Bones System

A Bones system is a jointed, hierarchical linkage of bone objects that can be used to animate other objects or hierarchies.

- Create panel > (Systems) > Standard > Object Type rollout > Bones button
- Standard menu: Animation menu > Bone Tools > Create Bones
- Enhanced menu: Objects menu > Characters and Bones > Bone Chain

Bones are especially useful for animating character models that have a continuous skin mesh. You can animate bones with forward or inverse kinematics. For inverse kinematics, bones can use any of the available IK solvers, or through Interactive or applied IK.

Bones are renderable objects. They have several parameters, such as taper and fins, that can be used to define the shape the bone represents. The fins make it easier to see how the bone is rotating.

For animation, it is very important that you understand the structure of a bone object. The bone's geometry is distinct from its link. Each link has a pivot point at its base. The bone can rotate about this pivot point. When you move a child bone, you are really rotating its parent bone.

It might be useful to think of bones as joints, because it is their pivot placements that matter, more than the actual bone geometry. Think of the geometry as a visual aid that is drawn lengthwise from the pivot point to the bone's child object. The child object is usually another bone.

Bones system seen alone and inside a wireframe model

Any hierarchy can display itself as a bone structure (see Using Objects as Bones), by simply turning on Bone On in the Bone Editing Tools rollout.

Creating Bones

You start creating bones by clicking the Create Bones button on the Bone Editing Tools rollout, or by clicking the Bones button in the Systems category on the Create panel.
To create bones, do the following.

1. Your first click in a viewport defines the start joint of the first bone.
2. The second click in a viewport defines the start joint of the next bone. Visually only one bone is drawn at this point because bones are visual aids drawn between two pivot points. It is the actual pivot point’s placement that is important.
3. Each subsequent click defines a new bone as a child of the previous bone. The result of multiple clicks is a single chain of bones.
4. Right-click to exit bone creation. This creates a small “nub” bone at the end of the hierarchy, which is used when assigning an IK chain. If you are not going to assign an IK chain to the hierarchy, you can delete the small nub bone.

Creating a simple chain of three bones

3ds Max lets you create a branching hierarchy of bones. To create a branching hierarchy, such as legs branching from a pelvis, do the following:

1. Create a chain of bones, and then right-click to exit bone creation.
2. Click Bones (or Create Bones) again, and then click the bone where you want to begin branching. The new chain of bones branches from the bone you click.

Warning: The behavior of a branching bone hierarchy is not always intuitive.

3. Note: You can also use Select And Link to connect one bone hierarchy to its branches. However, except for this one special case, using Select And Link with bones is not recommended. To edit an existing bone structure, whether branching or not, use the Bone Tools instead.

Assigning IK Controllers to Bones

By default, bones are not assigned inverse kinematics (IK). Assigning an IK solver can be done in one of two ways. Typically, you create a bone hierarchy, then manually assign an IK solver. This allows for very precise control over where IK chains are defined.

The other way to assign an IK solver is more automatic. When you create bones, choose IK solver from the list in the IK Chain Assignment rollout, and then turn on Assign To Children. When you exit bone creation, the chosen IK solver is automatically applied to the hierarchy. The solver extends from the first bone in the hierarchy to the last.

For more information about IK, see Introduction to Inverse Kinematics.

Setting the Initial Position of Bones

When you first create a bones system, the position of the bones is the initial state. Before you assign an IK solver or method, you can change the initial state of the bones by moving, rotating, or stretching the bones individually.

Bone Color
By default, bones are assigned the color specified for Bones in the Colors panel of the Customize User Interface dialog. Choose Object as the Element and then choose Bones in the list. You can change the color of individual bones by selecting the bone, clicking the active color swatch next to the bone’s name in the Create panel or Modify panel, and then selecting a color in the Object Color dialog.

You can also use the Bone Tools to assign bone colors, or to assign a color gradient to a bone hierarchy.

Bone Fins
Fins are visual aids that help you clearly see a bone’s orientation. Fins can also be used to approximate a character’s shape. Bones have three sets of fins: side, front, and back. By default, fins are turned off.

Bones can have fins.

Bones with various fin configurations

Renderable Bones
Bones can be renderable, though by default, they are not. To make a bone renderable, turn on the Renderable checkbox in the bone’s Object Properties dialog.

Object Properties for Bones
In addition to visual properties, bones have behavioral properties. The controls for these are located on the Bone Tools.

You can use these controls to turn other kinds of objects into bones.

Using Constraints with Bones
You can apply constraints to bones as long as an IK solver or method is not controlling the bones. If the bones have an assigned IK controller, you can constrain only the root of the hierarchy or chain. However, applying position controllers or constraints to a linked bone can cause undesirable effects, such as breaking of the bone chain.

The “nub” bone at the end of the chain has a Spring controller applied to it. The Spring controller is connected to an animated sphere.
Right: The sphere's movement breaks the bone chain.

To avoid this problem, don't apply position controllers directly to child bones. Instead, create an IK chain and apply the controller to the IK chain's end effector.

An IK chain has been applied, connecting the end nub to its parent bone. The IK chain's end effector is connected to the ball by a Spring controller.

Right: Now when the sphere moves, the IK chain prevents the bones from breaking.

Constraints and controllers that affect orientation only, such as Orientation or Look At, do not present this problem when applied to child bones.

Procedures
To create a bones system:

1. On the Create panel, click (Systems). On the Object Type rollout, turn on Bones. You can also access Create Bones through the Bone Tools rollout.

2. Click in a viewport.
   This creates a joint that is the base of the bone's hierarchy.

3. Drag to define the length of the second bone.
4. Click to set the length of the second bone, and then drag to create the third bone. Drag and click to continue creating new bones.
5. Right-click to end creation.
   3ds Max creates a small "nub" bone at the end of the hierarchy. This bone is used when assigning an IK chain.

   The first bone you create is at the top of the hierarchy. The last bone you create is at the bottom. For more about linked objects, see the Hierarchy Panel.

To create a bones hierarchy with an IK solver automatically applied:

1. On the Create panel, click (Systems). On the Object Type rollout, turn on Bones.
2. In the IK Chain Assignment rollout, select an IK solver from the list.
3. Turn on Assign To Children.
4. In a viewport, click and drag to create the bones. Right-click to end bone creation. After the bones are created, the chosen IK solver is applied to them.

To edit the appearance of a bone:

1. Select a bone.
2. Go to the Modify panel.

To change the length of bones after they’ve been created:

Attention: Repositioning a bone affects its length visually. More important, it affects the bone’s pivot position. The length of the bone is only a visual aid drawn between each bone’s pivot point. A bone has only one pivot. The bone you see visually is connecting its pivot point to the next bone’s pivot point.

1. Choose Animation menu ➤ Bone Tools.
2. On the Bone Tools dialog, click Bone Edit Mode.
3. Move the child of the bone you want to change. The length of its immediate parent changes to reach the child bone.
4. Turn off Bone Edit Mode when you are finished editing the bones.

To add fins to bones:

1. Select the bone.
2. Choose Animation menu ➤ Bone Tools.
3. Select the bones to which you want to add fins.
4. In the Fin Adjustment Tools rollout, turn on Side Fins, Front Fin or Back Fin.
5. Adjust the size and appearance of the fins with the appropriate spinners.

Note: You can also add fins to an individual bone on the Modify panel.

