Chapter 8 - Character Editing

8.1 Character Editing Introduction

The Character Editing tools let you build skeletons that will allow you to pose and animate objects. The most obvious uses are for characters and animals, but you can find them useful in other situations too, such as for creating ropes, hoses, chains, and more!

There are two aspects to this kind of posing and animation. The first is the mesh itself, the object that you will want to move, bend and pose. The next part is the skeleton, made up of a series of joints connected together by bones. The skeleton will control the mesh attached to it, making it bend and deform as you move the joints on the skeleton.

This chapter will deal with creating and setting up the skeleton, and how to adjust it so that it can perform the movements you need it to make, and so that it controls the object well, making the correct parts bend and deform as you work with the skeleton.

The chapter begins with a look through each of the tools associated with this process, and then moves on to some introductory tutorials that take you from start to finish on creating a skeleton that will control a human character.

8.2 The Tools

Here you will find an introduction to each tool in the Character Editing toolkit. They are listed here in a grouping that reflects what each tool does, in an order that reflects a common workflow.

8.2.1 Building the Skeleton

Build Skeleton 

Build Skeleton is the tool that lets you start building the skeleton that will control your mesh. You can build the skeleton on its own in an empty scene. Usually though you will want to build your skeleton inside the character it is intended for. This will help trueSpace position the bones and make it faster and easier to create your skeleton, as well as giving you a good visual guide to follow when creating your skeleton.

Note that while building a skeleton you can select a joint you have already created and adjust its position, and then resume building by clicking on the joint you want to build from. This means you have access to the same functionality as found in the Shape Skeleton tool, without any need to change tools.

Shape Skeleton 

Virtually no skeleton will come out perfect from the build step, even with trueSpace helping position the bones for you based on your mesh. That's not a problem though as you can adjust the skeleton at any point using the Shape Skeleton tool to select existing joints and adjust them. This tool is used to edit a skeleton once it is complete, and you do not want to add any more joints to it.

Editing is performed by moving the joints or using the Move tool. If you hold CTRL while moving, then the whole branch is moved; without CTRL held while moving, then only the selected joint is moved (see the video below).

video link

Note that the Shape Skeleton tool also works as a "mode" - you can combine Shape Skeleton with dynamic posing, switching it on to see how your character bends when you move joints as you use the Dynapose tool.

Delete Joint 

Delete Joint will remove whatever joint is currently selected. Note that you must first select the joint you want to delete – you can use the arrow keys to navigate through the skeleton to select the joint, or select the joint in the Link Editor, or you can use the Shape Skeleton tool to select the joint. Once the joint is selected, click the Delete Joint tool (if no joint is selected, then the Delete Joint tool will be grey and you will not be able to use it).

Removing a joint will rebuild the connections in your skeleton, so deleting a joint in the middle of your skeleton will not "split it in two" but instead will create new bones and connections to ensure your skeleton remains intact.

Also note that removing a joint removes some bones, and this can affect skinning if the skeleton is already attached to a skin.

Remove Limb 

Remove Limb will delete the sub-tree for a limb, starting with the selected joint. This lets you selectively remove an arm, leg, hand, or similar. Simply select the joint using Dynapose, Shape Skeleton, Build Skeleton or in the Link Editor, and then click the Remove Limb tool. Note that the selected joint is NOT removed; only all the joints and bones after it are removed.

8.2.2 Connecting Skin and Bone

Once your skeleton is created, you need to join the skeleton to the mesh and other objects it is going to control. We'll look at the processes involved in this below.

Attach Skin To Skeleton 

Once your skeleton is constructed, you need to attach the skin to it so that the skeleton can then control the skin and bend and deform it as you bend and move the skeleton itself. Ensure that the skeleton itself is selected, and then click on the Attach Skin To Skeleton tool. Now point at the mesh that you want the skeleton to control – the mesh will highlight. Click on the mesh, and the skeleton is attached to it, giving you a character that you can now pose and animate!

This tool can also be used to detach a skeleton and skin that are already attached – select the character, then select this tool; point at the mesh with the mouse, and it will highlight; left click, and the mesh becomes detached from the skeleton. This can be useful if you want to edit the skeleton using the Limb Libraries for example, or if you want to reuse the skeleton in another mesh.

Attach Object To Bone 

Sometimes you want an object to follow the movement of the character, but the object should not deform like skin. These objects are usually not part of the character's body, and are usually solid and rigid – a good example would be a cup held in a character's hand. The cup should follow where the hand goes, but it should not flex and deform like the skin of the arm or hand. Other items that you might want to attach in this way are hats, helmets, weapons, shields, etc.

