the Channel Box.
6. Verify that Maintain Offset is on.
7. Click Add or Apply.

   The constrained object is now offset from its targets by its current rotation and translation.

### Edit geometry constraints

Editing geometry constraints is described in the following topics.

## Edit geometry constraint channels

#### To edit geometry constraint channels with the Channel Box

1. In the scene, select the constrained object.

   The geometry constraint node is in the constrained object’s history, listed and automatically selected in the Channel Box under SHAPES.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Set the channels.
3. Click on a channel name with the left mouse button.
4. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

## Edit geometry constraint attributes

#### To edit geometry constraint attributes with the Attribute Editor

1. Select the geometry constraint node.
2. Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).
3. Set the attributes.
4. Do the following:

- Click Select to select the node you are now editing as the currently selected object in your scene.
- Click Load Attributes to load the attribute values of the currently selected node.
- Click Close to close the Attribute Editor.

## Add target objects for geometry constraints

After you've created a geometry constraint, you can add more target objects for additional control over the constrained object's position. Adding more target objects is similar to creating geometry constraints.

### To add target objects for geometry constraints

1. Select one or more objects you want to add as target objects, followed by the constrained object.
2. Select Constrain > Geometry > ❐.

   The Geometry Constraint Options window is displayed.
3. Be sure that Add Targets is selected as the Constraint Operation.
4. Click Add to add the selected objects as target objects.

   The constrained object's position changes, indicating that it is now constrained by the objects you've just added as target objects.

## Remove target objects for geometry constraints

After you've created a geometry constraint, you can remove any of the target objects so that the objects no longer constrain the constrained object. Removing target objects is similar to adding target objects.

Note that when you remove a target object, you also remove any animation curves attached to the constraint object for that target object.

For this procedure, see "To remove a target object from a constraint" on page 487.

## Change target object weights for geometry constraints

A target object's weight specifies how much the position of the constrained object can be influenced by a target object. The weights are attributes of the geometry constraint. For each target object, an attribute named *targetObject* W*n* is included that specifies the weight of each target object. By default, the weights are set to 1, which gives each target object an equal influence over the constrained object's position. However, you can change the weights so that some target objects can have more (or less) influence than others. You can change target object weights with the Channel Box or the Attribute Editor.

#### To change target object weights for geometry constraints with the Channel Box

Edit the *targetObject* W*n* channels as described in ”Edit geometry constraint channels” on page 474.

#### To change target object weights for geometry constraints with the Attribute Editor

Edit the *targetObject* W*n* attributes as described in ”Edit geometry constraint attributes” on page 474.

## Delete geometry constraints

To delete a geometry constraint, delete the geometry constraint node.

#### To delete a geometry constraint

1 Select the geometry constraint node only. (Select the geometry constraint’s selection handle if displayed, or use the Hypergraph to select the geometry constraint node.)
2 Select Edit > Delete (default shortcut: Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key).

### Edit normal constraints

## Edit normal constraint channels

#### To edit normal constraint channels with the Channel Box

1 In the scene, select the constrained object.

   The normal constraint node is in the constrained object’s history, listed and automatically selected in the Channel Box under SHAPES.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2 Set the channels.
3 Click on a channel name with the left mouse button.

4 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

## Edit normal constraint attributes

### To edit normal constraint attributes with the Attribute Editor

1 Select the normal constraint node.
2 Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).
3 Set the attributes.
4 Do the following:
   - Click Select to select the node you are now editing as the currently selected object in your scene.
   - Click Load Attributes to load the attribute values of the currently selected node.
   - Click Close to close the Attribute Editor.

## Add target objects for normal constraints

After you've created a normal constraint, you can add more target objects for additional control over the constrained object's orientation. Adding more target objects is similar to creating normal constraints.

### To add target objects for normal constraints

1 Select one or more objects you want to add as target objects, followed by the constrained object.
2 Select Constrain > Normal > ❐.

   The Normal Constraint Options window is displayed.
3 Be sure that Add Targets is selected as the Constraint Operation.
4 Click Add to add the selected objects as target objects.

   The constrained object's position changes, indicating that it is now constrained by the objects you've just added as target objects.

## Remove normal constraint target objects

After you've created a normal constraint, you can remove any of the target objects so that the objects no longer constrain the constrained object. Removing target objects is similar to adding target objects.

Note that when you remove a target object, you also remove any animation curves attached to the constraint object for that target object.

For this procedure, see ”To modify which axes of a constrained object are affected by the constraint” on page 486.

## Change normal constraint target object weights

A target object’s weight specifies how much the orientation of the constrained object can be influenced by a target object. The weights are attributes of the normal constraint. For each target object, an attribute named *targetObject* W*n* is included that specifies the weight of each target object. By default, the weights are set to 1, which gives each target object an equal influence over the constrained object’s orientation. However, you can change the weights so that some target objects can have more (or less) influence than others. You can change target object weights with the Channel Box or the Attribute Editor.

#### To change target object weights for normal constraints with the Channel Box

Edit the *targetObject* W*n* channels as described in ”Edit normal constraint channels” on page 476.

#### To change target object weights for normal constraints with the Attribute Editor

Edit the *targetObject* W*n* attributes as described in ”Edit normal constraint attributes” on page 477.

## Prevent rolling effects for normal constraints

In certain situations, a constrained object can rapidly roll about its aim vector. Rolling effects can happen when the aim vector approaches or points in the same direction or in the opposite direction as the up vector. For more information, see ”Rolling effects” on page 431.

You can avoid rolling effects by keeping the target vector clear of the world up vector’s direction. For example, if the world up vector points in the direction of the scene’s world space Y-axis (the default), you would try to avoid having the positive or negative Y-axis point in the same direction as the target vector. You could move the target object(s) as needed, or perhaps change the target object weights so that the target vector does not get to close to the Y-axis.

However, if your animation makes such avoidances impossible, you can prevent rolling by changing or animating the world up vector.

### To change the world up vector for normal constraints with the Attribute Editor

Edit the World Up Vector attribute as described in "Edit normal constraint attributes" on page 477. Note that you can also use the Channel Box to edit the World Up Vector.

### To animate the world up vector for normal constraints with the Channel Box

You can set keys on the World Up Vector attribute by using the Channel Box. To select the World Up Vector attribute, see "Edit normal constraint channels" on page 476. To set keys, after you select the attribute press the right mouse button and select Key Selected.

## Control motion history dependence effects for normal constraints

In certain situations, a constrained object can become motion history dependent. For more information, see "Motion history dependence effects" on page 431.

You can control motion history dependence by making sure that the aim vector and the up vector do not point in the same direction. If they are pointing in the same direction, the best way to prevent motion history dependence is to change the up vector's direction. You could also change the aim vector, but it's likely that you choose the aim vector so that the object aims in a particular way.

Additionally, if the normal constraint's World Up Type is set to None, the constrained object can be motion history dependent.

### To change the up vector or aim vector direction for normal constraints with the Attribute Editor

Check the Aim Vector and Up Vector attributes as described in "Edit normal constraint attributes" on page 477. If they are the same, edit one of the them so that they do not both point in the same direction.

### To change the normal constraint World Up Type attribute with the Attribute Editor

Check the World Up Type attribute as described in "Edit normal constraint attributes" on page 477. If set to None, the constrained object can be motion history dependent.

## Delete normal constraints

To delete a normal constraint, delete the normal constraint node.

#### To delete a normal constraint

1. Select the normal constraint node only. (Select the normal constraint's selection handle if displayed, or use the Hypergraph to select the normal constraint node.)
2. Select Edit > Delete (default shortcut: Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key).

### Edit tangent constraints

## Edit tangent constraint channels

#### To edit tangent constraint channels with the Channel Box

1. In the scene, select the constrained object.

   The tangent constraint node is in the constrained object's history, listed and automatically selected in the Channel Box under SHAPES.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).
2. Set the channels.
3. Click on a channel name with the left mouse button.
4. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you less control.

## Edit tangent constraint attributes

#### To edit tangent constraint attributes with the Attribute Editor

1. Select the tangent constraint node.
2. Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).
3. The following sections make available attributes: Transform Attributes, Tangent Constraint Attributes, Pivots, Limit Information, Display, Node Behavior, and Extra Attributes.
4. Do the following:
   - Click Select to select the node you are now editing as the currently selected object in your scene.

- Click Load Attributes to load the attribute values of the currently selected node.
- Click Close to close the Attribute Editor.

## Add target objects for target constraints

After you've created a tangent constraint, you can add more target objects for additional control over the constrained object's orientation. Adding more target objects is similar to creating tangent constraints.

#### To add target objects for target constraints

1. Select one or more objects you want to add as target objects, followed by the constrained object.
2. Select Constrain > Tangent > ❐.

   The Tangent Constraint Options window is displayed.
3. Be sure that Add Targets is selected as the Constraint Operation.
4. Click Add to add the selected objects as target objects.

   The constrained object's position changes, indicating that it is now constrained by the objects you've just added as target objects.

## Remove target objects for tangent constraints

After you've created a tangent constraint, you can remove any of the target objects so that the objects no longer constrain the constrained object. Removing target objects is similar to adding target objects.

Note that when you remove a target object, you also remove any animation curves attached to the constraint object for that target object.

For this procedure, see "To remove a target object from a constraint" on page 487.

## Change target object weights for tangent constraints

A target object's weight specifies how much the orientation of the constrained object can be influenced by a target object. The weights are attributes of the tangent constraint. For each target object, an attribute named *targetObject* W*n* is included that specifies the weight of each target object. By default, the weights are set to 1, which gives each target object an equal influence over the constrained object's orientation. However, you can change the weights so that some target objects can have more (or less) influence than others. You can change target object weights with the Channel Box or the Attribute Editor.

**To change target object weights for tangent constraints with the Channel Box**

Edit the *targetObject* W*n* channels as described in ”Edit tangent constraint channels” on page 480.

**To change target object weights for tangent constraints with the Attribute Editor**

Edit the *targetObject* W*n* attributes as described in ”Edit tangent constraint attributes” on page 480.

## Prevent rolling effects for tangent constraints

In certain situations, a constrained object can rapidly roll about its aim vector. Rolling effects can happen when the aim vector approaches or points in the same direction or in the opposite direction as the up vector. For more information, see ”Rolling effects” on page 431.

You can avoid rolling effects by keeping the target vector clear of the world up vector’s direction. For example, if the world up vector points in the direction of the scene’s world space Y-axis (the default), you would try to avoid having the positive or negative Y-axis point in the same direction as the target vector. You could move the target object(s) as needed, or perhaps change the target object weights so that the target vector does not get to close to the Y-axis.

However, if your animation makes such avoidances impossible, you can prevent rolling by changing or animating the world up vector.

**To change world up vector for tangent constraints with the Attribute Editor**

Edit the World Up Vector attribute as described in ”Edit tangent constraint attributes” on page 480. Note that you can also use the Channel Box to edit the World Up Vector.

**To animate the world up vector for tangent constraints with Channel Box**

You can set keys on the World Up Vector attribute by using the Channel Box. To select the World Up Vector attribute, see ”Edit tangent constraint channels” on page 480. To set keys, after you select the attribute press the right mouse button and select Key Selected.

## Control motion history dependence effects for tangent constraints

In certain situations, a constrained object can become motion history dependent. For more information, see "Motion history dependence effects" on page 431.

You can control motion history dependence by making sure that the aim vector and the up vector do not point in the same direction. If they are pointing in the same direction, the best way to prevent motion history dependence is to change the up vector's direction. You could also change the aim vector, but it's likely that you choose the aim vector so that the object aims in a particular way.

Additionally, if the tangent constraint's World Up Type is set to None, the constrained object can be motion history dependent.

#### To change up vector or aim vector direction for tangent constraints with the Attribute Editor

Check the Aim Vector and Up Vector attributes as described in "Edit tangent constraint attributes" on page 480. If they are the same, edit one of the them so that they do not both point in the same direction.

#### To change the World Up Type attribute for tangent constraints with the Attribute Editor

Check the World Up Type attribute as described in "Edit tangent constraint attributes" on page 480. If set to None, the constrained object can be motion history dependent.

# Delete tangent constraints

To delete a tangent constraint, delete the tangent constraint node.

#### To delete a tangent constraint

1. Select the tangent constraint node only. (Select the tangent constraint's selection handle if displayed, or use the Hypergraph to select the tangent constraint node.)
2. Select Edit > Delete (default shortcut: Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key).

### Edit pole vector constraints

Editing pole vector constraints is described in the following topics.

## Edit pole vector constraint channels

**To edit pole vector constraint channels with the Channel Box**

1 In the scene, select the constrained pole vector's IK rotate plane handle.

   The pole vector constraint node is in the IK rotate plane handle's history, listed and automatically selected in the Channel Box under SHAPES.

   Note that you can control which attributes are listed as keyable attributes (channels) in the Channel Box with the Channel Control editor (select Window > General Editors > Channel Control).

2 Set the channels.

3 Click on a channel name with the left mouse button.

4 In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key gives you finer control, and pressing the Shift key gives you coarser control.

## Edit pole vector constraint attributes

**To edit pole vector constraint attributes with the Attribute Editor**

1 Select the pole vector constraint node.

2 Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a).

3 The following sections make available attributes: Transform Attributes, Pole Vector Constraint Attributes, Pivots, Limit Information, Display, Node Behavior, and Extra Attributes.

4 Do one of the following:

   - Click Select to select the node you are now editing as the currently selected object in your scene.
   - Click Load Attributes to load the attribute values of the currently selected node.
   - Click Close to close the Attribute Editor.

## Add target objects to a pole vector constraint

After you've created a pole vector constraint, you can add more target objects for additional control over the constrained pole vector's position. Adding more target objects is similar to creating pole vector constraints.