Interface

IK Chain Assignment rollout (creation time only)

Provides the tools to quickly create a bone chain with an IK solver automatically applied. Also allows for bone creation with no IK solver.

IK Solver drop-down list

Specifies the type of IK solver to be automatically applied if Assign To Children is turned on.

Assign To Children

When on, assigns the IK solver named in the IK solver list to all the newly created bones except the first (root) bone. When off, assigns a standard PRS Transform controller to the bones. Default=off.
Note: Choosing the SplineIKSolver and turning on Assign To Children causes the Spline IK Solver dialog to appear after bones have been created.

**Assign To Root**

When on, assigns an IK solver to all the newly created bones including the first (root) bone.

Turning on Assign To Children also automatically turns on Assign To Root.

**Bone Parameters rollout (creation and modification time)**

<table>
<thead>
<tr>
<th>Bone Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bone Object</td>
</tr>
<tr>
<td>Width: 4.0</td>
</tr>
<tr>
<td>Height: 4.0</td>
</tr>
<tr>
<td>Taper: 90.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bone Fins group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Side Fins</td>
</tr>
<tr>
<td>Size: 2.0</td>
</tr>
<tr>
<td>Start Taper: 10.0</td>
</tr>
<tr>
<td>End Taper: 10.0</td>
</tr>
</tbody>
</table>

| Front Fin       |
| Size: 2.0       |
| Start Taper: 10.0|
| End Taper: 10.0 |

| Back Fin        |
| Size: 2.0       |
| Start Taper: 10.0|
| End Taper: 10.0 |

| Generate Mapping Coords |

These controls change the appearance of the bones.

**Bone Object group**

**Width**

Sets the width of the bone to be made.

**Height**

Sets the height of the bone to be made.

**Taper**

Adjusts the taper of the bone shape. A Taper of 0 produces a box-shaped bone.

**Bone Fins group**

**Side Fins**

Lets you add a set of fins to the sides of the bones you create.
- **Size** Controls the size of the fin.
- **Start Taper** Controls the start taper of the fin.
- **End Taper** Controls the end taper of the fin.

**Front Fin**

Lets you add a fin to the front of the bone you create.

- **Size** Controls the size of the fin.
- **Start Taper** Controls the start taper of the fin.
- **End Taper** Controls the end taper of the fin.

**Back Fin**

Lets you add a fin to the back of the bone you create.

- **Size** Controls the size of the fin.
- **Start Taper** Controls the start taper of the fin.
- **End Taper** Controls the end taper of the fin.

**Generate Mapping Coords**

Creates mapping coordinates on the bones. Since the bones are renderable, they can also have materials applied, which can use these mapping coordinates.

---

**IK Solvers**

Dec 17, 2014 | In-Product View

An IK solver creates an inverse kinematic solution to rotate and position links in a chain. It applies an IK Controller to govern the transforms of the children in a linkage. You can apply an IK solver to any hierarchy of objects. You apply an IK solver to a hierarchy or part of a hierarchy using commands on the Animation menu. Select an object in the hierarchy, choose an IK solver, and then click another object in the hierarchy to define the end of the IK chain.

- Standard menu: Select an object in a hierarchy where you want IK to start. > Animation menu > IK Solver > Apply an IK solver. > Click the object in the hierarchy where you want the IK chain to end.
- Enhanced menu: Select an object in a hierarchy where you want IK to start. > Animation menu > Relationships/IK > Apply an IK solver. > Click the object in the hierarchy where you want the IK chain to end.
Each type of IK solver has its own behavior and workflow, as well as its own specialized controls and tools that display in the Hierarchy and Motion panels. IK solvers are plug-ins, so programmers can expand IK capabilities of 3ds Max by customizing or writing their own IK solvers.

3ds Max ships with four different IK solvers. The two most commonly used are History Independent (HI) and History Dependent (HD). To see both working side by side, play the following video:

In the preceding video, robot arm 1 uses HI IK, while robot arm 2 uses HD IK. The end effectors for both are animated identically. Notice the different IK solutions the two algorithms use to reach the same points in space.

How Does an IK Solver Work?

An IK solver generally operates in this way: an inverse kinematic chain is defined on part of the hierarchy, say from the hip to the heel, or the shoulder to the wrist of a character. At the end of the IK chain is a gizmo, called the goal. The goal can be repositioned or animated over time in a variety of ways, often using linkage, parameter wiring or constraints. No matter how the goal is moved, the IK solver attempts to move the pivot of the last joint in the chain (also called the end effector) to meet the goal. The IK solver rotates the parts of the chain to stretch out and reposition the end effector to coincide with the goal.
Using an IK solver to animate an arm

Frequently, the end effector is constrained to the ground plane. For example, you might "pin" the toes as the heels lift in a character walk cycle. Then the movement of the root of the chain poses the legs up from the toes.

Four plug-in IK solvers ship with 3ds Max:

- **HI (History-Independent) Solver**
  The HI Solver is the preferred method for character animation, and for any IK animation in long sequences. With HI Solvers, you can set up multiple chains in a hierarchy. For example, a character's leg might have one chain from hip to ankle, and another from heel to toe.

  Because this solver's algorithm is history-independent, it is fast to use regardless of how many frames of animation are involved. Its speed is the same on frame 2000 as it is on frame 10. It is stable and jitter-free in viewports. This solver creates a goal and an end effector (although the display of end effector is off by default). It uses a swivel angle to adjust the solver plane to position the elbow or the knee. You can display the swivel angle manipulator as a handle in the viewport, and adjust it. HI IK also uses a preferred angle to define a direction for rotation, so the elbow or knees bend correctly.

- **HD (History-Dependent) Solver**
  The HD Solver is a solver well-suited to use for animating machines, especially ones with sliding parts that require IK animation. It lets you set up joint limits and precedence. It has performance problems on long sequences, so ideally use it on short animation sequences. It is good for animating machines, especially ones with sliding parts.

  Because this solver's algorithm is history dependent, it works best for short animation sequences. The later in the sequence it is solving, the longer it takes to calculate a solution. It allows you to bind the end effector to a follow object, and it uses a system of precedence and damping to define the joint parameters. It allows for sliding joint limits combined with IK animation, unlike the HI IK solver, which only allows for sliding joint limits when using FK movement.

- **IK Limb Solver**
  The IK Limb solver operates on only two bones in a chain. It is an analytical solver that is fast in viewports, and can animate the arms and legs of a character.
The IK Limb solver can be used for export to game engines.

Because this solver's algorithm is history-independent, it is fast to use regardless of how many frames of animation are involved. Its speed is the same on frame 2000 as it is on frame 10. It is stable and jitter-free in viewports. This solver creates a goal and an end effector (although the display of end effector is off by default). It uses a swivel angle to adjust the solver plane to position the elbow or the knee. You can target the swivel angle to another object to animate it. IK Limb Solver also uses a preferred angle to define a direction for rotation, so the elbow or knees bend correctly. It also allows you to switch between IK and FK by keyframing IK Enabling, and it has a special IK for FK pose function so you can use IK to set FK keys.

- Spline IK Solver

The Spline IK solver uses a spline to determine the curvature of a series of bones or other linked objects.

Vertices on a Spline IK spline are called nodes. Like vertices, the nodes can be moved and animated to change the curvature of the spline.

The number of spline nodes can be fewer than the number of bones. This provides the ease of posing or animating a long multiple-bone structure with just a few nodes, as opposed to animating each bone individually.