First, ensure the object you want to attach is in the correct location. First, select the Shape Skeleton tool and click on the bone that you wish to attach the object to. Next, click on the Attach Object To Bone tool, and then finally click on the object you wish attached to the bone.

Attach Skin To Bone 

This tool is used when you want to attach a skin to particular bones, without trueSpace assigning the vertices to any bones and only using Bone Envelopes to assign the range of influence of the bone over the skin.

To use the tool, select the skeleton, and then click on the Attach Skin To Bone tool. You must now select which bone you want to attach the object too, and with the bone highlighted, you then click on the mesh you want to assign to that bone. At this stage, no vertices are assigned to the bone, so if you use Dynapose to manipulate the skeleton, the attached object will not move at all (you can however use the regular move tools to move skeleton and skin together in the scene).

In most cases you will want to repeat this procedure for the same skin, selecting another bone, and using Attach Skin To Bone to attach the same mesh to that new bone.

Once you have attached the skin to all the bones you want it associated with, you then would use the Weight Paint tool or adjust the Bone Envelopes to assign the vertices to the bones. Once that is done, the bones will then influence the mesh when manipulated using Dynapose.

This tool would let you create a character with all the bones attached, but with no vertices assigned, giving you a "clean slate" to start from when assigning the vertices – using Attach Skin To Skeleton can result in you having to use Weight Paint to remove vertices from one bone, then repaint them to add them to another, while with Attach Skin to Bone you could have just the last step of using Weight Paint to assign vertices to bones.

It also lets you attach more than one mesh to the same skeleton, and define which bones affect that mesh (for example, for clothing on a character, where each piece of clothing need only be attached to some bones in the skeleton).

8.2.3 Controlling How the Skin Reacts

Bone Envelopes 

Bone Envelopes define the vertices that will be affected by a bone, and offer an alternative to using Weight Paint.

With Weight Painting, individual vertices are assigned to bones. However, if the mesh is dynamic, or you edit it later, then vertices may be created or destroyed. This makes it impossible to control the skin using the skeleton using Weight Paint, since the vertices are being created and destroyed, so there is no way of defining which vertices are influenced by which bones.

In such a situation, Bone Envelopes can prove useful, as they define the area of influence of a bone, and are independent of the geometry of the skin – if a vertex on the skin falls inside the envelope for a bon, it s influenced by that bone. This means it does not matter if vertices move, or are created or destroyed, as there is still a way of defining whether that vertex is influenced by a bone.

To use bone envelopes, click on the Bone Envelopes tool and then select a bone. An orange envelope is shown around the bone, defining the range of its influence. You can click on the center of this envelope to scale it larger or smaller, and click on the ends of the envelope to extend and smooth the transition region, or shorten and sharpen it.

Bone Envelopes are allowed to overlap, and indeed this is useful to provide blending between the bones where they meet. Adjusting the transition region of the bones will let you control how the influences of the bones are blended.

After setting the envelope for a bone, you can activate Weight Paint to see how the vertices are assigned to the bone. However, be careful if you choose to use the Weight Paint tool to adjust the way vertices are assigned – if you remove vertices using the Weight Paint tool, you cannot add them back by adjusting the Bone Envelope and must use the Weight Paint tool to add them to the bone again; also if you add vertices to the bone, you cannot remove them by adjusting the Bone Envelope, you must use the Weight Paint tool to remove them again. Note that any vertices not affected by the Weight Paint tool will continue to be controlled by the Bone Envelope.

Note – you can use CTRL and click to select multiple bones at once, so you can display and adjust the Bone Envelopes at the same time.

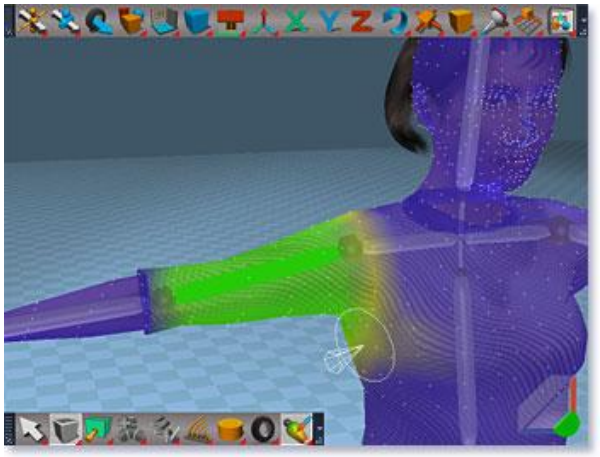
Weight Paint 

A skeleton works by having the bones "pull on" the vertices of the model. Wherever the bone goes, the vertices follow. A vertex can be affected by more than one bone, especially those vertices near joints, which can be affected by bones on either side of the joint.