#### To add target objects to pole vector constraints

1. Select one or more objects you want to add as target objects, followed by the constrained pole vector.
2. Select Constrain > Pole Vector > ❐.

   The Pole Vector Constraint Options window is displayed.
3. Be sure that Add Targets is selected as the Constraint Operation.
4. Click Add to add the selected objects as target objects.

   The constrained pole vector's position changes, indicating that it is now constrained by the objects you've just added as target objects.

## Remove target objects from pole vector constraints

After you've created a pole vector constraint, you can remove any of the target objects so that the objects no longer constrain the constrained pole vector. Removing target objects is similar to adding target objects.

Note that when you remove a target object, you also remove any animation curves attached to the constraint object for that target object.

For this procedure, see "To modify which axes of a constrained object are affected by the constraint" on page 486.

## Change target object weights for pole vector constraints

A target object's weight specifies how much the position of the constrained pole vector can be influenced by a target object. The weights are attributes of the pole vector constraint. For each target object, an attribute named *targetObject* W*n* is included that specifies the weight of each target object. By default, the weights are set to 1, which gives each target object an equal influence over the constrained pole vector's end position. However, you can change the weights so that some target objects can have more (or less) influence than others. You can change target object weights with the Channel Box or the Attribute Editor.

#### To change target object weights for pole vector constraints with the Attribute Editor

Edit the *targetObject* W*n* attributes as described in "Edit pole vector constraint attributes" on page 484.

## Offset constrained pole vector's end position

The constrained pole vector's end position is driven by the target point, but you can offset the end position from the target point. To do so, edit the Constraint Offset and Offset Polarity attributes with the Attribute Editor.

#### To offset constrained pole vector's end position

Edit the Constraint Offset and Offset Polarity attributes as described in "Edit pole vector constraint attributes" on page 484.

By default, these attributes are not displayed as channels in the Channel Box. If you'd like to control them from the Channel Box, you can add them by using the Channel Control editor (select Window > General Editors > Channel Control.).

## Delete pole vector constraints

To delete a pole vector constraint, delete the pole vector constraint node.

#### To delete a pole vector constraint

1. Select the pole vector constraint node only. (Select the point constraint's selection handle if displayed, or use the Hypergraph to select the point constraint node.)
2. Select Edit > Delete (default shortcut: Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key).

# Modify Constraint Axes

## Modify Constraint Axes

#### To modify which axes of a constrained object are affected by the constraint

1. Select the constrained object and its targets.
2. Select Constrain > Modify Constrained Axis.

   The Modify Constrained Axis options window appears.
3. Check the boxes beside the Constraint axes (X, Y, Z). You can also turn on Maintain Offset to preserve the relative position, rotation, or scaling of the constrained object.
4. To finish setting the node's new axes, do one of the following:
   - Click Modify to change which of the constrained object's axes are influenced by the target object. The Modify Constrained Axis options window closes.

- Click Apply to change which of the constrained object's axes are influenced by the target object. The Modify Constrained Axis options window remains open.
- Click Close to cancel the modify constrained axis operation. The Modify Constrained Axis options window closes.

# Remove a target object

## Remove a target object

### To remove a target object from a constraint

1. Select the driver object and then Ctrl-click the target object from which you want to remove the constraint connection.
2. Select Constrain > Remove Target > ❒.

   The Remove Target Options window appears.
3. Check the constraint types whose connections you want to sever.
   - If you want to sever all constraint connections between the selected driver and target object, turn on All.
   - If you want to preserve the relative transformation (translation, rotation, scale, and so on) of the target object, turn on Maintain Offset.
4. To finish this operation, do one of the following:
   - Click Remove to remove the specified connection between the selected driver object and the target object. The Remove Target options window closes.
   - Click Apply to remove the specified connection between the selected driver object and the target object. The Remove Target options window remains open.
   - Click Close to cancel the remove target operation. The Remove Target options window closes.

## What went wrong?

# Character Setup

## Troubleshooting constraints

### Constraint error messages

#### Warning: `No Flip` can only be used when all three rotation channels are driven by the constraint.

##### Cause

You tried to change an orient constraint's interpolation type to *Flip Mode*, but not all the constrained object's rotation channels are driven by the constraint.

The orient constraint works in quaternions, so its solution involves all three rotation axes. That is why when an orient constraint drives only one or two of a constrained object's rotation channels, the No Flip interpolation type has no effect.

## Reference

# Menus

## Animation menu set

### Constrain >

### Constrain > Point

#### Constrain > Point > ❐

##### Maintain Offset

Preserves the original (state prior to constraining), relative translation of the constrained object. Use this option to maintain spatial relationships between constrained objects.

##### Offset

Specifies an offset position (translate X, Y, and Z) for the constrained object relative to the target point. Note that the target point is the position of the target object's rotate pivot, or the average position of the rotate pivots of the target objects. Default values are all 0.

### Constraint Axes

Determines whether the point constraint is restricted to a specific axis (X, Y, Z) or to All axes. When All is checked, the X, Y, and Z boxes are dimmed.

### Weight

Specifies how much the position of the constrained object can be influenced by the target object(s). Use slider to select values from 0.0000 to 10.0000. Default is 1.0000.

# Constrain > Aim

## Constrain > Aim > ❐

### Maintain Offset

Preserves the original (state prior to constraining), relative translation and rotation of the constrained object. Use this option to maintain spatial and rotational relationships between constrained objects.

### Offset

Specifies an offset position (translate X, Y, and Z) for the constrained object relative to the target point. Note that the target point is the position of the target object's rotate pivot, or the average position of the rotate pivots of the target objects. Default values are all 0.

### Aim Vector

Specifies the direction of the aim vector relative to the constrained object's local space. The aim vector will point at the target point, forcing the constrained object to orient itself accordingly. The default specifies that the object's local rotation positive X-axis aligns with the aim vector to point at the target point (1.0000, 0.0000, 0.0000).

### Up Vector

Specifies the direction of the up vector relative to the constrained object's local space. The default specifies that the object's local rotation positive Y-axis will align with the up vector. In turn, by default, the up vector will try to align with the world up vector. Further, by default, the world up vector will point in the direction of the world space's positive Y-axis (0.0000, 1.0000, 0.0000).

If you define the up vector to point in the same direction as the aim vector, the constrained object will be motion history dependent. For more information, see "Motion history dependence effects" on page 431.

### World Up Vector

Specifies the direction of the world up vector relative to the scene's world space. Because Maya's world space is "Y-up" by default, the default world up vector points in the direction of the world space's positive Y-axis (0.0000, 1.0000, 0.0000).

### World Up Object

Specifies that the up vector try to aim at the origin of a specified object instead of aligning with the world up vector. The world up vector is ignored. The object whose origin the up vector tries to aim at is called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the up vector tries to aim at the origin of the scene's world space.

### Constraint Axes

Determines whether the Aim constraint is restricted to a specific axis (X, Y, Z) or to All axes. When All is checked, the X, Y, and Z boxes are dimmed.

### Weight

Specifies how much the orientation of the constrained object can be influenced by the target object(s). Use slider to select values from 0.0000 to 10.0000. Default is 1.0000.

# Constrain > Orient

## Constrain > Orient > ❐

### Maintain Offset

Preserves the original (state prior to constraining), relative rotation of the constrained object. Use this option to maintain rotational relationships between constrained objects.

### Offset

Specifies an offset position (translate X, Y, and Z) for the constrained object relative to the target point. Note that the target point is the position of the target object's rotate pivot, or the average position of the rotate pivots of the target objects. Default values are all 0.

### Constraint Axes

Determines whether the orient constraint is restricted to a specific axis (X, Y, Z) or to All axes. When All is checked, the X, Y, and Z boxes are dimmed.

#### Weight

Specifies how much the position of the constrained object can be influenced by the target object(s). Use slider to select values from 0.0000 to 10.0000. Default is 1.0000.

## Constrain > Scale

### Constrain > Scale > ❐

#### Maintain Offset

Preserves the original (state prior to constraining), relative scaling of the constrained object. Use this option to maintain scaling relationships between constrained objects.

#### Offset

Specifies an offset position (translate X, Y, and Z) for the constrained object relative to the target point. Note that the target point is the position of the target object's rotate pivot, or the average position of the rotate pivots of the target objects. Default values are all 0.

#### Constraint Axes

Determines whether the scale constraint is restricted to a specific axis (X, Y, Z) or to All axes. When All is checked, the X, Y, and Z boxes are dimmed.

#### Weight

Specifies how much the scaling of the constrained object can be influenced by the target object(s). Use slider to select values from 0.0000 to 10.0000. Default is 1.0000.

## Constrain > Parent

### Constrain > Parent > ❐

#### Maintain Offset

Preserves the original (state prior to constraining), relative translation and rotation of the constrained object. Use this option to maintain spatial and rotational relationships between constrained objects. See ”Offset parent constraints” on page 473.

#### Constraint Axes

Determines whether the parent constraint is restricted to a specific axis (X, Y, Z) or to All axes. When All is checked, the X, Y, and Z boxes are dimmed.

#### Weight

This is useful only when there are multiple target objects.

Sets the amount of influence that a target object’s position and rotation has on the constrained object. If the weight is 0.0000, the target object has no influence over the translation and rotation of the constrained object. If the weight is 1.0000, the target has full control over the constrained object’s translation and rotation. Default weight is 1.0000.

## Constrain > Geometry

### Constrain > Geometry > ❐

Weight

Specifies how much the position of the constrained object can be influenced by the target object(s). Use slider to select values from 0.0000 to 10.0000. Default is 1.0000.

## Constrain > Normal

### Constrain > Normal > ❐

#### Weight

Specifies how much the orientation of the constrained object can be influenced by the target object(s). Use slider to select values from 0.0000 to 10.0000. Default is 1.0000.

#### Aim Vector

Specifies the direction of the aim vector relative to the constrained object’s local space. The aim vector will align with the target vector, forcing the constrained object to orient itself accordingly. The default specifies that the object’s local rotation positive X-axis aligns with the aim vector to align with the target vector (1.0000, 0.0000, 0.0000).

#### Up Vector

Specifies the direction of the up vector relative to the constrained object’s local space. The default specifies that the object’s local rotation positive Y-axis will align with the up vector. In turn, by default, the up vector will try to align with the world up vector. Further, by default, the world up vector will point in the direction of the world space’s positive Y-axis (0.0000, 1.0000, 0.0000).

If you define the up vector to point in the same direction as the aim vector, the constrained object will be motion history dependent. For more information, see "Motion history dependence effects" on page 431.

#### World Up Vector

Specifies the direction of the world up vector relative to the scene's world space. Because Maya's world space is "Y-up" by default, the default world up vector points in the direction of the world space's positive Y-axis (0.0000, 1.0000, 0.0000).

#### World Up Object

Specifies that the up vector try to aim at the origin of a specified object instead of aligning with the world up vector. The world up vector is ignored. The object whose origin the up vector tries to aim at is called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the up vector tries to aim at the origin of the scene's world space.

# Constrain > Tangent

## Constrain > Tangent > ❐

#### Weight

Specifies how much the orientation of the constrained object can be influenced by the target object(s). Use slider to select values from 0.0000 to 10.0000. Default is 1.0000.

#### Aim Vector

Specifies the direction of the aim vector relative to the constrained object's local space. The aim vector will align with the target vector, forcing the constrained object to orient itself accordingly. The default specifies that the object's local rotation positive X-axis aligns with the aim vector to align with the target vector (1.0000, 0.0000, 0.0000).

#### Up Vector

Specifies the direction of the up vector relative to the constrained object's local space. The default specifies that the object's local rotation positive Y-axis will align with the up vector. In turn, by default, the up vector will try to align with the world up vector. Further, by default, the world up vector will point in the direction of the world space's positive Y-axis (0.0000, 1.0000, 0.0000).

If you define the up vector to point in the same direction as the aim vector, the constrained object will be motion history dependent. For more information, see ”Motion history dependence effects” on page 431.

#### World Up Vector

Specifies the direction of the world up vector relative to the scene’s world space. Because Maya’s world space is “Y-up” by default, the default world up vector points in the direction of the world space’s positive Y-axis (0.0000, 1.0000, 0.0000).

#### World Up Object

Specifies that the up vector try to aim at the origin of a specified object instead of aligning with the world up vector. The world up vector is ignored. The object whose origin the up vector tries to aim at is called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the up vector tries to aim at the origin of the scene's world space.

## Constrain > Pole Vector

### Constrain > Pole Vector > ❐

#### Weight

Specifies how much the constrained pole vector’s end position can be influenced by the target object(s). Use slider to select values from 0.0000 to 10.0000. Default is 1.0000.

## Constrain > Remove Target

### Constrain > Remove Target > ❐

With Remove Target, you can sever the connection between a target object and a constrained object without deleting the constraint node. When you remove a target from a constrained object, it no longer affects the constrained object.

## Constrain > Set Rest Position

With Set Rest Position, you can set where in world space your constrained object is positioned when its target weight is 0. Rest position gives you control of the constrained object, rather than relying on the last position of the target objects. Setting the rest position prevents constrained objects from sticking to undesired positions in the scene when their target weights are set to 0.

For example, a book is point constrained to a character's hand. When the character places the book on the table, the book's position on the table is set as its rest position. This prevents the book from popping back to the model's hands when the weight of the hand changes to 0. The books target (the model's hand) weight is then set to 0. The hand now no longer influences the book's position. The character then proceeds to release the book, turn, and walk away from the table. The book does not snap back to the hand because its rest position was set.

## Constrain > Modify Constrained Axis

Modify Constrained Axis lets you change which axes of a constrained object are influenced by the target object. Use this feature when you want to modify the axes of a constraint. Modify Constrained Axis can only be used in conjunction with the following constraint types: point, orient, scale, aim and parent.