Spline IK provides a more flexible animation system than other IK solvers. Nodes can be moved anywhere in 3D space, so the linked structure can be intricately shaped.

A helper object is automatically placed at each node when Spline IK is assigned. Each node is linked to its corresponding helper, so a node can be moved by moving the helper. Unlike the HI Solver, the Spline IK system does not use a goal. The positions of nodes in 3D space is the only factor that determines the shape of the linked structure. Rotating or scaling nodes has no effect on the spline or structure.

Note: 3ds Max also provides two other methods of inverse kinematic manipulation of hierarchies, which don't depend on a solver: Interactive IK and Applied IK.

IK with Bones

While you can apply an IK solver to any hierarchy of objects, a system of Bones combined with an IK solver is a good way to animate a character.

A bones system is a jointed, hierarchical linkage of bone objects. Bones are used as an armature on to which objects are linked. If you use the skin modifier, you can "skin" an object to the bones, so the animation of the bones deforms the mesh that models a character. If you have a jointed character, you can use linkage or constraints so the bones animate the mesh.

Animating bones with skin causes the skin to stretch or shrink.
Animating bones with skin causes the skin to stretch or shrink.

Turning Other Objects into Bones

Any object can be turned into a bone object. Select the object, choose Animation ➤ Bone Tools. On the Object Properties rollout, turn on Bone On. You can then choose Show Links Only to replace the display of the objects with the bones. This can be useful if you have a geometrically intensive hierarchy to animate. The interactive viewport response will be much quicker when the geometry is hidden and displayed only at links.

You can display any object as a bone object. Select the object, then choose Animation ➤ Bone Tools. This opens the Bone Tool’s floater. On the Object Properties rollout, turn on Bone On. Then go to the Display panel, and on the Link Display rollout turn on Display Links and Link Replaces Object, which displays the bones instead of the object. This can be useful if you have a geometrically intensive hierarchy to animate. The interactive viewport response is faster when the geometry is hidden and displayed as links only.
Any object hierarchy can be displayed as bones.

Bones can scale, squash and stretch over time. See Using Objects as Bones.

Link Display
You can use Display Links and Link Replaces Object to display the links instead of the object. These settings are found on the Link Display rollout on the Display panel. This can be useful if you have a geometrically intensive hierarchy to animate. The interactive viewport response is faster when the geometry is hidden and displayed only at links.

Advantages of Animating Bones with IK
It is possible to animate a character's motion through forward kinematics, rotating each limb into position from the shoulder to the fingers, and the hips to the toes. But it's a lot quicker and more realistic to use inverse kinematics to simulate the foot interacting with the ground. And it is a lot easier to control when you need to make changes to the animation. Rather than having keyframes on every bone in the chain, you have to make changes to only one node, to change the animation of the entire chain.

On the other hand, it is common for animators to use IK for the legs and FK for the torso and the arms. FK offers a bit more control for posing the upper body. It is not necessary to use IK for every character animation task. Using the HI IK solver allows you to jump back and forth easily between FK and IK.

How to Apply an IK Solver
You can apply an IK solver when you create a Bones system, or from the Animation menu:

- **Bones creation**
  When you create bones, turning on the Assign to Children option lets you apply an IK solver immediately. Default=Off.

- You can apply the IK solver to a part of the skeleton using Animation menu ➤ IK Solver. You must select the node where you want the IK chain to start, then choose Animation menu ➤ IK Solvers and select the solver. Then click the node where you want the chain to end. The chain will be created between the pivot points of the two nodes. If you are using an IK Limb Solver, the IK goal will be placed two bones down from the highest node you've selected in the chain.

Where to Adjust the IK Solver
You adjust IK solver settings in the Motion and Hierarchy panels:

- **Motion panel**
  When the goal of an IK chain is selected, the motion panel displays the rollouts for the individual IK solver.

  **HI Solver**
Motion panel parameters adjust the swivel angle of the solver plane, to point the knees and elbows. Also, the controls in the IK Solver rollout let you mix periods of IK with FK (forward kinematics) through the Enabled button and the IK button.

**HD Solver**

Motion panel parameters assign, remove, and edit the end effector for the currently selected joint. You can parent the end effector to another object, and return the skeleton to an initial pose. Changing IK controller parameters in the Motion panel affects the entire HD IK chain, even when you’ve selected only a single bone.

**IK Limb Solver**

Motion panel parameters for the IK Limb solver are the same as for the HI Solver.

**Spline IK Solver**

Motion panel parameters allow you to activate/deactivate the solver, adjust the bones assigned to the start and end joints, adjust start and end twist angles and make twist handle display settings.

**Hierarchy panel**

**HI Solver**

When a bone is selected, the IK panel displays controls to activate and limit the rotational joints, and set a preferred angle for the joints. The preferred-angle poses of all the joints help control the direction of rotation of the limbs. When a goal is selected, the IK panel is blank.

**HD Solver**

Select the end effector of an HD IK chain. In the Hierarchy panel, click IK. The controls that appear affect the HD Solver. You will also find the tools to bind to follow objects, and set precedence and joint limits, damping and spring back.

**IK Limb Solver**

When a bone is selected, the IK panel displays controls to activate and limit the rotational joints, and set a preferred angle for the joints. The preferred-angle poses of all the joints help control the direction of rotation of the limbs. When a goal is selected, the IK panel is blank. If you select a bone before applying an IK Solver, a different set of sliding and rotational joint parameters will be displayed, but these will be replaced once an IK Solver is applied.

**Spline IK Solver**

Select the spline of an Spline IK chain. In the Hierarchy panel, click IK. The controls that appear affect the Spline IK Solver. You will also find the tools to bind to follow objects, and set precedence and joint limits, damping and spring back. The controls in the IK panel are similar to the HD Solver.

**Procedures**

To add an IK solver to a hierarchy or bones system:

1. Create a bones system or any other linked hierarchy of objects.
2. Select a bone or an object where you’d like the IK chain to start.
3. Choose Animation menu ➔ IK Solver, and then choose the IK solver:
   - HI Solver for character animation
   - HD Solver for mechanical assemblages with sliding joints
4. Click where you want the IK chain to end.

If you are using the IK Limb Solver, you must apply the IK Solver to control only two bones.

The IK solver appears in the viewport.

To create a bones hierarchy that uses an IK solver:

1. Go to the Create panel, click (Systems), and click Bones.
2. On the IK Chain Assignment rollout, choose an IK solver from the list.
3. Turn on Assign To Children.
4. Click and drag in a viewport to create the bones. Right-click to stop bone creation.

The bones are created with the IK solver already applied.

Note: If you use the Spline IK solver, a Spline IK Solver dialog opens where you can make special settings for the spline and helpers used by the solver.

To display a hierarchy of objects as bones:

1. Select the hierarchy of objects in the viewport.
2. From the Animation menu, choose Bone Tools.

This opens the Bone Tools dialog.

3. Expand the Object Properties rollout.
4. In the Bone Properties group, turn on Bone On.
5. On the Display panel, scroll down to Link Display and expand it.
6. On the Link Display rollout, turn on Display Links, and Link Replaces Object.

The objects disappear and the links are displayed as bones.