trueSpace automatically assigns each vertex on your model to a bone (or bones) in the skeleton when you use the Attach Skin To Skeleton tool. However, you will normally want to fine tune this to ensure the skin moves the way you want when the skeleton moves. If you use Attach Skin To Bone, then vertices are not assigned to the bone except using the default bone envelopes.

Whichever method you used to attach the skin to the bone, the Weight Paint tool lets you finely control the amount of influence a bone has on the vertices in the mesh.

You skeleton must be attached to the skin for this tool to work of course. With your skin and attached skeleton selected, choose the Weight Paint tool, and then click on the bone you want to work with. You will see some vertices on the mesh become colored.



The green color represents vertices which are strongly bound to the bone. The color fades through yellow, which are vertices that are pulled less strongly by the bone, through to blue, which are not influenced by the movement of the bone at all.

Assigning vertices to a bone, or removing them from the influence of a bone, is done with a brush tool. The controls for the tool are in the panel on the stack.

Note About SDS – In earlier versions of trueSpace, there was a limitation when attaching a skin with SDS to the skeleton, in that if an SDS layer was removed or SDS weights changed, then the vertex weights were lost and you could no longer use the skeleton with the mesh. This is no longer a hard limitation, and you can now apply SDS before attaching the skin to the skeleton. Afterwards, you can reduce the level of SDS, and weights are retargeted. The recalculations may still cause some data loss as there may be some recalculations to clean up vertex weights.

Heal Skin 

The Heal Skin tool lets you automatically apply a range of functions to tidy up, improve or repair how vertices are assigned to bones. For example, it is useful if you test your mesh and find that a vertex has not been assigned to a bone – rather than find the vertex and paint the weight on manually, you may be able to use the Heal Skin tool to automatically assign that stray vertex to the appropriate bone, based on the weight painting of the vertices around it.

The full range of options for Heal Skin are explored in section 8.4.1.

Convert Skin Weights to Soft Selection 

When in Paint Weight mode, this will create a soft selection of the vertices that matches the weights for the selected bone. Once this is done, you can manipulate this selection by using any of the Point Edit selection tools (Select By Painting, Select By Rectangle, Soft Select, etc). Note that you are still in Weight Paint mode while doing this. Also, you can go straight to using the Point Edit tools without using Convert Skin Weights To Soft Selection, this will swap to those tools without translating the current skin weights.

When done, the Convert Soft Selection To Skin Weights tool will convert the selected points back into skin weights, and a right click will exit the Point Edit select tools and return you to regular Weight Paint mode, with the newly defined weight paints being shown.

Convert Soft Selection to Skin Weights 

This is used to convert point selected using the Point Edit selection tools back into skin weights. The conversion is controlled by the Selection To Skin aspect of the Character Editor panel in the stack (right click on the Build Skeleton tool to open this). See section 8.4.1 for full details of those options. After using this tool, you need to right click to exit Point Edit mode and return to Weight Paint mode. If you right click without using this tool, you return to Weight Paint mode without using the Point Edit selection to make any changes to the skin weights.

8.2.4 Controlling How the Skeleton Behaves

Joint Limits 

Joint Limits let you control the range that a joint can move through. This is necessary to get realistic posing and animation from your skeletons – for example, you would not want your characters knees or elbows to bend backward.

Setting joint limits is best done visually. You can create and change joint limits before or after you have attached the skeleton to the skin – it is often best afterward, since the process gives you visual feedback on how the joints look when moved to their maximum and minimum values, and seeing the model move with the skeleton can be helpful in deciding what range of motion looks realistic and is most appropriate for your character.

Setting Joint Limits for the shoulder.

First choose the Joint Limits tool, and then select the joint you wish to work with. In this example, we are working with the Lucinda mesh, and have selected the shoulder joint, as seen to the left.

Useful Tip: Note that you can use CTRL + click to select more than 1 joint at once. This will display the joint limits information for all selected joints in the Panels tab on the stack. You can use this to copy values from one joint to another, say from the left shoulder to the right shoulder, if you need joints to match precisely. Note that even with multiple joints selected, the widgets still only control the joint they are associated with and do not set the values for all selected joints.