# Nodes

## Character nodes

## General constraint node attributes

The node behavior attributes include Caching and Node State.

### Pivots

Selections for displaying the point constraint's local rotate and scale pivots in local or world space.

### Limit Information

For Maya internal use only: attributes inherited from transform node.

### Display

Selections for selection handle display attributes, including handle display, local axis display, selection handle position (relative to current Translate X, Y, and Z attribute values), default manipulator display selections, visibility, and template. Bounding Box Information and Drawing Overrides not applicable.

## Node Behavior

### Caching

Specifies that Maya store the results of input evaluations, and then provide those results to the node. This saves Maya from having to re-evaluate the input nodes every time the node needs the results. If there are no changes to the input nodes, then this setting can improve display performance with no loss of results. However, note that caching uses more memory than would otherwise be used, which could adversely affect performance. Also, if there are changes to input nodes, more memory is allocated and then freed, which could also adversely affect display performance.

### Node State

Set Node State to Normal, HasNoEffect, Blocking, Waiting-Normal, Waiting-HasNoEffect, or Waiting-Blocking. (Note that for constraints the Node State attribute is available as a channel in the Channel Box.)

| | |
|---|---|
| Normal | Specifies that Maya evaluate and display the constraint. Maya will evaluate the node as usual. This is the default. |
| HasNoEffect | Specifies that Maya prevent the constraint, but display the object. Maya will evaluate the nodes in the node's history, but not the node itself. |
| Blocking | Specifies that Maya prevent the constraint, and not display the object. Maya will not report the results of any evaluations of input nodes to this node. |
| Waiting-Normal | (For Maya internal use only.) Specifies that if the dependency graph evaluation refresh performance setting (Window > Settings/ Preferences > Performance Settings) is set to Demand or Release, the node will take the Normal state when you click Update or release the mouse button. |
| Waiting-HasNoEffect | (For Maya internal use only.) Specifies that if the dependency graph evaluation refresh performance setting is set to Demand or Release, the node will take the HasNoEffect state when you click Update or release the mouse button. |

| | |
|---|---|
| Waiting-Blocking | (For Maya internal use only.) Specifies that if the dependency graph evaluation refresh performance setting is set to Demand or Release, the node will take the Blocking state when you click Update or release the mouse button. |

### Extra Attributes

Lists the weights for each target object. Their initial values are all from the weight creation option.

## Rest Position attributes

The rest position settings for each constraint are located in the constrained object's *constraint* attributes.

#### Enable Rest Position

If on, the rest position is active for the constrained object. If off, the rest position is inactive for the constrained object. Turning Enable Rest Position off does not remove it from the constrained object.

#### Rest Translate

Sets the X, Y, and Z translation for the constrained object's Translate rest position. These fields are present for only Point and Parent constraints.

#### Rest Rotate

Sets the X, Y, and Z rotation for the constrained object's Rotate rest position. These fields are present only for Orient, Aim, Parent, Tangent, and Normal constraints.

#### Rest Scale

Sets the X, Y, and Z scaling for the constrained object's Scale rest position. These fields are present only for Scale constraints.

## targetObject W*n*

Specifies a target object's weight. The weight specifies how much the target point, which drives the position of the constrained object, can be influenced by a target object. The *n* in W*n* is an identifier for each target object, starting from 0.

# pointConstraint

## Point Constraint Attributes

### Offset

Specifies an offset position (translate X, Y, and Z) for the constrained object relative to the target point. Note that the target point is the position of the target object's rotate pivot, or the average position of the rotate pivots of the target objects. Default values are all 0.

### Offset Polarity

Specifies the polarity of the Constraint Offset. In effect, the Constraint Offset values are multiplied by the Offset Polarity to give the constrained object's position. Default is 1.

### Constraint Translate

Informs you of the constrained object's current position. Useful to know when you are specifying the Constraint Offset and Offset Polarity.

### Enable Rest Position

See "Rest Position attributes" on page 497.

### Rest Translate

See "Rest Position attributes" on page 497.

# aimConstraint

## Aim Constraint Attributes

### Offset

Specifies an offset position for the constrained object relative to the target point. Note that the target point is the position of the target object's rotate pivot, or the average position of the rotate pivots of the target objects. Default values are all 0.

### Aim Vector

Specifies the direction of the aim vector relative to the constrained object's local space. The aim vector points at the target point, forcing the constrained object to orient itself accordingly. The default specifies that the object's local rotation positive X-axis aligns with the aim vector to point at the target point (1.0000, 0.0000, 0.0000).

### Up Vector

Specifies the direction of the up vector relative to the constrained object's local space. The default specifies that the object's local rotation positive Y-axis aligns with the up vector. In turn, by default, the up vector tries to align with the world up vector. Further, by default, the world up vector points in the direction of the world space's positive Y-axis (0.0000, 1.0000, 0.0000).

If you define the up vector to point in the same direction as the aim vector, the constrained object will be motion history dependent. For more information, see "Motion history dependence effects" on page 431.

### World Up Type

Specifies the role of the world up vector. Selections include Scene Up, Object Up, Object Rotation Up, Vector, and None.

| | |
|---|---|
| Scene Up | Specifies that the up vector try to align with the scene's up axis instead of the world up vector. The world up vector is ignored.<br><br>(To specify the scene's up axis, select Window > Settings/Preferences > Preferences. In the Settings category of the Preferences window, select Y or Z for the Up Axis of the World Coordinate System. Y is the default.) |
| Object Up | Specifies that the up vector try to aim at the origin of a specified object instead of aligning with the world up vector. The world up vector is ignored. The object whose origin the up vector tries to aim at is called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the up vector tries to aim at the origin of the scene's world space. |
| Object Rotation Up | Specifies that the world up vector is defined relative to some object's local space instead of the scene's world space. The up vector tries to align with the world up vector after transforming it relative to the scene's world space. The object whose origin the up vector tries to aim at is called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the world up vector is defined relative to the scene's world space. |

| | |
|---|---|
| Vector | Specifies that the up vector tries to align with world up vector as closely as possible. The world up vector is defined relative to the scene’s world space. (This is the default.) |
| None | Specifies no calculation of the constrained object’s orientation about the aim vector. The orientation continues as whatever the orientation is right before you specify None. With None selected, the constrained object becomes motion history dependent. For more information, see ”Motion history dependence effects” on page 431. |

#### World Up Vector

Specifies the direction of the world up vector relative to the scene’s world space. Because Maya’s world space is “Y-up” by default, the default world up vector points in the direction of the world space’s positive Y-axis (0.0000, 1.0000, 0.0000).

#### Constraint Rotate

Informs you of the current orientation of the constrained object.

#### Constraint Vector

Informs you of the current target point, which is what the aim vector aims at.

#### Enable Rest Position

See ”Rest Position attributes” on page 497.

#### Rest Rotate

See ”Rest Position attributes” on page 497.

## orientConstraint

### Orient Constraint Attributes

#### Offset

Specifies an offset position for the constrained object relative to the target point. Note that the target point is the position of the target object’s rotate pivot, or the average position of the rotate pivots of the target objects. Default values are all 0.

#### Interp Type

Sets the rotation interpolation type for the constraint. Interp Type is relevant only when the constrained object has multiple targets.

| | |
|---|---|
| Average | Averages rotation between the constrained object and its targets. |
| Shortest | Finds the shortest path between rotations from the constrained object to its targets using quaternion interpolation. |
| Longest | Finds the longest path between rotations from the constrained object to its targets using quaternion interpolation. This path is in the opposite direction from Shortest. |

> Tip Set keyframes between Shortest and Longest rotation interpolation to prevent flipping.

| | |
|---|---|
| No Flip | No Flip caches the current rotation onto the constrained object. Once you select No Flip, you must click the Cache button to create the cache. |

### Cache

Cache appears in the Interp Type menu field when there is cached interpolation on the constrained object.

| | |
|---|---|
| Create | Caches the current rotation interpolation onto the constrained object. Caching reduces the chance of flipping. Only available when you select No Flip. |
| Delete | Deletes the current rotation interpolation cache from the constrained object. Only available when Cache appears in the Interp Type menu field. |

### Constraint Rotate

Informs you of the constrained object’s current orientation.

### Enable Rest Position

See ”Rest Position attributes” on page 497.

### Rest Translate

See ”Rest Position attributes” on page 497.

# scaleConstraint

## Scale Constraint Attributes

### Offset

Specifies an offset position (translate X, Y, and Z) for the constrained object relative to the target point. Note that the target point is the position of the target object's rotate pivot, or the average position of the rotate pivots of the target objects. Default values are all 0.

### Constraint Scale

Informs you of the constrained object's current scaling.

### Enable Rest Position

See "Rest Position attributes" on page 497.

### Rest Scale

See "Rest Position attributes" on page 497.

# parentConstraint

In the Attribute Editor, you can view or edit the characteristics of the parent constraint node. To edit this node, you must first select its name in the Outliner or its tab in the Attribute Editor. If you selected it in the Outliner, go to Windows > Attribute Editor or use the shortcut Ctrl+a to view its attributes.

For more information, search the online help for Constraint node attributes.

The following section describes the attributes specific to the Parent constraint node (parentConstraint*).*

## Parent Constraint Attributes

### Interp Type

Sets the rotation interpolation type for the constraint. Interp Type is relevant only when the constrained object has multiple targets.

| | |
|---|---|
| Average | Averages rotation between the constrained object and its targets. |
| Shortest | Finds the shortest path between rotations from the constrained object to its targets using quaternion interpolation. |

Longest — Finds the longest path between rotations from the constrained object to its targets using quaternion interpolation. This path is in the opposite direction from Shortest.

No Flip and Cache — No Flip caches the current rotation onto the constrained object. Once you select No Flip, you must click the Cache button to create the cache. Cache appears in the Interp Type menu field when there is cached interpolation on the constrained object.

Cache Create button — Caches the current rotation interpolation onto the constrained object and creates an animation curve for the current rotation interpolation on the constraint's node. Caching reduces the chance of flipping. Only available when you select No Flip.

Cache Delete button — Deletes the current rotation interpolation cache from the constrained object. Only available when Cache appears in the Interp Type menu field.

### Constraint Translate

Displays the constrained object's current position. This value can be useful when specifying the Offset Translate value. This field is read-only.

### Constraint Rotate

Displays the constrained object's current rotation. This value can be useful when specifying the Offset Rotate value. This field is read-only.

### Enable Rest Position

See "Rest Position attributes" on page 497.

### Rest Translate

See "Rest Position attributes" on page 497.

### Rest Rotate

See "Rest Position attributes" on page 497.

# normalConstraint

## Normal constraint attributes

### Aim Vector

Specifies the direction of the aim vector relative to the constrained object's local space. The aim vector aligns with the target vector, forcing the constrained object to orient itself accordingly. The default specifies that the object's local rotation positive X-axis aligns with the aim vector, which aligns with the target vector (1.0000, 0.0000, 0.0000).

### Up Vector

Specifies the direction of the up vector relative to the constrained object's local space. The default specifies that the object's local rotation positive Y-axis aligns with the up vector. In turn, by default, the up vector tries to align with the world up vector. Further, by default, the world up vector points in the direction of the world space's positive Y-axis (0.0000, 1.0000, 0.0000).

If you define the up vector to point in the same direction as the aim vector, the constrained object will be motion history dependent. For more information, see "Motion history dependence effects" on page 431.

### World Up Vector

Specifies the direction of the world up vector relative to the scene's world space. Because Maya's world space is "Y-up" by default, the default world up vector points in the direction of the world space's positive Y-axis (0.0000, 1.0000, 0.0000).

### World Up Type

Specifies the role of the world up vector. Selections include Scene Up, Object Up, Object Rotation Up, Vector, and None.

| | |
|---|---|
| Scene Up | Specifies that the up vector try to align with the scene's up axis instead of the world up vector. The world up vector is ignored.<br><br>(To specify the scene's up axis, select Window > Settings/Preferences > Preferences. In the Settings category of the Preferences window, select Y or Z for the Up Axis of the World Coordinate System. Y is the default.) |
| Object Up | Specifies that the up vector try to aim at the origin of a specified object instead of aligning with the world up vector. The world up vector is ignored. The object whose origin the up vector tries to aim at is |

called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the up vector tries to aim at the origin of the scene's world space.

Object Rotation Up
: Specifies that the world up vector is defined relative to some object's local space instead of the scene's world space. The up vector tries to align with the world up vector after transforming it relative to the scene's world space. The object whose origin the up vector tries to aim at is called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the world up vector is defined relative to the scene's world space.

Vector
: Specifies that the up vector tries to align with world up vector as closely as possible. The world up vector is defined relative to the scene's world space. (This is the default.)

None
: Specifies no calculation of the constrained object's orientation about the aim vector. The orientation continues as whatever the orientation is right before you specify None. With None selected, the constrained object becomes motion history dependent. For more information, see "Motion history dependence effects" on page 431.

### World Up Object

Specifies that the up vector try to aim at the origin of a specified object instead of aligning with the world up vector. The world up vector is ignored. The object whose origin the up vector tries to aim at is called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the up vector tries to aim at the origin of the scene's world space.

### Constraint Rotate

Informs you of the current orientation of the constrained object.

### Constraint Vector

Informs you of the current target vector, which is what the aim vector aligns with.

### Enable Rest Position

See "Rest Position attributes" on page 497.

### Rest Rotate

See "Rest Position attributes" on page 497.

# tangentConstraint

## Tangent Constraint Attributes

### Aim Vector

Specifies the direction of the aim vector relative to the constrained object's local space. The aim vector aligns with the target vector, forcing the constrained object to orient itself accordingly. The default specifies that the object's local rotation positive X-axis aligns with the aim vector, which aligns with the target vector (1.0000, 0.0000, 0.0000).