Definitions:
- **History-Independent (HI) IK Solver**
  The HI (History-Independent) Solver does not rely on IK solutions calculated in previous keyframes in the timeline, so it is just as fast to use at frame 2000 as it is at frame 20.
- **History-Dependent (HD) IK Solver**
  Animating with the HD Solver lets you use sliding joints combined with inverse kinematics. It has controls for spring back, damping, and precedence not found in the HI Solver. It also has quick tools for viewing the initial state of the IK chain. Use it for animations of machines and other assemblies. Since this is history dependent, performance is slower at the end of long animations. For lengthy scenes, use HI Solvers, if possible.
- **IK Limb Solver**
  The IK Limb solver is specifically meant for animating the limbs of human characters; for example, the hip to the ankle, or the shoulder to the wrist. Each IK Limb solver affects only two bones in a chain, but multiple solvers can be applied to different parts of the same chain. It is an analytical solver that is very fast and accurate in viewports.
- **Spline IK**
  The Spline IK solver uses a spline to determine the curvature of a series of bones or other linked objects.

**Inverse Kinematics (IK)**
Inverse kinematics (IK) is a method of animating that reverses the direction of the chain manipulation. Rather than work from the root of the tree, it works from the leaves.

Using IK to animate a leg

Let's take the example of an arm. To animate an arm using forward kinematics, you rotate the upper arm away from the shoulder, then rotate the forearm, the hand from the wrist and so, on adding rotation keys for each child object.

To animate the arm using inverse kinematics, you move a goal that positions the wrist. The upper and lower arms are rotated by the IK solution which moves the pivot point of the wrist, called an end effector, toward the goal.

In the case of a leg, for example, the foot is constrained to the floor by the goal. If you move the pelvis, the foot stays put since the goal has not moved, and this causes the knees to bend. The entire animation is contained in keyframes for the goal and the root, without keys being applied to the individual chain objects.

With inverse kinematics you can quickly set up and animate complex motions. The basic procedure involves these tasks:

- Build a model. It could be a jointed structure or many pieces or a single continuous surface.
- Link the jointed model together and defining pivot points, as described in Hierarchies. For a continuous-surface model, create a Bones structure or use a biped to animate the skin of the character.
Apply **IK solvers** to the jointed hierarchy. You will probably create several IK chains throughout the hierarchy, rather than just one. You might also create several independent hierarchies, rather than link everything together in one large hierarchy. For simple inverse kinematic animation you can use interactive IK, without applying any IK Solver.

Define joint behavior at the pivot points, setting limits or preferred angles, depending on the type of IK solvers you are using. Here you can set up sliding joints or rotating joints. You might also need to move the root of the hierarchy, and you might want to add control objects such as dummies or points at this point.

Animate the goal (in the case of an **HI Solver** or **IK Limb solver**) or the end effector (in the case of the **HD Solver**). This animates all the components of the IK chain. You can apply constraints to the goals or control objects or to the root of a chain.

You can externally reference IK chains in your scene. An XRef IK chain behaves the same as a non-XRef chain, except that you cannot retarget its XRef controller once it is in your master scene. For more information, see **XRef Objects**.

### Control Objects to Assist IK

You can link a goal or an end effector to points, splines, or dummy objects that serve as quick controls to translate or rotate the end of the chain. These control objects can be linked together as well, or they can be controlled with constraints. You can also use parameter wiring to build relationships between these control objects.

You can wire control objects to **manipulator helpers** or to **custom attributes**, creating an easily accessible interface for your animatable model.

You can add further controls to manipulate the elements in the middle of the chain.

Note: In the **HI Solver**, the **swivel angle** has its own manipulator, which can be animated or linked to another target object.

### Differences Between Forward and Inverse Kinematics

Forward kinematics uses a top-down method, where you begin by positioning and rotating parent objects and work down the hierarchy positioning and rotating each child object.

Basic principles of forward kinematics include:

- Hierarchical linking from parent to child.
- Pivot points defining joints between objects.
- Children inheriting the transforms of their parents.

These principles are fairly forgiving. As long as everything is linked together and the pivots are located at joint locations, you can successfully animate the structure.

Inverse kinematics (IK) uses a goal-directed method, where you position a goal object and 3ds Max calculates the position and orientation of the end of the chain. The final position of the hierarchy, after all of the calculations have been solved, is called the IK solution. There are a variety of IK solvers that can be applied to a hierarchy.

Inverse kinematics starts with linking and pivot placement as its foundation and then adds the following principles:

- Hierarchical linking from parent to child.
- Pivot points defining joints between objects.
- Children inheriting the transforms of their parents.

These principles are fairly forgiving. As long as everything is linked together and the pivots are located at joint locations, you can successfully animate the structure.
Joints can be limited with specific positional and rotational properties.

Position and orientation of parent objects are determined by the position and orientation of child objects. Because of these additions, IK requires greater thought about how you link your objects and place pivots. Where many different solutions for linking objects may be suitable for forward kinematics, there are usually just a few good solutions for any given IK approach.

Inverse kinematics is often easier to use than forward kinematics, and you can quickly create complex motions. If you need to edit those motions later, it can be simpler to revise the animation if you are using IK. It also is the best way to simulate weight in an animation.

Animating with Forward Kinematics
Dec 17, 2014 | In-Product View

The default method of manipulating a hierarchy uses a technique called forward kinematics.

The basic principles employed by this technique are:

- Hierarchical linking from parent to child
- Placement of pivot points to define the connecting joint between linked objects
- Inheritance of position, rotation, and scale transforms from parent to child

You animate the objects of a hierarchy in much the same way you animate anything else. Turn on the Auto Key button and transform members of the hierarchy at different frames. However, you need to be aware of a few special issues for animating hierarchies.

How Links and Pivots Work

Once two objects are linked together, the child object maintains its position, rotation, and scale transforms relative to its parent object. These transforms are measured from the pivot of the parent to the pivot of the child.

For example, consider the two boxes in the following figure. The larger box is the parent of the smaller. The pivots and link between the boxes are indicated to show how the link works. The link extends from the pivot of the parent and connects to the pivot of the child. You can think of the child’s pivot as being the joint between the parent and child.
Parent and child objects linked by their pivot points.

Rotating the parent affects the position and orientation of the child object.
Rotating the child does not affect the parent.

Links act as a one-way conduit to transmit the transforms of a parent object to its child object. If you move, rotate, or scale the parent, the child is moved, rotated, or scaled by the same amount. Because hierarchical links are one-way, moving, rotating, or scaling the child has no effect on its parent.

The end result is that transforms applied to a child object are applied in addition to any transforms inherited from the child’s parent.

**Animating a Parent Object**

Only transforms are passed from parent to child. Animating a parent object using move, rotate, or scale animates the parent and the subtree attached to the parent.

Animating a parent’s modifiers or creation parameters has no effect on its descendants.

Moving the root parent moves the whole hierarchy.
Rotation of a parent object is passed to all the child objects.

**Animating a Child Object**

With forward kinematics, a child is not constrained by its link to a parent. You can move, rotate, and scale children independent of their parents.

Moving the last child object does not affect any of the previous objects in the hierarchy.
Rotating a child object in the middle of the hierarchy affects all the descendants but none of the parents.

If you want to manipulate parent objects by moving the last child in the chain, use inverse kinematics.

Manipulating the Hierarchy
A child object inherits the transforms of its parent, and the parent inherits the transforms of its ancestors all the way up the hierarchy to the root object. Because forward kinematics employs this method of inheritance, you must position and animate your hierarchies using a top-down method.
Manipulating the hierarchy of a leg.