Set Pose As Default

It is usually a good idea to use this tool to let trueSpace know what position is the default state for your skeleton and mesh. At some point once you have built the skeleton, either before or after attaching the skin (or even after setting joint limits and assigning vertices to the bones), you should ensure the skeleton is in the pose you want to use as the default, then click this tool.

At any point when working with this skeleton in the future, you can then simply click the Reset Pose tool to return to the default pose.

Reset Pose

Sometimes when you are done moving your skeleton into different poses, you want to return to the start, to an "un-posed" state. Clicking this tool will reset the skeleton to its default state. The default will be the pose saved using Set Pose As Default (if Set Pose As Default has not been used, then the default pose will be the same as when the skin was attached to the bones).

About Locks

Locks allow you to "freeze" a joint and prevent it from moving. Normally when you pull on the arm of the skeleton for instance, the whole body will follow through on that motion, so not only do you move the arm, but you pull the whole character over to one side

A lock lets you stop the motion from carrying through the joint and into the rest of the skeleton. Then when you pull on the arm of the character, the arm will move, but the body of the character will stay in place. This lets you pose or animate the arm without affecting the rest of the body.

There are two steps to using a lock with Dynapose. The first step is to place it on the skeleton, and the second is to activate and deactivate it as you need it. For example, during the rigging stage you might create locks at the hips, at the shoulders, and at the neck. Later during the animation stage you would activate the locks to be able to pose the skeleton in the way you want – for example, enabling the neck lock so you can turn the head without the body following, or enabling both hip locks to bend the character forward, and so on.

The other way to use Locks is in conjunction with IK Handles to let you can set up controls to instantly carry out particular actions like posing just the left arm, or making the character crouch. Locks are also used to define IK groups or chains that work with IK Handles. You will find tutorials on these subjects in the relevant section of this chapter.

Locks can be assigned a Tag – this is effectively a name for the Lock, so that IK Handles when used with Limbs can recognize the Lock even if it is outside the Limb itself. Locks can also be stored inside a Limb in a Limb Library.

trueSpace has three kinds of locks that you can position on your skeleton, and these are listed below.

Full Lock

This lock is likely to be the one you use most often. It locks the joint both in position, and in rotation, so is basically a combination of both the other kinds of lock. It also starts a fully locked branch so that all joints behind it will not move or rotate either.

Position Lock

Position Lock freezes the joint in position only, but does not stop it (and other parts of the skeleton after it) from being affected by rotation.

Rotation Lock

A Rotation Lock freezes the joint from rotating, but allows it (and other joints beyond it) to still change position.

A Rotation Lock also specifies start and stop bones for an IK group. The exact way this is done depends on the "Invert IK Groups" parameter for the Handle controlling the IK group. When in normal mode ("Invert IK Groups" is unchecked), then the first Rotation Lock ends the IK group, and a second Rotation Lock starts a new IK group (which will be associated with the same Handle). A third Rotation Lock will end that IK group, and a fourth will start a new IK group, and so on.

For details on setting up IK groups using Locks, see the tutorial later in this chapter.

Add IK Handle

Add IK Handle lets you attach an IK Handle onto a particular bone in the skeleton. This lets you set up IK Groups to speed up posing and to allow IK interpolation during animation. Once an IK Handle has been placed, you can click and drag on it to manipulate the skeleton to achieve particular movements. Usually an IK Handle is associated with a particular set of Locks. This lets you create IK Handles for specific purposes, such as for posing the left arm only without moving the entire character, etc.

IK Handles can also be stored inside a Limb in a Limb Library.

Align IK Handle

Align IK Handle lets you set which bone is used to define the alignment of an IK Handle. By default when you add an IK Handle, it is aligned to the skeleton space. You may wish to change this – for example, if you are creating handles that control the fingers, you want them to pose relative to the palm of the hand, so that up, left, right, etc are relative to the palm rather than relative to the whole skeleton.

This can be achieved by using the Align IK Handle to the desired bone. Simply select this tool, select the IK Handle you want to align, then click on the bone you want it aligned to. The IK Handle will change orientation to match its new alignment.

If you aligned the finger IK Handles to the palm, for instance, then the Handle will always cause movement and rotation relative to the palm, no matter what orientation the overall skeleton is in, giving you the control you need over the fingers.