### Up Vector

Specifies the direction of the up vector relative to the constrained object's local space. The default specifies that the object's local rotation positive Y-axis aligns with the up vector. In turn, by default, the up vector tries to align with the world up vector. Further, by default, the world up vector points in the direction of the world space's positive Y-axis (0.0000, 1.0000, 0.0000).

If you define the up vector to point in the same direction as the aim vector, the constrained object will be motion history dependent. For more information, see "Motion history dependence effects" on page 431.

### World Up Vector

Specifies the direction of the world up vector relative to the scene's world space. Because Maya's world space is "Y-up" by default, the default world up vector points in the direction of the world space's positive Y-axis (0.0000, 1.0000, 0.0000).

### World Up Type

Specifies the role of the world up vector. Selections include Scene Up, Object Up, Object Rotation Up, Vector, and None.

| | |
|---|---|
| Scene Up | Specifies that the up vector try to align with the scene's up axis instead of the world up vector. The world up vector is ignored. |

(To specify the scene's up axis, select Window > Settings/Preferences > Preferences. In the Settings category of the Preferences window, select Y or Z for the Up Axis of the World Coordinate System. Y is the default.)

| | |
|---|---|
| Object Up | Specifies that the up vector try to aim at the origin of a specified object instead of aligning with the world up vector. The world up vector is ignored. The object whose origin the up vector tries to aim at is called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the up vector tries to aim at the origin of the scene's world space. |
| Object Rotation Up | Specifies that the world up vector is defined relative to some object's local space instead of the scene's world space. The up vector tries to align with the world up vector after transforming it relative to the scene's world space. The object whose origin the up vector tries to aim at is called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the world up vector is defined relative to the scene's world space. |
| Vector | Specifies that the up vector tries to align with world up vector as closely as possible. The world up vector is defined relative to the scene's world space. (This is the default.) |
| None | Specifies no calculation of the constrained object's orientation about the aim vector. The orientation continues as whatever the orientation is right before you specify None. With None selected, the constrained object becomes motion history dependent. For more information, see "Motion history dependence effects" on page 431. |

### World Up Object

Specifies that the up vector try to aim at the origin of a specified object instead of aligning with the world up vector. The world up vector is ignored. The object whose origin the up vector tries to aim at is called the world up object. You can specify the world up object with the aimConstraint MEL command (use -wuo flag). If no world up object is specified, the up vector tries to aim at the origin of the scene's world space.

Constraint Rotate

Informs you of the current orientation of the constrained object.

Constraint Vector

Informs you of the current target vector, which is what the aim vector aligns with.

Enable Rest Position

See "Rest Position attributes" on page 497.

Rest Rotate

See "Rest Position attributes" on page 497.

## poleVectorConstraint

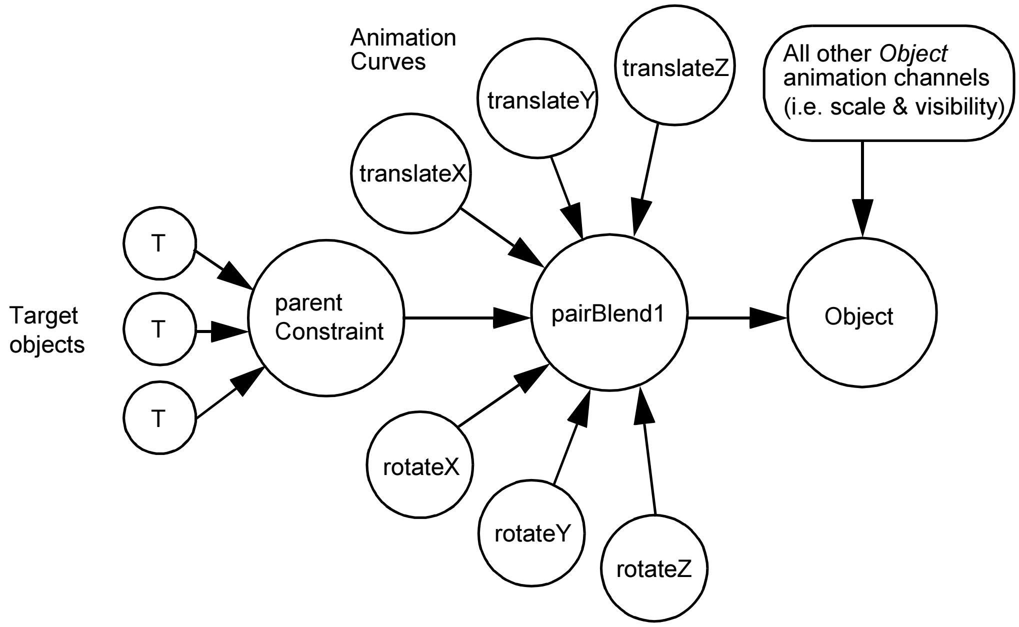
See "pointConstraint" on page 498.

## pairBlend

The pairBlend node is automatically generated when both keyframe animation and a constraint are applied to an object. Once the animation and constraint are linked to the pairBlend node, you can modify the weight of the animation-constraint blend to generate various effects. See "Blend Weights in the Channel Box" on page 511.

In the dependency graph, the pairBlend node functions as a link between constraints, animation, and the object to which both are applied. The pairBlend node connections resemble the following: all target objects connect to the constraint node, the constraint node and keyframe animation connect to the pairBlend node, and the pairBlend node connects to the object.

When multiple constraints are applied to the same object, a separate pairBlend node is created for each constraint and the animation channels of the constrained object connect to the same pairBlend node as their related constraints.




| Note | Input 1 always represents the keyframe animation on the constrained object and Input 2 always represents the constraint node. |
|---|---|

### Caching

Unlike other cache data in Maya (for example, caches in Dynamics, Cloth, and Fluid Effects), the cache data for pairBlend is also an animation curve.

For more information, search the online help for node Caching.

### Node State

For more information, search the online help for the Node State Attributes.

### InTranslate*n*

The InTranslate fields display the X, Y, and Z translation values for Input 1 and Input 2. *n* represents the input number.

### InRotate*n*

The InRotate fields display the X, Y, and Z rotation values for Input 1 and Input 2. *n* represents the input number.

### Weight

Displays a blend value that determines the amount of influence that the constraint and keyframe animation have on the constrained object. This field is read-only. See "Blend Constraintn" on page 511.

### TranslateXMode

| | |
|---|---|
| Blend Inputs | The blended X translation of Input 1 and Input 2 controls the driven object's X translation. |
| Input 1 only | The X translation of Input 1 controls the X translation of the driven object. |
| Input 2 only | The X translation of Input 2 controls the X translation of the driven object. |

### TranslateYMode

| | |
|---|---|
| Blend Inputs | The blended Y translation of Input 1 and Input 2 controls the driven object's Y translation. |
| Input 1 only | The Y translation of Input 1 controls the Y translation of the driven object. |
| Input 2 only | The Y translation of Input 2 controls the Y translation of the driven object. |

### TranslateZMode

| | |
|---|---|
| Blend Inputs | The blended Z translation of Input 1 and Input 2 controls the driven object's Z translation. |
| Input 1 only | The Z translation of Input 1 controls the Z translation of the driven object. |
| Input 2 only | The Z translation of Input 2 controls the Z translation of the driven object. |

### RotateMode

Rotation is calculated by blending the rotation of the specified inputs along all axes.

| | |
|---|---|
| Blend Inputs | The blended X, Y, and Z rotation of Input 1 and Input 2 controls the driven object's rotation along all axes. |
| Input 1 only | The X, Y, and Z rotation of Input 1 controls the driven object's rotation along all axes. |
| Input 2 only | The X, Y, and Z rotation of Input 2 controls the driven object's rotation along all axes. |

### Rotation Interpolation

**Euler Angle** Use Euler Angles when the pairBlend weight value is between 0 (driven by keyframes) and 1 (driven by the constraint). Euler is the default rotation interpolation.

For more information, search the online help for Independent Euler Angles.

| | |
|---|---|
| Note | Euler angles may cause flipping when used to interpolate between keyframes and constraints. If this occurs, we recommend that you switch to quaternion rotation. |

**Quaternion** Quaternions smoothly interpolate the keyframe-constraint blend without producing anomalies (such as flipping). Use Quaternion when the pairBlend weight value is 1 or 0.

For more information, search the online help for Synchronized Quaternions.

## Blend Weights in the Channel Box

| | |
|---|---|
| Note | When the driven object is selected, all its target and blend weight values can be viewed in the Channel Box under SHAPES > object*n*_typeConstraint*n* or INPUTS > pairBlend*n*. |

### Blend Constraint*n*

The keyframe animation-constraint blend value determines the amount of influence that the constraint and keyframe animation have on the constrained object. If the value is 0, the keyframe animation has total control over the constrained object’s transformations. If the value is 1, the constraint has total control over the constrained object’s transformations.

If there are more than one of the same type of constraint applied to the object, *n* stands for the constraint number. This value appears in the pairBlend node’s attributes as *Weight*.

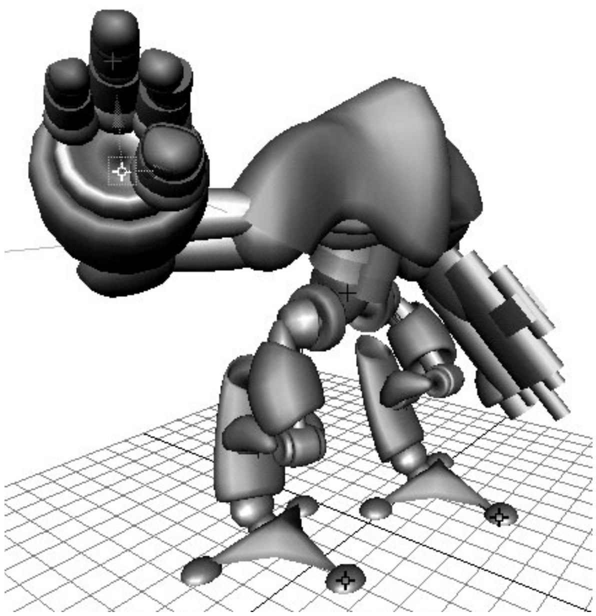
# 6 Character Sets

## About

### Character sets

### Character sets

In Maya, a character set is a node that brings together into a set all the attributes of any collection of objects that you want to animate together. The character set could be anything: a well-armed robot, an automobile, or even some seemingly unrelated collection of objects. Maya enables you to bring together all the attributes together in a character node, so you only have to select one node, the character node, when you want to animate all the various attributes.



Character created
by Jason Schleifer

For example, suppose you have created a snowman that consists of several NURBS spheres and cylinders parented into a hierarchy, along with several IK handles and some locators that include certain attribute controls (by adding new attributes from the Attribute Editor and using Animate > Set Driven Key > Set). By selecting all these entities that you want to animate together and creating a character node, you bring all the attributes of these entities together. You can name the character node "snowman," and when you select the "snowman" character node, you have immediate access to all of the snowman's attributes that you might want to animate. All the keyable attributes (channels) relevant for animating the snowman can be immediately accessible from the Channel Box, for instance. You can then set keys all on all the channels or on just some of them. Without the character node, you would have to select and animate the various objects that make up the snowman separately.

The power of character sets in Maya is that once you have created a character set, you can leverage Maya's animation features to animate at the character level. This enables Maya to provide the kind of intuitive interaction typically associated with traditional animation techniques. With character sets, you can also take advantage of Maya's nonlinear character animation feature, the Trax Editor. The Trax Editor takes character animation to a new, powerful level of artistic control and productivity (see *"Nonlinear animation tools in Maya"* in the Nonlinear Animation chapter of the Animation guide). Finally, you can build libraries of character sets that are all organized in a common manner.

In summary, Maya's character set feature brings together all of a character's attributes that are essential for its animation. By bringing these attributes together, you can set up a character that is much easier and faster to animate. Animators can take advantage of Maya's animation features to work on the character as a whole, and do not have to worry about the more technical details of a character's setup.

## Character node behavior

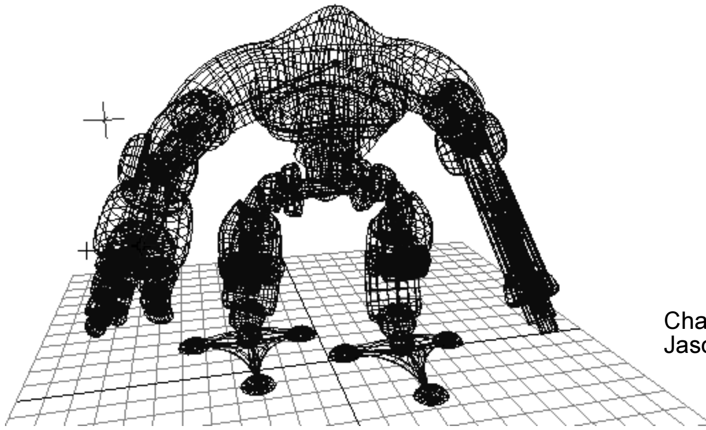
You don't need to know about character node behavior in order to use character sets effectively. If you are new to character sets, you can skip this section. However, familiarity with character node behavior can provide you with more control over character manipulation and performance.

For each object in your scene, if there has been any change to its node or any of the nodes in its history (its input or output nodes), Maya will evaluate the nodes and update the display based on the node's node behavior attributes. The node behavior attributes for character nodes can affect how characters are evaluated and displayed.

### Creating Character Set

## Creating Character Set

Defining character sets includes creating and editing character sets. Once you've defined your character set, you've completed character setup.



Character by
Jason Schleifer

During character setup, you create a complex, hierarchical organization of objects that provides the features of a distinct character. The character might be a character in the traditional sense (a robot, for example), or could be any collection of objects that make up something you want to animate as a distinct entity (a flying logo, for example).