Consider the linked mannequin in the figure. If you want to position the mannequin’s right foot to rest on top of the soccer ball beside it, you perform the following steps:

1. Rotate the right thigh so the entire leg is above the soccer ball.
2. Rotate the right shin so the foot is near the top of the soccer ball.
3. Rotate the right foot so it is parallel with the top.
4. Repeat steps 1 through 3 until the foot is properly placed.

You always start transforming objects at the highest-level parent affected by the motion and work your way down the hierarchy to the last child.

You have considerable control over the exact placement of every object in the hierarchy using forward kinematics. However, the process can become tedious with large and complex hierarchies. In such situations, you might want to use inverse kinematics.

Topics in this section

- **Using Dummy Objects**
  The primary use of dummy helper objects is to assist in creating complex motions and building complex hierarchies. Because dummies are invisible when rendered, they are an excellent choice for offset joints, connectors between objects, and handles for complex hierarchies. Dummies and Points can act as null objects that function as controls for transforming parts of an IK chain.

- **Animating Links**
  You assign a Link constraint to an object to animate links from one parent to another. You use a link constraint instead of using the regular Select and Link and Unlink Selection buttons on the toolbar. (See Animation Constraints.)

- **Adjusting Object Transforms**
  You use the features on the Adjust Transform rollout to transform objects after they have been linked without transforming descendents, and to reset an object’s transform.

- **Locking Object Transforms**
  You can lock an object’s ability to move, rotate, or scale about any of its local axes by selecting objects and then setting options on the Locks rollout of the Hierarchy panel.

- **Animating Attachment**
  You assign an Attachment constraint to cause an object to hold a position on the surface of another object.

- **Changing Link Inheritance**
  Links can transmit transform information from a parent to a child. By default, a child inherits all of the transforms of its parent. To set an object’s ability to inherit the move, rotate, and scale transforms of its parent, you use the Inherit rollout of the Hierarchy panel. Use its settings to limit which transforms a child inherits.

- **Link Inheritance (Selected) Utility**
  The Link Inheritance (Selected) utility constrains the links between multiple objects in a selection set for any axis of position, rotation, or scale.

**To Use the Skin Modifier**

Dec 17, 2014  |  In-Product View
The Skin modifier is a skeletal deformation tool that lets you deform one object with another object. Mesh, patch, or NURBS objects can be deformed by bones, splines, and other objects.

- Select a mesh, patch, or NURBS object. > Modify panel > Modifier List > Object-Space Modifiers > Skin
- Select a mesh, patch, or NURBS object. > Modifiers menu > Animation > Skin

When you apply the Skin modifier to a mesh and then assign bones within the modifier, each bone receives a capsule-shaped "envelope." Vertices of the modified object within these envelopes move with the bones. Where envelopes overlap, each vertex's motion is a blend of the motions of bones that affect the vertex. This is accomplished with weighting.

By default, each vertex that's affected by a single bone is given a Weight value of 1.0, which means the vertex responds only to that bone's motion. Vertices within the intersection of two bones' envelopes have two Weight values: one for each bone. And you can use Skin modifier tool sets such as the Weight Tool dialog to arbitrarily assign vertices to any number of bones.

The ratio of a vertex's Weight values, which always total 1.0, determine the relative extent to which each bone's motion affects the vertex. For example, if a vertex's weight with respect to bone 1 is 0.8 and its weight with respect to bone 2 is 0.2, then the motion of bone 1 will have four times greater influence on the vertex than will the motion of bone 2.

The initial envelope shape and position depends on the type of bone object. Bones create a linear envelope that extends along the longest axis of the bone geometry. Spline objects create envelopes that follow the curve of the spline. Primitive objects create an envelope that follows the longest axis of the object.

You can apply the Skin modifier to several objects at the same time.

In 3ds Max you can mirror envelope and vertex assignments from one side of the mesh to the other with commands on the Mirror Parameters rollout.

Tip: When working with a high-resolution skin mesh, you can improve viewport performance significantly by applying the Turn to gPoly modifier below the Skin modifier on the stack.
Angle Deformers
You can deform the skin mesh based on the angle of the bones. Three deformers that you add via the *Gizmos rollout* let you shape the mesh based on bone angles:

- The *Joint Angle and Bulge Angle deformers* use a lattice similar to an *FFD lattice* to shape the mesh at a specific angle.
- The *Morph Angle deformer* morphs the mesh at specified angles. You create morph targets with modifiers above the Skin modifier in the stack. Alternatively, you can create a copy of the mesh with the *Snapshot tool*, then deform the mesh with standard tools.

Procedures
To use the Skin modifier:

By default, the Skin modifier assigns an envelope to each bone and automatically adjusts the envelope to encompass nearby mesh vertices. Then, when the bone moves, it brings along all vertices within its envelope.

Note: The bone "carries" its associated vertices by virtue of the envelope's having assigned Weight values for that bone to the vertices. You can change these assigned values by adjusting the envelopes and/or specifying Weight values explicitly.

1. Prepare the skin (mesh or patch object) and skeleton (bones, CAT or Biped skeletons, or other objects).
   Carefully place the skeleton inside the mesh or patch object so that its components are able to influence polygons or patches in their immediate vicinity.
   Tip: The skeleton can already be animated, or you can animate it after applying its components to the Skin modifier.
2. Select the mesh or patch object and apply the Skin modifier.
3. In the Parameters rollout, click Add and choose the skeleton objects.
4. Turn on Edit Envelopes and select an envelope to modify the volume in which each bone can influence the surrounding geometry.
To weight vertices manually:

The default method of using envelopes to determine which bones affect which mesh vertices can suffice for many situations, but if you need more control, you can opt to apply vertex weights explicitly.

1. Make sure Parameters rollout > Edit Envelopes is on, then turn on Vertices.
2. On the mesh, select vertices to weight manually.
   Each selected vertex is surrounded by a small white rectangle.
3. In the list of bones on the Parameters rollout, highlight the name of the bone for which you want to change the vertex weights.
4. In the Weight Properties group on the Parameters rollout, set the Abs. Effect value to the new vertex weight.
   The Abs. Effect value is a decimal number between 0.0 and 1.0. The latter value means the vertex is controlled only by the highlighted bone. Values less than 1.0 means that other bones can also affect the vertex’s motion.

To mirror envelope or vertex weight settings:

1. Adjust envelopes and vertex weights on one side of the mesh.
2. On the Mirror Parameters rollout, click Mirror Mode. The mirror plane appears at the position and orientation of the mesh's pivot point.

3. If the mirror plane is not at the center of the mesh, change the Mirror Offset parameter to move the plane to the center.

4. If some vertices in the left or right side of the mesh are red rather than blue or green, increase the Mirror Thresh value until all vertices are blue or green.

5. On the Mirror Parameters rollout, click the appropriate Paste button to paste green or blue envelopes or vertex weights to the other side of the mesh.

To adjust the skin and/or bones without affecting the envelopes:

1. Save the scene. This is a potentially destructive operation, so it's best not to take any chances with your data.

2. Select the object to which the Skin modifier is applied.

3. On the Advanced Parameters rollout, turn off Always Deform.

4. Apply any necessary transforms to the mesh/patch object or bones objects.

5. Turn Always Deform back on.

To adjust the bones only, you can also use skin pose.