8.3 The Tutorials

8.3.1 Character Rigging Tutorial, Building the Skeleton

We will be creating a skeleton for this female character who we have named Lucinda. The model was provided by aXYZ Models and was imported using OBJ format, and has one layer of SDS applied.

8.3.1.1 Starting Point

We start with the character in the “T Pose” that is most commonly used to build a skeleton for a model, as seen on the left.

We select the model using the Select Object tool, then we click on the Build Skeleton tool.

8.3.1.2 Adding Joints

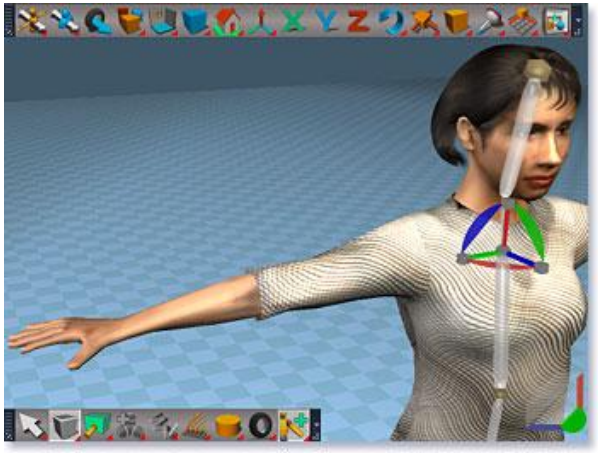
In this case the skeleton has been at the waist and we build upward. The first click adds the bottom joint, and on the second click another joint is added and the bone connecting them appears, as seen above on the left.

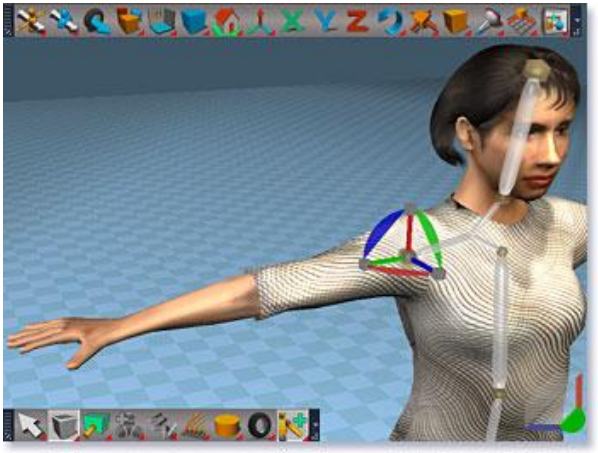
Keep clicking where you want to add the joints, placed where you need your character to bend. The bones are added automatically connecting those joints together. In the picture above and on the right, the spine has been completed with the placing of the last "joint" on top of the head - this will not act as a joint since it has no bone above it, but it will act as a point for us to interact with the skeleton, and is necessary to create the bone to which the head will attach.

8.3.1.3 Creating Branches

Most skeletons have branches in them (unless you are creating a snake!). Creating a branch is simple.

In this example, start by creating the right arm of your character, for which you need to create a branch coming off the spine. First, adjust your view ready to make the branch.





Next, click on the joint that you want the branch to start from. Note that for better control, do not select the joint in the neck, as that joint is for moving the head, so select the joint below that, as seen in the picture above on the left.

From this point on, it is the same as building the first part of the skeleton, simply click where you want each joint to go, starting with one for the shoulder itself, shown above on the right.

You can continue creating branches by the same process of clicking the joint you want the branch to start from, then clicking to place the new joints that are to make the branch.

For the left arm, select the same joint you used to branch off for the right arm, and then continue to place the joints for the left arm.

8.3.1.4 Shape The Skeleton

Now that you have built a substantial part of the skeleton, you may want to pause in building it and tidy up some of what you have created. You can see that trueSpace has done a good job of placing the joints inside the skeleton, but since the perspective view has been used, the joints are not as neatly positioned as they could be.

Here the orthogonal views (Front, Side, Top) are useful for tidying up the skeleton. Once you have the desired view selected, click on the Shape Skeleton tool. You can move between Build Skeleton and Shape Skeleton at any time. Build Skeleton lets you add new joints, while Shape Skeleton lets you move existing joints.

In just a few seconds, you can have the skeleton laid out the way you want, changing it from something like the picture on the left to something like the picture on the right. You can use the information panels to position joints accurately (though in these example pictures I have done this purely by eye).

8.3.1.5 Continue Building

Now select the Build Skeleton tool once more and continue building the skeleton by adding the legs, creating branches just as you did for the arms, as seen above left.