You can bring together all the attributes of these objects you want to animate together by defining a character set for these objects. Defining a character set provides greater convenience during animation because all the attributes are available in the same place, and also because you can leverage Maya’s animation features to act on the character rather than on various separate objects.

You can also develop a hierarchy of character sets by creating subcharacter sets within a character set. With subcharacter sets, you can maintain a hierarchical relationship between a character’s various parts, while still providing character-level control over those various parts. A subcharacter is a subset of a character set.

Subcharacter sets are useful for keyframing and for creating animation clips with the Trax Editor. For example, you could define the attributes of a character’s right arm as a subcharacter set because you plan to do extensive keyframe animation of the right arm as compared to other body parts.

When a subcharacter set is current and you set a key, Maya keys only the subcharacter set’s attributes. When a character set is current and you set a key, Maya keys the character set’s attributes and all its subcharacter set’s attributes.

If you create a clip while a subcharacter set is current, the clip contains only the keyed subcharacter set's attributes. If you create a clip while the character is current, the clip contains all keyed attributes on the character and the subcharacter. The hierarchical relationship of a subcharacter set to its parent character set is displayed in the Outliner and in the Trax Editor.

When you define a collection of objects as a Maya character set (or subcharacter set), Maya creates a character node that brings together all the various attributes that you might want to animate. These attributes are placed in a type of set, called a character set, which is by default placed in Maya's character partition. (If you are not familiar with sets and partitions, refer to the *Basics* guide.) By default, Maya places all the keyable attributes of the objects into the character set. You can, however, edit the character set, adding any other attributes, or removing any attributes that you feel are not relevant to the animation of the character.

Defining a character set in Maya is the process of creating the character node, and then editing it so that you are then ready to animate it.

### Animating Character Sets

## Animating Character Sets

After you've defined character sets, you can leverage Maya's animation features to animate them at the character level.

Image by Cristoph Berndt

Defining character sets enables you to animate characters as a single entity rather than as a group of separate objects. For your convenience, all the attributes relevant for the character's animation can be available together in one place. For example, animators can set keys and breakdowns on characters instead of on the various objects that make up a character. This enables Maya to provide animators with a more intuitive approach to animating characters. Further, you can set a character as the current character set, identifying it as the character you want to focus on and animate.

## Keyframing character sets

As with any animatable object in Maya, you can set keys and breakdowns on character sets. For more information on animation, including setting keys and breakdowns, see the *Animation* guide.

## Creating expressions for character sets

You can create expressions for character sets, or for the objects that make up a character set. Expressions provide an excellent way to incorporate automatic or overlapping, secondary actions into a character's behavior. For example, you could create an expression that acts on smooth skin influence objects behind a character's chest or belly, making the character seem to breathe. For more information on expressions, see the *Expressions* guide.

## Using motion capture for character sets

You can impart motion to your character sets by using motion capture data. For more information on capturing, filtering, and using motion capture data, see the Motion Capture chapter in the *Animation* guide.

### Redirecting character animation

## Motion redirection

Redirecting motion is the process by which you can modify a character's path or orientation at anytime during its animation. For example, by redirecting a character's motion, you can change where a character lands, as well as the direction the character is facing, at the end of a character's jump animation.

You can use the *redirection controls* to move and rotate a character, and key these new positions and orientations, at any point during its animation.

> **Note** You can use the redirection controls to redirect any type (keyframe or motion capture animation, expressions, dynamics, and so on) of motion in Maya.

## Related topics

❖ "Character > Redirect" on page 549

### Mapping animation between characters

## Character mapping

Use the Character Mapper to create a correspondence between a source and target character's nodes or attributes so that you can import and export or copy and paste animation clips between the mapped characters in the Trax Editor.

Mapping or *linking* one character's nodes and attributes to that of another character creates a characterMap node. This node stores the array of connections between the mapped nodes and attributes. The information stored on this node can only be accessed by Maya.

### Related topics

- "Character > Character Mapper" on page 544
- "Map one character to another" on page 528

#### Workflow summary

### Workflow summary

Using Maya's character set feature involves defining the character set and then animating it. Defining the character set includes creating the character set and editing its collection of attributes. Animating the character set includes setting the current character set, and then setting and editing keys. For more information on defining character sets, see "Creating Character Set" on page 514. For more information on animating character sets, see "Animating Character Sets" on page 516.

By setting up characters with Maya's new character set feature, you can provide a complete, ready-to-animate character whose essential attributes are all gathered together for the animator's convenience. Once you've done this, character setup is complete and animation can begin.

## How do I? Set Character Sets

### Create character sets

When creating character sets, you can first set creation options and then create a character set, or you can immediately create a character set with the currently set creation options.

#### To set creation options

1. If you also want to create a character set now, select all the objects whose attributes you want to use to animate the character.
2. Select Character > Create Character Set > ❒.

   The options window is displayed.
3. Click Create Character Set if you want to create a character set and close the window now. Click Apply to create and keep the window open. Click Close to close the window.

#### To create a character set

1. Select the objects whose attributes you want to use to animate the character set.
2. Select Character > Create Character Set.

   A character set is created. The character node (default name: character*n*) provides a set of the keyable attributes from all the selected objects. All the keyable attributes are now conveniently organized in one character set provided by the character node.

   You might find that you don’t need immediate access to all of the attributes in the character set. Removing some of the attributes from the character set can make the listing of channels in the Channel Box shorter and more manageable. For information on removing and adding attributes, see ”Edit character sets” on page 522.

#### To create a character set from the Relationship Editor

You can also create characters while using the Relationship Editor in Character Editing mode (select Window > Relationship Editors > Character Sets). Select the objects you want to include in the character set, and in the Relationship Editor, select Edit > Create Character Set.

## Create character sets within character sets

You can create character sets within character sets. When you create a character set, you can include character sets among the objects you select before selecting Character > Create Character Set. By creating character sets within character sets, you can create a hierarchy of character sets that you want to animate together. You could also think of the character sets within character sets as subcharacter sets, aspects of a character that you might want to animate separately in certain situations.

## Create subcharacter sets

You can create subcharacter sets within previously defined character sets. When you create a subcharacter set, Maya adds the subcharacter set to the current character set. This is useful because you can apply the power of Maya’s character animation features to parts of a character’s hierarchy.

Creating subcharacter sets is similar to creating character sets: you select a character’s objects that you want to define as a subcharacter set (for example, the objects that make up the character’s face) and then select Character > Create Subcharacter Set. The creation options for subcharacters are the same as the options for creating characters.

## Select character sets

### To select a character set

Select Character > Select Character Set Node > character*n*, where character*n* is the default name for a character set.

The character set is selected. (Note that the objects in the character set are not selected.)

### To select the objects in a character set

Select Character > Select Character Set Members > character*n*, where character*n* is the default name for a character set.

The objects that make up the character set are selected, but the character set itself is not selected.

### To select a character set from the Relationship Editor

If using the Relationship Editor in Character Editing mode, you can select a character set so that it is currently selected in the workspace. To do so, select the character set, and then select Edit > Select Character Set.

## Add channels to a character set

You can quickly add the channels (attributes) of any object to a character set. The object need not already be part of the character set.

### To add channels to the current character set

1. Make sure the character set you want to add the channels to is the current character set.
2. Select the objects some or all of whose channels you want to add to the character set.
3. In the Channel Box, select the channels you want to add to the character set.
4. Select Character > Add to Character Set.

   Maya adds the selected channels to the current character set.

### To add channels to a character set using drag-and-drop in the Outliner

You can move an attribute from one character set to another. You can similarly add to a character set any numeric attribute that's not currently in a character set.

1. In the Outliner, display the attribute and the destination character set.

2 Use the middle mouse button to drag and drop the attribute on the character set.

## Edit character sets

### Edit character sets

Editing a character set involves adding or removing attributes from the set. By default, a character set includes all the keyable attributes of the objects included in the character. Typically, you only want to work with some of these attributes. Depending on the number and complexity of the objects included in the character, keeping all of these attributes in the character set can result in a needlessly long list of channels in the Channel Box. Consequently, after you create a character set, you may find you want to remove some of the attributes from the character set. Of course, you can later add them back to the character set.

For general information about sets and partitions, see the *Basics* guide. Editing a character set involves using the Relationship Editor. For more information about the Relationship Editor, see the *Basics* guide.

### Edit character set node behavior

#### To set node behavior with Attribute Editor

1 Open the node's Attribute Editor.
2 In the Attribute Editor, open Node Behavior.
3 Click Caching on or off.
4 Select the Node State as Normal, HasNoEffect, or Blocking. (The Waiting-Normal, Waiting-HasNoEffect, and Waiting-Blocking states are for Maya's internal use.)
5 Close the Attribute Editor.

#### To set Node State channel with Channel Box

When editing constraint channels with the Channel Box, you can set the Node State to Normal, HasNoEffect, Blocking, Waiting-Normal, Waiting-HasNoEffect, or Waiting-Blocking.

### Remove channels from a character set

You can quickly remove the channels from a character set.

#### To remove channels

1. Make sure the character set you want to remove the channels from is the current character set.
2. In the Channel Box, select the channels you want to delete from the character set.
3. Select Character > Remove from Character Set.

Maya removes the selected channels from the current character set. (see "Create character sets" on page 519).

## Edit character set channels

Channels are the attributes displayed in the Channel Box. The Channel Box provides a convenient way to edit a character set's channels.

To edit all attributes, use the Attribute Editor (see "Edit character attributes" on page 523).

#### To edit channels with the Channel Box

1. Select a character set.
2. In the Channel Box, the character set's channels are listed by default.
3. Click on a channel name with the left mouse button.
4. In your scene, press the middle mouse button and move the mouse to the left or right. By moving the mouse, you interactively change the value of the selected channel. As you move the mouse, note that pressing the Ctrl key will give you finer control, and pressing the Shift key will give you less control.

## Edit character attributes

#### To edit attributes with the Attribute Editor

1. Select the character set.
2. Open the Attribute Editor by selecting Window > Attribute Editor (default shortcut: Ctrl+a). Note that you can also open the Attribute Editor by double-clicking on the character set icon in the Outliner.

## View objects in a character set

#### To view which objects are in a character set

1. Select Window > Relationship Editors > Character Sets, or if you already have the Relationship Editor open, select its Character Editing mode option.

The editor's left column (Character Sets) lists all the character sets in your scene.

2 Select a character set so that it is highlighted.

3 From the Relationship Editor, select Edit > Select Character Set Members.

The objects in the character set are now all selected in the workspace. This provides a quick way to select and view those objects. It's useful if you just want to check which objects are part of a particular character set.

## Edit character set attributes

### To remove attributes from the character set

1 Select Window > Relationship Editors > Character Sets, or if you already have the Relationship Editor open, select its Character Editing mode option.

The editor's left column (Character Sets) lists all the character sets in your scene.

2 Select a character set so that it is highlighted.

3 Click on the + icon next to the selected character set to list all the attributes in the character set.

4 Select the attributes you want to remove from the character set so that they are highlighted in yellow. Remember that you can select items next to each other by pressing the Shift key and left mouse button, and that you can select items not next to each other by pressing the Ctrl key and the left mouse button.

5 In the Relationship Editor, select Edit > Remove Highlighted Attributes.

The selected attributes are removed from the character set.

6 If you'd like to return to the workspace and pose or animate the character, with the character set still selected, select Edit > Select Character Set.

The Channel Box now lists the new collection of attributes in the character set.

### To add attributes to the character set

1 Select Window > Relationship Editors > Character Sets, or if you already have the Relationship Editor open, select its Character Editing mode option.

The editor's left column (Character Sets) lists all the character sets in your scene; the right column (Objects) lists all the objects in your scene.

2. In the left column (Character Sets), select the character set to which you want to add attributes so that it is highlighted.
3. In the right column (Objects), select the object whose attributes you want to add to the character set.
4. Expand the object so that its attributes are listed (click the + icons next to the object's name).

   The attributes currently in the selected character set are highlighted in yellow. Also, note that the names of these attributes are displayed in italics.
5. Click on the names of the attributes you want to add to the character set.

   The selected attributes are added to the character set.
6. If you'd like to return to the workspace and pose or animate the character, with the character set still selected, select Edit > Select Character Set.

   The Channel Box now lists the new collection of attributes in the character set.

## View and edit the character partition

By default, each character set you create is placed in a default character partition. With all the character sets in the same partition, you can be sure that the attributes in one character set will not be in some other character set. To view the default character partition, you can use the Relationship Editor.

You should avoid editing the character partition. Editing the character partition can lead to problems because in doing so you might unintentionally end up having attributes in more than one character set.

**To view the character partition:**

1. If you already have the Relationship Editor open to edit character sets, change the Character Editing selection to Partition Editing. If you don't have the Relationship Editor open, select Window > Relationship Editors > Partitions.
2. In the Relationship Editor, note the character partition (default name: characterPartition). To find out the character sets in the character partition, click on the + icon. To find out all the attributes in a character set, click on the + icon next to the character set's icon.

For more information about using the Relationship Editor, refer to the *Basics* guide.

## Merge character sets

You can merge multiple character sets into a single character set without losing any clip data.

#### To merge multiple character sets into a single character set

1 Select the character sets.

  If you select a character set and one or more of its subcharacter sets, the selected subcharacter sets are merged into the top level character set. If none of the selected character sets are hierarchically related, the character sets are merged into the last selected character set.

2 Select Character > Merge Character Sets.

## Delete character sets

#### To delete a character set

1 Select the character set.

2 Select Edit > Delete (default shortcut: Backspace (IRIX, Linux, and Windows) or Delete (Mac OS X) key).

  The character node is deleted. However, the objects that made up the character are not deleted.