Example: To apply the Skin modifier to a cylinder with a bones skeleton:

1. On the Create panel, with (Geometry) active, under Standard Primitives, click Cylinder.

2. In the middle of the Top viewport, click and drag 20 units to create the base of the cylinder.

3. Release the mouse button and drag up 130 units to establish the height of the cylinder.

4. On the parameters rollout, set Height Segments to 20. This provides mesh detail for a smooth surface deformation.

5. On the Create panel, click (Systems). On the Object Type rollout, click Bones. Make sure an IK Solver is chosen in the IK Solver list. Turn on Assign To Children. (This should turn on Assign To Root as well.)

6. In the Front viewport, click successively three times: below the cylinder, in the middle of the cylinder, and above the top of the cylinder.

7. Right-click to end Bones creation. Three bones display. Two of them are within the middle of the cylinder.

8. Select the cylinder.


10. On the Skin modifier Parameters rollout, click Add, and use the Select Bones dialog to select the three bones. The names of the bones are now displayed in the list.

11. In the Front viewport, select the bone end effector (IK Chain01) and move it around.
The cylinder deforms to follow the bones. To adjust envelopes to refine the surface deformation, choose the Skin modifier's Envelope sub-object level, and use the Edit Envelopes controls to resize envelopes and change vertex weights.

**Extension** Example: To use dual-quaternion (DQ) skinning:

By default, the Skin modifier uses linear weighting, which works fine in most cases when deforming a character mesh with a skeleton. However, if a bone twists about its longest axis, the skin mesh tends to collapse, causing an unsightly loss of volume. In such cases you can apply dual-quaternion skinning, which prevents the loss of volume.

1. Make sure the joint where you'll apply DQ skinning uses blended weighting between the two adjacent bones. One way to do this is by increasing the diameters of the envelope capsules at the joint so they overlap.

   ![Image showing increased diameter of envelope capsules](image)

   The yellow area shows where the vertex weighting is shared between the neighboring bones.

2. Rotate the "driver" bone about its longest axis so that the resulting loss of volume is readily apparent. Then undo the rotation.

   ![Image showing mesh constriction](image)

   Mesh constriction caused by rotating the "driver" bone about its long axis

3. Select the mesh object, go to the Envelope sub-object level, turn on Vertices, and select the vertices at the joint.

   ![Image selecting vertices](image)
4. In the Dual Quaternion group on the Parameters rollout, make sure DQ Skinning Toggle is on, as it is by default, and then turn on Blend Weights.

This turns the mesh black in shaded viewports and allows you to use all vertex-weighting tools except Weight Table to apply a DQ mask instead. In this example, you'll use the Weight Tool dialog, but you can use whichever tools for setting weights you're most comfortable with.

5. In the Weight Properties group, click (Weight Tool) to open the Weight Tool dialog.

6. On the dialog, click the 1 button, at the right end of the second row.

This sets the highest possible DQ blending value for all selected vertices, as depicted in the mesh with white geometry.

7. Also on the dialog, click the Blend button several times. It's located at the right end of the bottom row of buttons.

This smooths the transition between the weighted vertices and their neighbors.
8. Close the Weight Tool dialog, then turn off Blend Weights on the Parameters rollout. The mesh's appearance returns to normal.

9. Rotate the driver bone again about its long axis. This time the loss of volume resulting from the twisting motion is significantly reduced.

Example: To use a morph angle deformer:

Create the cylinder and bones from the preceding procedure before you continue with this procedure.

1. At frame 50, animate bone 2 so that bones 1 and 2 represent a 90-degree angle.
2. At frame 0, the bones should be straight at about a 180-degree angle.
3. Go to frame 0.
4. In the Parameters rollout, turn on Edit Envelopes.
5. Select the child bone (bone 2) in the modifier's list of bones.
6. In the Select group, turn on Vertices.
   This allows you to select vertices.

7. In the viewports, region-select a good portion of the vertices that are controlled by both bones.
8. In the Gizmos rollout, select the Morph Angle Deformer in the drop-down list, and then click Add Gizmo.
   The Deformer Parameters rollout displays. A base morph target is the first and only target in the list.
9. Scrub the Time Slider to frame 50.
10. Add an Edit Mesh modifier above the Skin modifier on the modifier stack.
11. In the Edit Mesh modifier, turn on Vertex and Soft Selection.
12. Edit the mesh to the shape you want.
13. In the stack, go back down to the Skin modifier. If 3ds Max displays the topology warning dialog, click Yes.
15. Delete the Edit Mesh modifier from the stack. There is a doubling effect of the morph if you don’t delete or deactivate the Edit Mesh modifier.
16. Scrub the time slider. The mesh morphs as the bone angle changes.

Interface
Modifier Stack

Envelope
Access this sub-object level to modify envelopes and vertex weights.

Tip: You can use the quad menu (right-click) to choose this sub-object level.

Quad Menu
A number of Skin modifier commands are available from the quad menu on the Tools 1 and Tools 2 quadrants:

<table>
<thead>
<tr>
<th>Remove Cross Section</th>
<th>Isolate Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Cross Section</td>
<td>Unfreeze All</td>
</tr>
<tr>
<td>Remove Bones</td>
<td>Freeze Selection</td>
</tr>
<tr>
<td>Add Bones</td>
<td></td>
</tr>
<tr>
<td>Paste to All Envelope</td>
<td>Unhide by Name</td>
</tr>
<tr>
<td>Paste Envelope</td>
<td>Unhide All</td>
</tr>
<tr>
<td>Copy Envelope</td>
<td>Hide Unselected</td>
</tr>
<tr>
<td>Envelope Top-level</td>
<td>Hide Selection</td>
</tr>
</tbody>
</table>

Rollouts
The Skin modifier has several rollouts. These are described in the topics that follow.
Parameters Rollout (Skin Modifier)
The Parameters rollout provides controls for adding bones to the modifier, adjusting envelopes, and setting vertex weights manually. Alternative methods for setting weights are also available here.

Mirror Parameters Rollout (Skin Modifier)
Mirror mode lets you edit envelopes and vertex assignments symmetrically.

Display Rollout (Skin Modifier)
Controls how Skin features appear in viewports.

Advanced Parameters Rollout (Skin Modifier)
Contains additional global Skin modifier controls.

Gizmos Rollout (Skin Modifier)
Lets you set up Joint Angle, Bulge Angle, and Morph Angle deformations.

Load Envelopes Dialog (Skin Modifier)
The Load Envelopes dialog associated with the Skin modifier allows you to load saved envelopes to specific bones. This resizable dialog shows the current envelopes in your scene and the incoming envelopes. Use the controls to manipulate the incoming envelopes so they align with the current envelopes.

Weight Tool Dialog (Skin Modifier)
This dialog is launched from the Skin modifier and provides tools to select vertices and assign them weights. You can also copy, paste, and blend weights between vertices. Each vertex you select displays the objects contributing to its weighting in the dialog list.

Weight Table (Skin Modifier)
The weight table for the Skin modifier is used to change vertex weights for several vertices and bones at a time. This table appears when you click the Weight Table button.

Bone Editing Tools Rollout

Dec 17, 2014 | In-Product View

Controls on the Bone Editing Tools rollout let you create and modify bone geometry and structure, and set bone color for one or more bones.

Standard menu: Animation menu > Bones Tools > Bones Tools dialog > Bone Editing Tools rollout
Enhanced menu: Animation menu > Create (Animation Set) > Bones Tools > Bones Tools dialog > Bone Editing Tools rollout
Bone Pivot Position group

Bone Edit Mode

Lets you change the lengths of bones and their positions relative to one another.

When this button is on, you can change the length of a bone by moving its child bone. In effect, you can scale or stretch a bone by moving its child bone while in this mode. You can use this tool both before and after assigning an IK chain to the bone structure.

When Bone Edit Mode is on, you cannot animate, and when Auto Key or Set Key is on, Bone Edit Mode is unavailable. Turn off Auto/Set Key to edit bones.

Note: Moving a bone in Bone Edit mode affects the length of both the child and its parent. If the bones aren't spatially aligned in the usual way (for example, if you are using other objects as bones), this might have unexpected results.

Bone Tools group

Create Bones

Begins the bone-creation process. Clicking this button is the same as clicking Create panel ➞ Systems ➞ Bones System.

Create End

Creates a nub bone at the end of the currently selected bone. If the selected bone is not at the end of a chain, the nub is linked in sequence between the currently selected bone and the next bone in the chain.

Remove Bone—Removes the currently selected bone. The bone’s parent bone is stretched to reach the removed bone’s pivot point, and any children of the removed bone are linked to its parent. Any IK chains that included the removed bone will remain intact.

Connect Bones—Creates a connecting bone between the currently selected bone and another bone. When you click this button, a dotted line appears in the active viewport from the first selected bone. Move the cursor to another bone to create...
a new connecting bone. The first selected bone will become a parent to the connecting bone, which is in turn a parent to the second selected bone.

**Delete Bone** - Deletes the currently selected bone, removing all its parent/child associations. A nub is placed at the end of the deleted bone's parent. Any IK chains that included this bone become invalid.

**Reassign Root** - Makes the currently selected bone the root (parent) of the bone structure.

If the current bone is the root, clicking this has no effect. If the current bone is the end of the chain, the chain is completely reversed. If the current bone is in the middle of the chain, the chain becomes a branching hierarchy.

**Refine** - Splits a bone in two. Click Refine, and then click a bone where you want it to split.

**Mirror** - Opens the Bone Mirror dialog (see following), which lets you create mirror copies of selected bones without changing the sign of the bones' scale. Instead, Mirror flips one of the bone axes: Y or Z. You can specify the mirroring axis and the flip axis with the dialog controls.

### Bone Mirror dialog

Opens when you click the Mirror button. Use it to specify the mirroring axis, the flip axis, and an offset value.

While the dialog is open, you can see a preview of the mirrored bone(s) in the viewports. Click OK to create the bones, or Cancel to prevent creation.

**Mirror Axis** - Choose an axis or plane about which the bones will be mirrored: X/Y/Z or XY/YZ/ZX.

**Bone Axis to Flip** - To avoid creating a negative scale, choose the bone axis to flip: Y or Z.

**Offset** - The distance between the original bones and the mirrored bones. Use this to move the mirrored bones to the other side of the character.

### Bone Coloring group

**Selected Bone Color** - Sets the color for selected bones.

**Apply Gradient** - Applies a gradient color across several bones based on the Start Color and End Color values. This option is available only when two or more bones are selected. The Start Color is applied to the highest parent bone in the selected chain, while the End Color is applied to the last child object in the selected chain. Intermediate colors in the gradient are applied to bones in between.

**Start Color** - Sets the starting color for the gradient.

**End Color** - Sets the ending color for the gradient.

### Physique

Use the Physique modifier to attach a skin to a skeleton structure such as a biped. The skin is a 3ds Max object: it can be any deformable, vertex-based object such as a mesh, a patch, or a shape. When you animate the skeleton with skin attached, Physique deforms the skin to match the skeleton's movement.
Animating the underlying skeleton enables you to animate a single contiguous model of a character that bends, creases, and bulges about an arbitrary number of joints within the attached skeleton.

With Physique, you can define how the skin behaves when it deforms. For example:

- You can make portions of the skin solid, excluding them from Physique's deformation, though solid portions still move along with the root node of the skeleton they are attached to. These solid portions are said to be root vertices.
- You can make portions of the skin deformable. They move with the deformation spline, the smooth curve running through the links of the skeleton they are attached to.
- You can make portions of the skin rigid, directly moving along with the skeleton they're attached to.
- You can add bulges to simulate bulging muscles. Bulges are controlled by editable cross sections of the skin, and by bulge angles that you set.
- You can add tendons to distribute the effect of one bone's motion to areas of the skin other than those around the bone itself.
- You can save Physique data to a Physique (.phy) file, preserving data common to all objects sharing a given Physique modifier. Later, you can reload the data file, either to restore the data that belongs to a particular skin or portion of skin.

Physique works with bipeds created and animated using the Biped plug-in, and with 3ds Max hierarchies, including the Bones systems. Physique also works with bones that are not in a hierarchy and splines.

**Understanding Physique**

Physique is a modifier that, when applied to a mesh, allows the movements of an underlying skeleton to move the mesh seamlessly, like bones and muscle under a human skin. Physique works on all point-based objects including geometric primitives, editable meshes, patch-based objects, NURBS, and even FFD space warps. For NURBS and FFDs, Physique deforms the control points, which in turn deform the model. You can attach it to any skeleton structure including a biped, 3ds Maxbones, splines, or any 3ds Max hierarchy. When you apply Physique to the skin objects and attach the skin to the skeleton, Physique determines how each component of the skeleton influences each vertex of the skin, based on settings you specify.

Physique affects a mesh after you click Attach To Node on the Physique rollout and select a root node in the viewports. During the attach process, Physique works its way through all of the children in a hierarchy, starting at the object you select, and creates its own links with associated envelopes for each link it finds. The links created by Physique are referred to within this documentation as the Physique deformation spline. Vertices that fall within envelopes are influenced to follow the links and animate the mesh. Splines and 3ds Max bones can also be added using the Add button in the Floating Bones rollout.

**Biped and Physique**
When attach a biped to Physique, the modifier traces its way from the pelvis down the legs to the toes and up through the spine, branching at the collar to the arms, hands, and fingers, and up the neck to the head. A link and associated envelopes are created for each link found. If any other objects, including 3ds Max bones, are linked to the biped, Physique treats them similarly: it creates a link and envelopes.

If your character has additional limbs, be sure to link 3ds Max bones to the biped for the extra arms before using Attach To Node to create links and envelopes. When you use Attach To Node, it creates links and envelopes for all the links in the biped and for the linked bones. Link non-deformable objects like a sword *after* using Attach To Node. That way Physique doesn’t create a link and envelopes for the sword.

### Envelopes and Weighted Vertices

The Physique modifier uses envelopes as its primary tool for controlling skin deformation. It also provides tendons and bulge angles for fine-tuning mesh deformation after envelopes are adjusted. All envelopes have an inner and outer bound (boundary). Vertices falling within the inner bound of a single link receive a full weight of 1.0 from that link. Those falling outside the outer bound receive no weight from that link. Vertices falling between the inner and outer bounds receive a weight from 0 through 1.

Vertices move together with the link that influences them. Where multiple envelopes encompass a vertex, that vertex receives weight from each envelope and follows each link to an average position based on these weights. This weighting from multiple links is considered *blending*. It is possible that weights assigned to some vertices don’t reach a total weight of 1.0 or greater. Rather than leaving these vertices behind, Physique by default normalizes them to a value of 1.0.

Adjusting falloff, overlap, scale, and other envelope parameters changes vertex weight distribution across links. This change, in turn, affects the way skin behaves as the biped moves. By and large, you correct the way skin deforms on a character by adjusting envelopes.