  Note that if you select all of the nodes that are in a character and delete them, the character node will also be deleted.

#### To delete a character set with the Relationship Editor

You can also delete the character set while using the Relationship Editor in Character Editing mode (select Window > Relationship Editors > Character Sets). In the Relationship Editor, select a character set, and then select Edit > Delete Highlighted Character Sets. Note that the Edit > Undo selection from Maya's main menu applies to actions you perform in the Relationship Editor.

## Set the current character Set

As you animate, you can set a character set as the current character set so that you can animate that character set only. Similar to layers, the current character set is separate from items in the selection list. This allows you to select items in your scene without changing which character set is the current character set. You can set the current character set from the Time Slider, the Character menu, or the Relationship editor.

#### To set the current character set from Time Slider:

You can quickly set any character set as the current character set as you animate. In the Time Slider, near the Timeline, note the black triangle button and the text field next to the Auto Keyframe icon. The text field displays the current character set's name. To set another character set as the current character set, click on the triangle icon. Select the character set you want as the current character set, or access the Relationship Editor by selecting Other.

#### To set the current character set from Character menu

You can set the current character set by selecting Character > Set Current Character Set. Select the character set you want as the current character set, or access the Relationship Editor by selecting Other. You can select character sets by selecting Character > Select Character Set Node > character*n* (the default name of a character).

#### To check and set current character sets from Relationship Editor

When using the Relationship Editor in Character Editing mode, you can check which character sets are current by selecting Edit > Highlight Current Character Sets. To make character sets you've selected in the editor as current character sets, select Edit > Make Highlighted Character Sets Current.

For more information about selecting and editing character sets with the Relationship Editor, see "Select character sets" on page 521, and "Edit character sets" on page 522. For more information about using the Relationship Editor, refer to the *Basics* guide.

## Map one character to another

## Workflow for mapping characters

- Open the Character Mapper window.
- Select the character from which you want to copy clips.

  This character is the source character.
- In the Character Mapper, click Load Source.

  All the nodes or attributes for the source character appear in the lefthand Unmapped column.

The state of the menu items *Display Node Names* and *Display Channel Names* determines whether node or attribute names appear in the Unmapped and Mapped columns. See ”Display Node Names” on page 545 and ”Display Channel Names” on page 546.

- Select the character to which you want to paste clips.

  This character is the target character.
- In the Character Mapper, click Load Target.

  All the nodes and attributes for the target character appears in the righthand Unmapped column.
- Select pairs of corresponding nodes or attributes from the left and righthand Unmapped columns, and click *Map*.

  The mapped pairs of nodes or attributes appear in the Mapped column.

Creating these pairs of mapped nodes or attributes generates a *character map* between the source and target characters. See *characterMap*.

## Map one character to another

For more information on mapping character animation, see Character > Character Mapper.

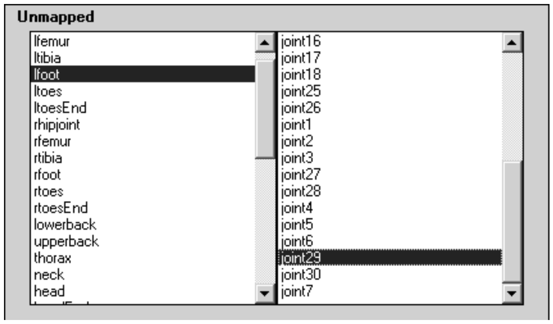
#### To map animation from one character to another

1. Select Character > Character Mapper.

   The Character Mapper window appears.
2. Select the root joint of your *source* character.
3. In the Character Mapper window, click Load Source.

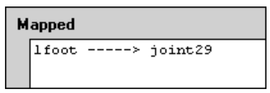
   The source character’s name appears in the *From:* field and all the names of its nodes or attributes appear in the left Unmapped column.
4. Select the root joint of your *target* character.
5. Click Load Target.

   The target character’s name appears in the *To:* field and all the names of its nodes or attributes appear in the right Unmapped column.
6. Select one node or attribute from the left unmapped column, and select its corresponding node or attribute in the right column. For example, if the left ankle joint for your source character is *lfoot* and the left ankle joint of your target character is named *joint29*, then your selection would resemble the following:




**7** Click Map.

The nodes or attributes that you selected in the Unmapped columns now appear as a pair of nodes or attributes in the Mapped column.




**8** Repeat steps 6 and 7 until you have mapped all the nodes or attributes from the source and target characters.

Now you can copy and paste animation clips between the mapped characters in the Trax Editor.

### To map animation from one character to another based on hierarchy

**1** Select the root joint of your source character.

**2** Shift-click to select the root joint of your target character.

**3** Select Character > Character Mapper.

The Character Mapper window appears with the source and target characters already loaded.

The name of the first character you selected appears in the From field and all its nodes or attributes appear in the lefthand Unmapped column. The name of the second character you selected appears in the To field and all its nodes or attributes appear in the righthand Unmapped column.

**4** Do one of the following:

- In the Character Mapper, select Edit > Map based on hierarchy.

  The character mapper searches for the source and target character's root joints, maps the roots, and then continues down the hierarchy mapping each of the character's nodes or attributes.

- Select a node or attribute from both the Source and Target character.

- In the Character Mapper, select Edit > Map based on hierarchy.

  The character mapper finds the selected nodes or attributes in the Source and Target character's hierarchies, maps those nodes or attributes, and from there continues down the hierarchy mapping each node or attribute. For example, if the you selected the Source and Target character's *left_hip* joints, then the character mapper would automatically map those nodes and then would continue down the hierarchy (left_knee, left_ankle, and so on) mapping each node.

## Redirect the motion of an animated character

### Workflow for redirecting animation

1. Step forward in your animation to where you want your character's redirection to occur.
2. Select the character (if you are redirecting an entire character, this would be its root joint) or the part of a character (if you are redirecting the motion of a character's back, this would be one of its spine joints) you want to redirect.
3. Create the desired (translation, rotation, or translation and rotation) redirection controls. See "To redirect an object's animation" on page 540.
4. Reposition your redirection controls.
   - For the rotation redirection control, move it to the point around which you want your character to rotate. For example, if you want to use the rotation redirection control to make a character go around a corner during its walk cycle, you would move the rotation redirection control to the ball of the foot the character pivots on when rounding the corner.
   - For the translation redirect control, move it to where you want the character or the part of a character to move to.
5. Step backward in your character's animation to where you want the character's redirection to begin.
6. Key the appropriate channels for your controls.

   You have to first key your controls at their base values. Base values are the values of the control's Rotate and Translate channels at the time of the control's creation.
7. Step forward in your character's animation to where in the Trax timeline you inserted the redirection controls.
8. Manipulate your redirection controls.

- Rotate your rotation redirection control.
- Translate your translation redirection control.

**9** Key the Rotate or Translate channels for your redirection controls once again.

This forms the redirection animation that redirects the character's original animation.

## Related topics

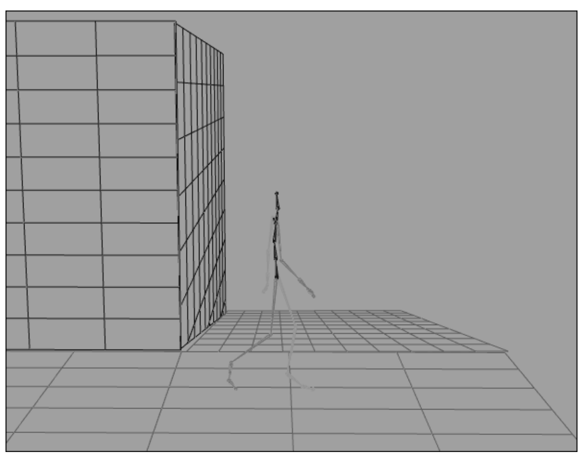
❖ "Animation redirection example" on page 531
❖ "To map animation from one character to another" on page 528
❖ "To map animation from one character to another based on hierarchy" on page 529

# Animation redirection example

*Scenario*
You want to modify the walk cycle animation of a biped character so that during its walk cycle the character turns a corner and then walks down a ramp.

### To make a biped character turn a corner during its walk cycle

**1** Step forward in the character's animation until you locate the point at which you want the character to turn around the corner, then stop the animation at a place in the walk cycle where the left foot of the biped character is on the ground.



**2** Select the root joint of the character's skeleton.

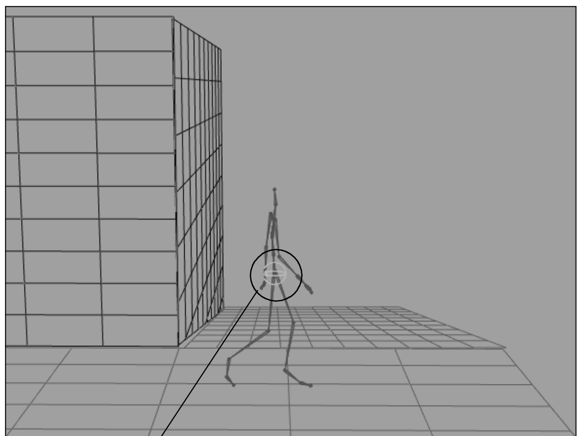
When redirecting the motion of a character, you want to create redirection controls for the parts of the character whose translation or rotation you want to change during its animation. In this scenario, the root joint is selected because we want to modify the orientation of the character's entire body during its walk cycle.

**3** In the Animation menu set, select Character > Redirect > ❐.

The Redirection Options window appears. See "Character > Redirect" on page 549.

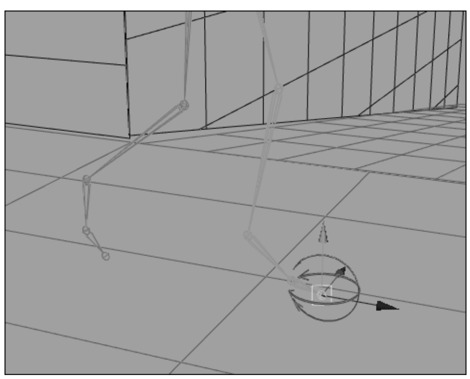
**4** Turn on Rotation Only and click Redirect.

The rotation redirection control appears at the root joint of the character.



Rotation redirection control

**5** Select the rotation redirection control and using the Translate Tool, move the control to the tip of your character's left foot.

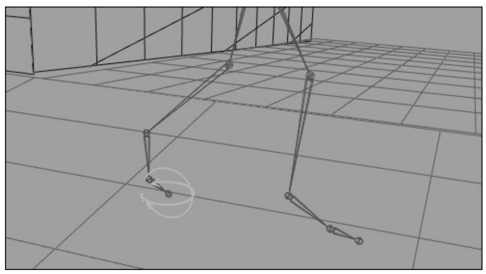


Moving the rotation redirection control lets you reposition the point around which the biped character pivots. In this scenario, you want the character to pivot on the ball of its left foot when it turns the corner.

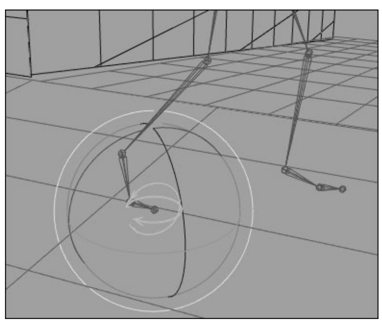
**6** In the Channel Box, key the Rotate X, Y, and Z channels for the left foot's rotation redirection control.

**7** Step forward in the animation until the right foot is on the ground and the left foot's heel is raised.

This is where the character will begin its turn.



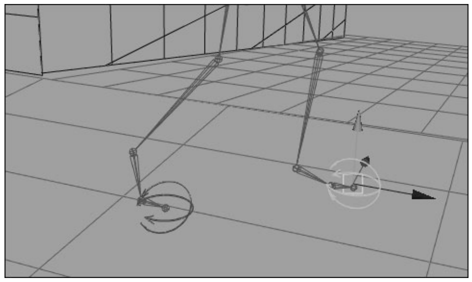
**8** Select the left foot rotation redirection control, rotate it 45 degrees, and key its Rotate X, Y, and Z channels.



**9** Select the character's root joint and select Character > Redirect.

A rotation redirection control appears at the character's root.

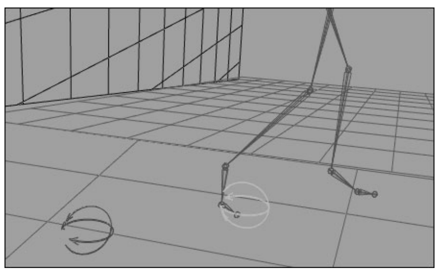
**10** Select the rotation redirection control and using the Translate Tool, move the control to the tip of your biped's right foot.



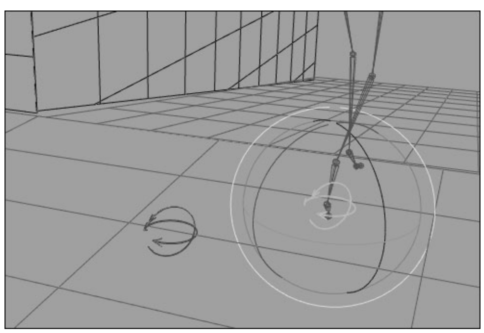
**11** In the Channel Box, key the Rotate X, Y, and Z channels for the right foot's rotation redirection control.

**12** Step forward in the animation until the left foot is once again on the ground and the right foot's heel is raised.

This is the point where the character will begin its descent down the ramp.