### Deformable and Rigid Envelopes

There are two Envelope types per link: **deformable and rigid**. Deformable envelopes follow the Physique deformation spline that runs through the joints in the hierarchy, and can be deformed using bulge angles, tendons, and link parameters. Rigid envelopes determine vertex-link assignment based upon the linear 3ds Max link and move in an immobile relationship to the link. Vertices in a rigid envelope, however, are deformed (blended) in the overlap area of other envelopes.

Typically you use deformable envelopes; however, game developers with game-engine restrictions might want to use rigid envelopes exclusively. Both rigid and deformable envelopes can be turned on for the same link. For example, by scaling both envelopes, you could deform the shoulder with a rigid envelope and the armpit with a deformable envelope.
The Number of Links That Can Affect a Vertex

Any number of overlapping envelopes (N Links) can influence vertices. Normally, N Links are preferred. For special purposes such as games requirements, you can limit the number of links (envelopes) that can affect a vertex. The No Blending parameter is like the method used in version 1 of Physique: a vertex is assigned to only one link.

Physique Workflow

Before Physique is applied, align the biped to the mesh in Figure mode. Use a pose with the arms outstretched so the hands are away from the torso. Save a figure file, so it’s easy to return to this pose whenever you need. Select the mesh and choose Physique in the Modify panel. Turn on Attach to Node and select the root node in the hierarchy (biped Pelvis or root node in a bones hierarchy, not the COM). In the Physique Initialization dialog box, click Initialize to create default envelopes based on the links in the hierarchy. The remainder of the work is adjusting envelopes and optionally adding bulge angles and tendons.

Envelope size, overlap, and other parameters are adjusted with the character in an animated position (with Figure mode turned off). By scrubbing the time slider back and forth, you can spot problem areas and adjust the envelopes affecting the problem areas. In Place mode is useful to keep the character stationary during envelope adjustment.

The finishing touches are link parameters, bulge angles, and tendons. You use link parameters to control skin sliding, the amount of twist, and crease blending as a character moves. Bulge angles let you expand areas like the biceps, and chest relative to the angle created by a link and its child in the hierarchy. Tendons can span multiple links in the hierarchy to stretch and pull a character skin.

Understanding Biped

Biped is a 3ds Max component that you access from the Create panel. Once you create a biped, you animate it using the Biped controls on the Motion panel. Biped provides tools to let you design and animate the figures and motion of characters.

The Biped

The biped skeleton created with the Biped module is a two-legged figure created as a linked hierarchy, and designed for animation. The biped skeleton has special properties that make it instantly ready to animate.

Figure and Keyframe Modes
character studio is designed to interchange motion and characters. In Figure mode, you pose the biped to fit your character model. In Keyframe mode, you animate the skeleton. Motions created for the biped can be saved and loaded onto other biped skeletons with different physical characteristics. For example, you could animate a giant ogre, save the animation, and load it onto a small child. Motion files are saved in the proprietary BIP format.

You can use these files in a variety of ways with Motion Flow, the Motion Mixer, and the Crowd tools to combine animation or animate multiple characters.

## Animating the Biped

The two primary methods used in creating biped animation are footsteps method and freeform method. Each has its advantages. You can convert animation between the two methods, or you can combine both in a single animation. For detailed information, see these sections: Footstep Animation and Freeform Animation.

## Biped Properties

The biped skeleton has these properties, designed to help you animate faster and more accurately:

- **A human structure**—Joints on the biped are hinged to follow human anatomy. By default, the biped resembles a human skeleton and has a stable inverse-kinematics hierarchy. This property means that when you move a hand or foot, the corresponding elbow or knee orients itself accordingly, and produces a natural human posture.

- **Customizable for non-human structures**—The biped skeleton can easily be made to work with a four-legged creature or an animal that naturally leans forward, such as a dinosaur.

- **Natural rotations**—When you rotate the biped spine, the arms maintain their relative angle to the ground, rather than behaving as though fused to the shoulders. For example, take a biped in a standing position, with arms hanging at its sides. If you rotate the spine forward, the fingers touch the ground rather than pointing behind it. This position is more natural for the hands, which speeds the process of keyframing the biped.

  This feature also applies to the biped head: When you rotate the spine forward, the head maintains a forward-looking orientation.

- **Designed for footsteps**—The biped skeleton is specially designed to animate with character studio footsteps, which help solve the common animation problem of locking the feet to the ground. Footstep animation also provides an easy way to rough out animation quickly. For more information, see Footstep Animation.
To work most efficiently with bipeds, it is important to follow the general workflow described in this topic.

Create Skin Geometry

Before you create a skeleton for a character, have a character skin ready to put the skeleton into.

Create a basic skin shape for your character using any of the 3ds Max modeling tools and surface types. Be sure to place your character's skin in a neutral pose with arms outstretched and legs spaced slightly apart. You may also want to add sufficient detail to your skin's mesh or control points around joints to facilitate deformation during movement.

Tip
Before adding a biped skeleton, freeze your character mesh. When the mesh is frozen, you can still see it, but you can't select or alter it, reducing the chance for error or frustration.

Create a Biped Skeleton

Once you have a character mesh, you can create a biped skeleton, or rig, to fit inside. Use Figure mode to set up your biped.

Before you position the skeleton, use controls on the Structure rollout to alter the biped to match your mesh, setting the number of links for the spine, arms, neck, or fingers, or adding props to represent weapons or tools.

Tip
You can use ponytails to create animated jaws, ears, or horns.

Note
Certain biped body parts, including fingers, tails, ponytails, props, and clavicles, can be repositioned in Figure mode to suit different characters.

When you position the biped inside your mesh, start with the center of mass (COM), which is the parent of all objects in the biped hierarchy. The COM should be positioned in line with the hips of the mesh character. Scale the pelvis so that the legs fit properly in the mesh, and then use Move and Scale on the 3ds Max toolbar to position your biped skeleton.
In addition to the standard move, rotate, and scale operations, you can also use modifiers to adjust the parts of the biped.

Biped body parts cannot be removed, however unwanted parts can be hidden. If you delete a part the entire biped will be deleted.

The following list includes some tips for positioning your skeleton:

- Use the **Page Up** and **Page Down** keys to cycle through links.
- Use **Rubber Band mode** to move and scale the arm and leg links on your skeleton simultaneously.
- Use tools from the **Bend Links rollout**, such as **Bend Links Mode** and **Twist Links Mode** to adjust tail, neck, spine, and ponytail links.
- The head, toes, and fingertips should extend slightly beyond the mesh extents to fulfill the requirements of Physique.
- Use the minimum number of fingers and toes.
  You need extra fingers or links only if you are planning on complex hand or foot animation. If your character is wearing gloves or shoes, then you probably only need one finger or toe, with one link.
- To create a biped with knees that bend backwards, rotate the biped calves or thighs of both legs 180 degrees about their local X-axis (along the length of the limb). When you exit Figure mode, the biped walks, runs, and jumps with reversed knees.
- When working with a mesh in a symmetrical pose, pose one side of the skeleton, and use controls on the **Copy/Paste rollout** to paste the posture to the opposite side of the biped.
When you are satisfied with your pose, check the alignment in all viewports to make sure that the skeleton is positioned correctly in the mesh.

Once you have successfully positioned a skeleton inside your character mesh, you are ready to attach the mesh with Physique. For more on this workflow, see Understanding Physique.