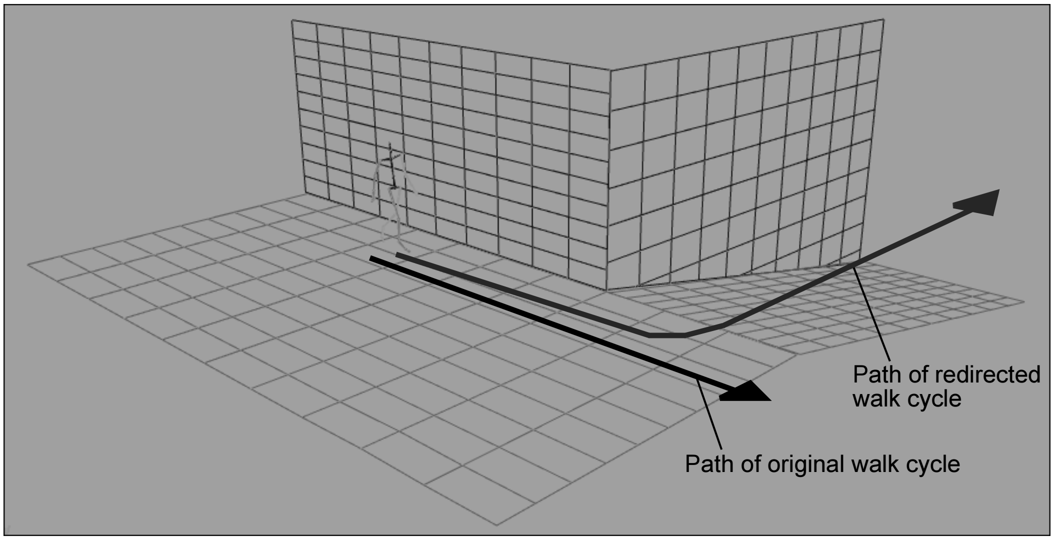


**13** Select the right foot's rotation redirection control, rotate it 45 degrees, and key its Rotate X, Y, and Z channels.



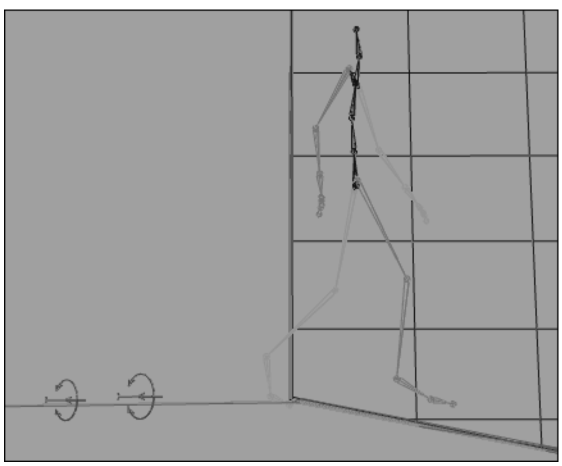
**14** Go to the beginning of the animation's playback range and play the animation.

The character now walks and turns left around the corner.




**To make a biped character walk down a ramp during its walk cycle**

1 Step forward in the animation until you reach the frame where the character is just about to walk down the ramp.



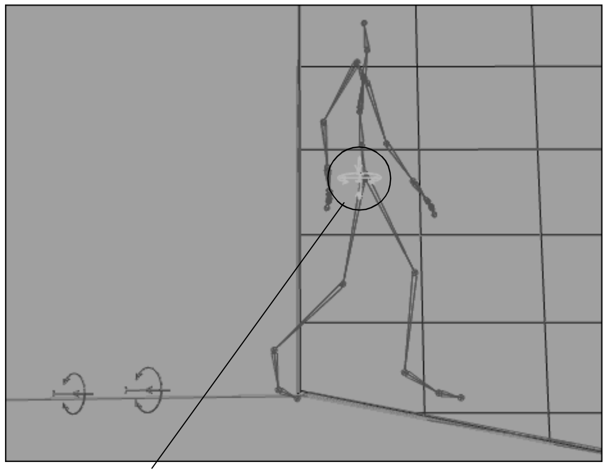
2 Select the root joint of the character.

3 In the Animation menu set, select Character > Redirect > ❐.

The Redirection Options window appears. See "Character > Redirect" on page 549.

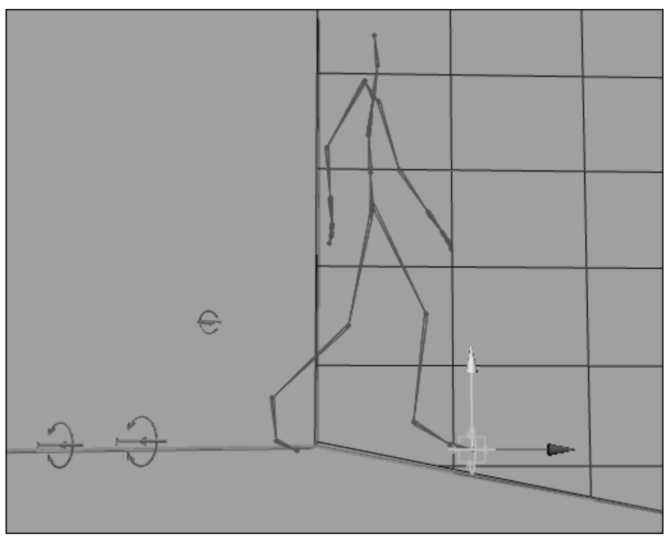
4 Turn on Rotation and Translation and click Redirect.

The rotation and translation redirection controls appear at the root joint of the character.

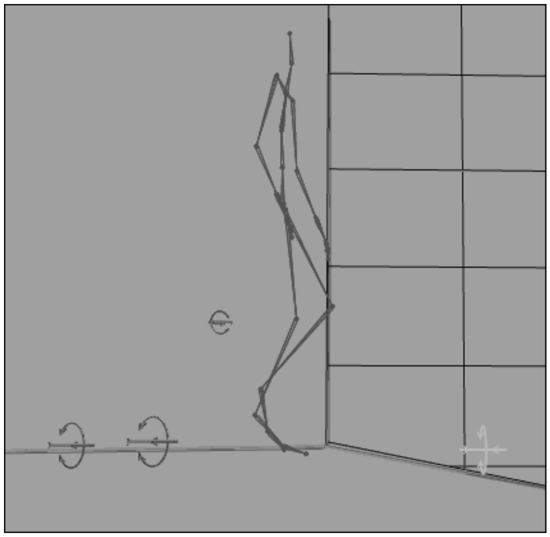


Rotation redirection control

5 Select the rotation and translation redirection controls and using the Translate Tool, move the controls to the tip of your biped’s right foot.



6 Step back a few frames until the character appears to be standing on the edge of the ramp with both feet side-by-side.

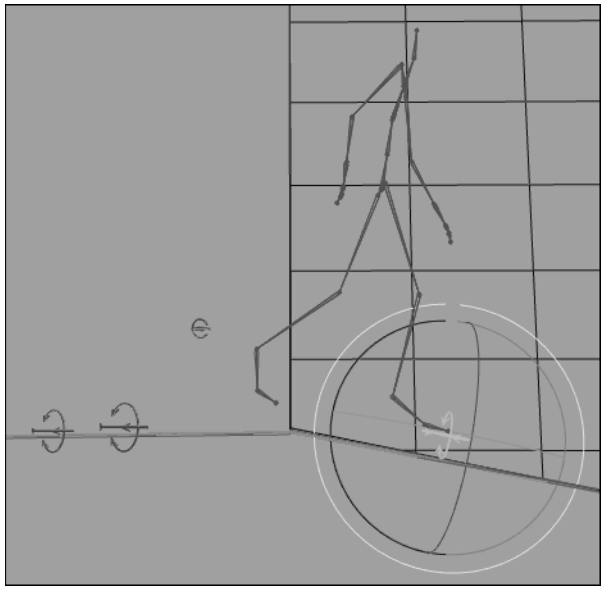


**7** Do the following:

- For the rotation redirection control, key all its Rotate X, Y, and Z channels.
- For the translation redirection control, key all its Translate X, Y, and Z channels.

**8** Step forward in the animation to the point where you inserted the redirection controls.

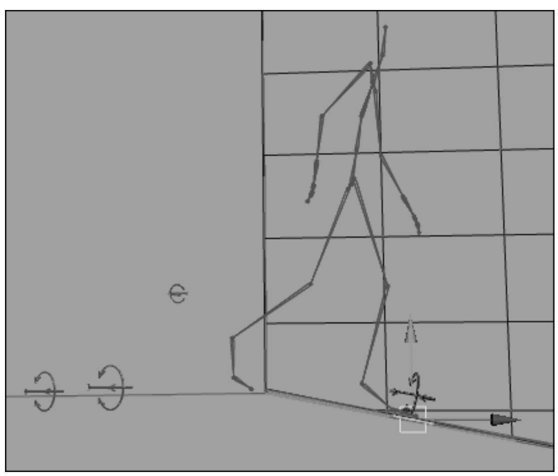
**9** Select the rotation redirection control and using the Rotate Tool, rotate the control until the character's feet are parallel to the ramp.



Note that there is a gap between your character's feet and the floor/ramp.

**10** In the Channel Box, key the rotation redirection control's Rotate X, Y, and Z channels.

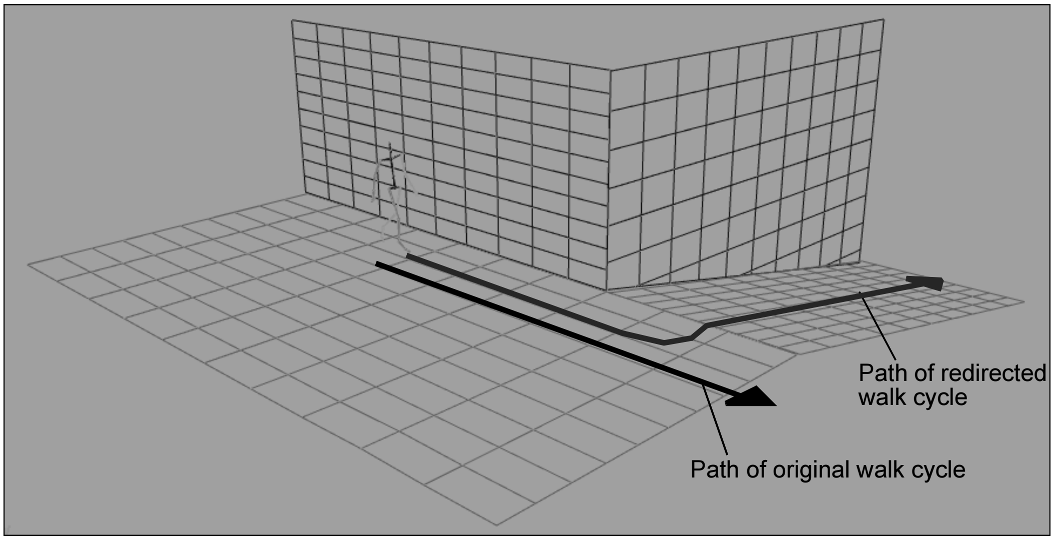
**11** Select the translation redirection control and using the Move Tool, translate your character downward until the right foot touches the ramp and the left foot touches the floor.



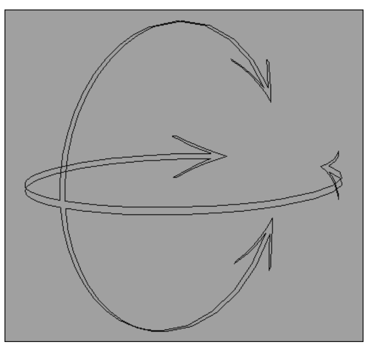
**12** In the Channel Box, key the translation redirection control's Translate X, Y, and Z channels.

**13** Go to the start of the animation's playback range and play your animation.

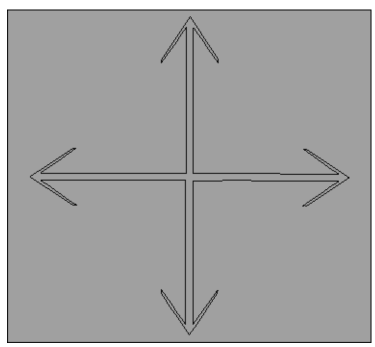
After the character turns the corner, it now walks down the ramp.




## Redirect the motion of an animated character



Rotation redirection control



Translation redirection control

#### To redirect a character set's animation

1. Step forward in your animation to where you want your character's redirection to occur.
2. Select the character set whose animation you want to redirect.
3. Select Character > Create Character Set > ❐.

   The Create Character Set Options window appears. See Character > Create Character Set in the Character Setup guide.

4 Set the options in the Character and Attributes sections as desired.
5 In the Redirection section, turn on Redirect Character.
6 Do the following:
   - If you want to redirect the rotation of your character, turn on Rotation Only.
   - If you want to redirect the translation of your character, turn on Translation Only.
   - If you want to redirect the rotation and translation of your character, turn on Rotation and Translation.
7 Click Apply.

   Rotation and/or translation redirection controls appear at the root of the current character set. The Create Character Set Options window remains open.
8 Do the following:
   - Move the rotation redirection control to change the position around which your character pivots.
   - Move the translation redirection control to change the current character's position relative to the position around which your character pivots.
9 Do steps 5 to 9 in the redirection workflow. See "Workflow for redirecting animation" on page 530.

### To redirect an object's animation

1 Step forward in your animation to where you want your object's redirection to occur.
2 Select the root of the object whose motion you want to redirect.
3 Select Character > Redirect > ❐.

   The Character Redirection Options window appears.
4 In the Redirection Type section, do the following:
   - If you want to redirect only the rotation of your character, turn on Rotation Only.
   - If you want to redirect only the translation of your character, turn on Translation Only.
   - If you want to redirect both the rotation and translation of your character, turn on Rotation and Translation.
5 Do one of the following:
   - Click Redirect. The rotation and/or translation redirection controls appear at the root of the object and the Character Redirection Options window closes.

- Click Apply. The rotation and/or translation redirection controls appear at the root of the object and the Character Redirection Options window remains open.

**6** Do the following:

- Move the rotation redirection control to change the position around which your character pivots.
- Move the translation redirection control to change the current character's position relative to the position around which your character pivots.

**7** Do steps 5 to 9 in the redirection workflow. See "Workflow for redirecting animation" on page 530.

# Reference

# Menus

## Animation menu set

### Character >

## Character > Create Character Set

## Character > Create Character Set > ❒

Name

Specifies the name of the character set. Default name is character*n*.

Character Set Attributes

Specifies which attributes will be included with the character set as keyable. You can choose All Keyable, From Channel Box, or All Keyable Except.

All Keyable

Specifies that all the keyable attributes of all selected objects will be included as the character set's attributes.

From Channel Box

Specifies that only the currently selected channels in the Channel Box will be included with the character set.

### All Keyable except

Specifies whether certain attributes will be included with the character set. This allows you to control the number of attributes included in a character set as you create the character set. This can save you time, reduce the number of attributes listed in the Channel Box, and help make your animation work more efficient.

| | |
|---|---|
| No Translate | Specifies that the translation attributes will be included as keyable attributes unless checked on (default is off). |
| No Rotate | Specifies that the rotation attributes will be included as keyable attributes unless checked on (default is off). |
| No Scale | Specifies that the scaling attributes will be excluded as keyable attributes unless checked off (default is on). |
| No Visibility | Specifies that the visibility attribute will be excluded as a keyable attribute unless checked off (default is on). |
| No Dynamic | Specifies that any dynamic attributes will be excluded as keyable attributes unless checked off (default is off). |

### Hierarchy

Specifies that all of the objects in the hierarchy below the selected objects are included in the character set. When turned off, only the selected objects are included in the character set.

### Redirect Character

Makes the current character set redirectable. When a character set is redirectable, this mean that you can now change the translation and orientation of already established (for example, motion capture) animation.

| | |
|---|---|
| Rotation and Translation | Creates a rotation and translation control for the current character set. |
| Rotation Only | Creates a rotation redirection control for the current character set. The rotation redirection control appears at the origin of the current character.<br><br>The rotation redirection control is useful if you want to change the orientation of your character set's pivot. For example, you can manipulate a rotation |

redirection control to get a character to turn 90 degrees (around a corner perhaps) halfway through its walk cycle.

Translation Only
: Creates a translation redirection control for the current character set. The rotation redirection control appears at the origin of the current character.

The translation redirection control is useful if you want to change the translation of the point around which your object pivots. For example, you can manipulate the translation redirection control to change the place at which a character lands from a jump.

# Character > Create Subcharacter Set

## Character > Create Subcharacter Set > ❐

### Name

Specifies the name of the subcharacter set. Default name is subCharacter*n*.

### Subcharacter Set Attributes

Specifies which attributes will be included with the subcharacter set as keyable. You can choose All Keyable, From Channel Box, or All Keyable Except.

All Keyable
: Specifies that all the keyable attributes of all selected objects will be included as the character set's attributes.

From Channel Box
: Specifies that only the currently selected channels in the Channel Box will be included with the character set.

All Keyable except
: Specifies whether certain attributes will be included with the character set. This allows you to control the number of attributes included in a character set as you create the character set. This can save you time, reduce the number of attributes listed in the Channel Box, and help make your animation work more efficient.

No Translate

Specifies that the translation attributes will be included as keyable attributes unless checked on (default is off).

No Rotate

Specifies that the rotation attributes will be included as keyable attributes unless checked on (default is off).

No Scale

Specifies that the scaling attributes will be excluded as keyable attributes unless checked off (default is on).

No Visibility

Specifies that the visibility attribute will be excluded as a keyable attribute unless checked off (default is on).

No Dynamic

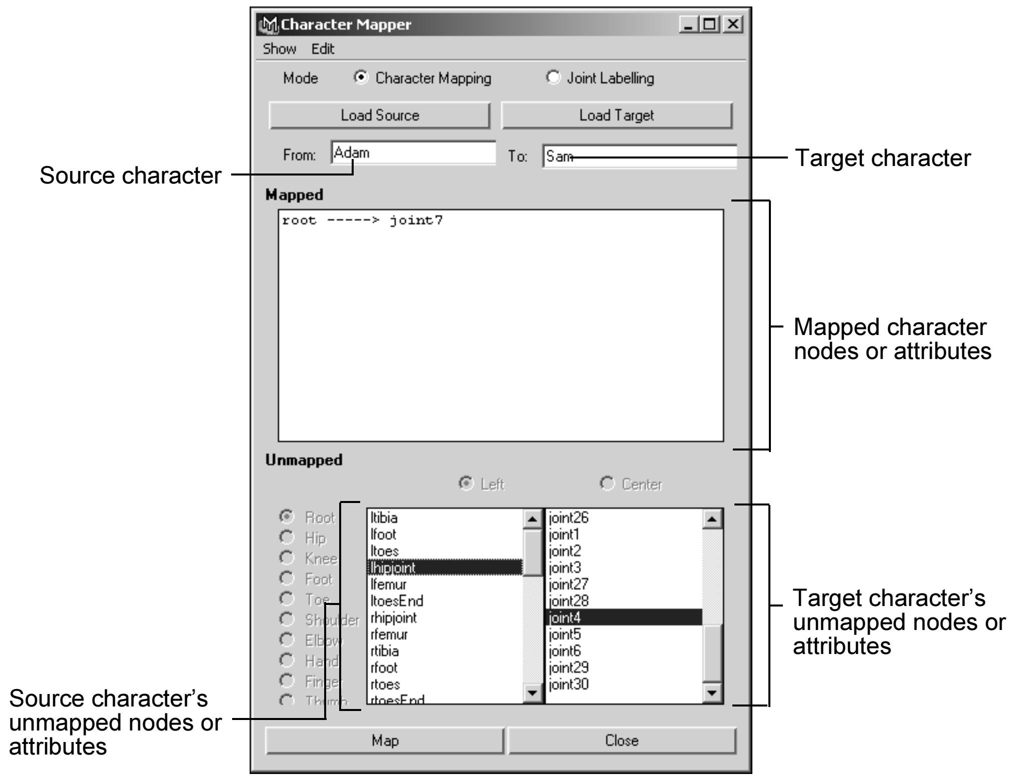
Specifies that any dynamic attributes will be excluded as keyable attributes unless checked off (default is on).

## Character > Character Mapper

The new Character Mapper window lets you map animation from one character (*source*) to another (*target*) character. Use the Character Mapper to establish a relationship between a source and target character's nodes or attributes so that you can import and export or copy and paste animation clips between the mapped characters in the Trax Editor.

There are four different ways you can view a character's mapped and unmapped nodes or attributes in the Character Mapper: Hierarchical, Alphabetical, Node, and Channel views. See "Show menu" on page 545. These views can help you better navigate long node or attribute lists (hierarchical or alphabetical) and isolate specific nodes (Node view) or animation channels (Channel view).

The Character Mapper is most useful when you want to map animation from one similar character to another. For example, mapping the animation from the translateX channel of one biped character to the translateZ channel of another biped character.




## Show menu

### Alphabetical

Lists the nodes or attributes in the Mapped and Unmapped columns alphabetically, starting with A.

### Hierarchical

Lists the nodes or attributes in the Mapped and Unmapped columns by their hierarchical relationships. This means that the nodes or attributes are displayed in parent-child order starting from the root. This option is on by default.

### Display Node Names

Displays all the character set's nodes in the Mapped and Unmapped columns. This option lets you map animation from source to target character nodes. The node view format is as follows:

```
SourceJoint  ----->  TargetJoint
```

### Display Channel Names

Displays all the character set's keyable channels in the Mapped and Unmapped columns. Use this option if you need to map animation from certain source character channels to specific target character channels. For example, use this option if you want to map the animation on the X rotation channel for the source character's left shoulder to the Z rotation channel of the target character's left shoulder. The channel view format is as follows:

```
SourceNode.SourceChannel  -----> TargetNode.TargetChannel
```

## Edit Menu

### Unmap

Breaks the link between the node or attribute pair selected in the Mapped column.

### Unmap All

Breaks all the links between the nodes or attributes of the source and target characters that are currently loaded in the Character Mapper. These links or relationships between the nodes or attributes of the source and target characters are displayed in the Mapped column.

### Clear All

Clears the current contents of the Unmapped and Mapped columns.

### Refresh

Refreshes the Mapped and Unmapped columns to display the most recent (current) version of their contents.

### Map based on node names

Maps the source and target characters by their node or attribute names. Select this menu item when both your source and target character's node or attribute names are the same. For example:

```
lshoulder -----> lshoulder
```

```
lelbow -----> lelbow
```

```
lhand -----> lhand
```

### Map based on hierarchy

Maps the source and target characters by their hierarchical structures. Select this menu item when you want to automatically map the various node hierarchies in your character.

There are two ways you can map your animation based on the hierarchies of the Source and Target characters:

- You can automatically map all the nodes and attributes of the source and target characters. If no nodes or attributes are selected in the Unmapped columns, the Character Mapper will search for the *root* of the Source and Target characters and will then traverse and map their hierarchies accordingly.
- You can automatically map the nodes or attributes of a specified segment of the character's hierarchy (for example, limb to limb). You specify which segment you want to map by selecting a single Source and a single Target node or attribute at the top of the segment's hierarchy. The Character Mapper will then map down the segment of each character's hierarchy based on the nodes or attributes you selected.

  For example, if you selected the *left_shoulder* joint of each character, then the Character Mapper will map the Source and Target *left_shoulder* nodes, and will continue down the hierarchy mapping the *left_elbow* nodes, the *left_forearm* nodes, and so on.

See *"To map animation from one character to another based on hierarchy"* in the Animation chapter of the *What's New in Maya 6.0* guide.

Map based on order

Maps the source and target characters by the hierarchical order (descending) of their nodes or attributes. Select this menu item when both your source and target character's nodes or attributes have the same basic hierarchy. For example:

```
joint3 -----> 3joint

joint2 -----> 2joint

joint1 -----> 1joint
```

## Fields

From:

Displays the name of the *source* character. The source character set contains the animation data that is remapped to the target character set's skeleton.

To:

Displays the name of the *target* character. The target character set's skeleton receives the remapped animation data from the source character set.

## Buttons

### Load Left button

Loads all the node or attribute names from the selected character into the Left (source) column.

### Load Right button

Loads all the node or attribute names from the selected character into the Right (target) column.

### Map button

Creates a correspondence between the animation from the selected *source* node or attribute (left Unmapped column) to the *target* node or attribute (right Unmapped column). For example, if you want to map node1 from the source character to node2 of the target character, the node pair or relationship would appear in the Mapped column as the following:

```
node1 -----> node2
```

### Close button

Closes the Character Mapper window.

## Columns

### Mapped column

Displays all the node or attribute relationships between the source and target character sets. Once you link nodes or attributes by clicking on them in the source (left) and target (right) Unmapped columns, the linked pairs appear in the Mapped column.

### Unmapped columns

The left unmapped column lists all the nodes or attributes from the *source* character set's skeleton, and the right unmapped column lists all the nodes or attributes from the *target* character set's skeleton. To link pairs of nodes or attributes, see *"How do I map one character to another?"* in the Animation chapter of the *What's New in Maya 6.0* guide.

# Character > Add to Character Set

Adds the selected attribute to the current character or subcharacter set.

# Character > Remove from Character Set

Removes the selected attribute from the current character or subcharacter set.

# Character > Merge Character Sets

Combines or *merges* the selected character or subcharacter sets.

# Character > Redirect

Adds a translation and/or rotation redirection controls to the current character. The controls appear at the origin of the character.

When a character is redirectable, you can change the translation and orientation of its already established (for example, motion capture) animation.

## Character > Redirect > ❐

When you select this menu item, the Character Redirection Options window appears.

### Redirection Type

| | |
|---|---|
| Rotation and Translation | Creates a rotation and translation control for the current object. |
| Rotation Only | Creates a rotation redirection control for the current character. The rotation redirection control appears at the origin of the current object.<br><br>The rotation redirection control is useful if you want to change the orientation of the point around which your object pivots. For example, you can manipulate a rotation redirection control to get a character to turn 90 degrees (around a corner perhaps) halfway through its walk cycle. |
| Translation Only | Creates a translation redirection control for the current character set. The translation redirection control appears at the origin of the current object.<br><br>The translation redirection control is useful if you want to change the translation of the point around which your object pivots. For example, you can |

manipulate the translation redirect controls to change the place at which a character lands from a jump.

# Nodes

## Character nodes

### General character set node attributes

#### Node Behavior

See "*Node behavior*" in the *Basics* guide.

### character

See character node in Help > Node and Attribute Reference.

#### Character Set Attributes

Lists all of the attributes in the character set. To add object attributes to the character set, see "To add channels to the current character set" on page 521. To edit which attributes are in the character set, see "Edit character sets" on page 522.

### characterMap

Stores the data (character map) that forms a correspondence between the the nodes or attributes of multiple characters.

See characterMap node in Help > Node and Attribute Reference.

#### Related topics

- "Character mapping" on page 518
- "Map one character to another" on page 528

### characterOffset

This node redirects most types of motion in Maya. This includes: keyframe animation, motion capture animation, expressions (unsupported), and dynamics (unsupported).

See characterOffset node in Help > Node and Attribute Reference.

## Related topics


- ”Motion redirection” on page 518
- ”Redirect the motion of an animated character” on page 539

# Index

## A



## B

## D

## E

## F



## G

## N



## O

## P

## R

## T

## X



## Y