It will not take you long to have a completed skeleton, similar to the one seen above on the right. It is up to you how many joints you use and where you position them – the sample skeleton here is probably the most basic you can use – for example it does not build joints for the fingers, though you can do that if you wish. Make sure you build a skeleton that is complex enough to achieve the range of motions you need, but as simple as it can be too so it is easy to work with. Just how many joints you create will depend on what you need from that skeleton!

8.3.1.6 Attach Skin To Skeleton

The last step at this point is to join the skin and the skeleton together. With the skeleton selected, choose the "Attach Skin To Skeleton" tool. When you point at the mesh, you will find it highlights in a light green color, showing which object will be selected as the skin. Click once when the skin is highlighted. When the skin is attached to the skeleton, the highlight will change color to show the process is complete – this can take several seconds depending on the complexity of the skeleton and the complexity of the skin.

You can review all these steps in building the skeleton and attaching it to the mesh in the video below:

video link

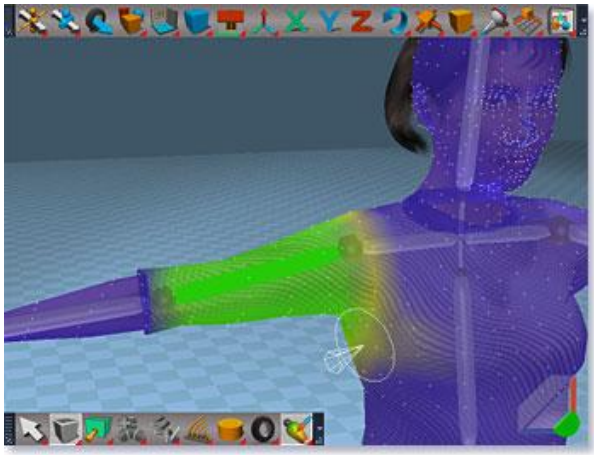
This completes the first stage of creating your character, and you can now pose them using the Dynapose tool! In some cases, this may be all you need to do. Usually though you will want to go on to assign the vertices to the bone, and to assign limits to the joints. Finally you might also want to create some IK Handles for easier posing. These topics are covered in the tutorials that follow.

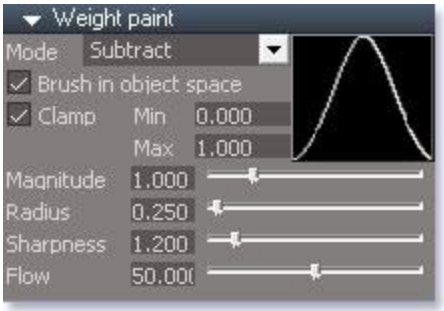
8.3.2 Weight Painting Tutorial

It is important to assign which vertices on the mesh will be controlled by which bones in the skeleton. In over-simplified terms, each bone on the skeleton will "pull" on vertices in the mesh, so that they follow the movement of the bone. It is possible for the same vertex to be influenced by more than one bone, especially around joint areas.

While trueSpace assigns vertices to bones automatically, you will usually want to make some adjustments to this. One common area that needs tweaking is around the shoulder, where some vertices on the side of the body become assigned to the bone in the arm. This can make the side of the body crumple or expand as the arm moves up and down. This tutorial will go through how you remedy that with the Lucinda mesh example.

Ensure you have the character with skeleton selected, and then click on the Weight Paint tool, and the skeleton will appear. Now click on the bone that you want to adjust, in this case, the top bone in the character's right arm.

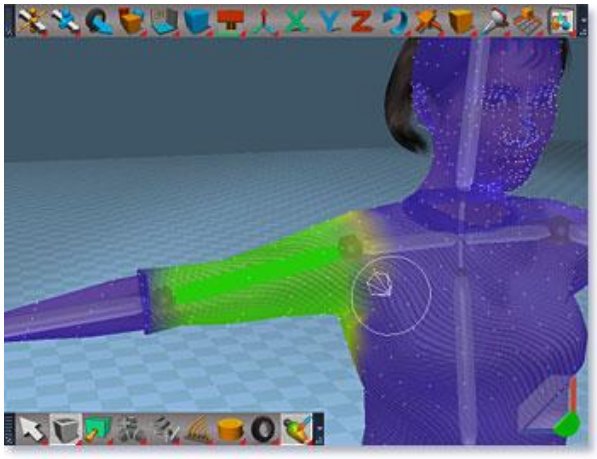


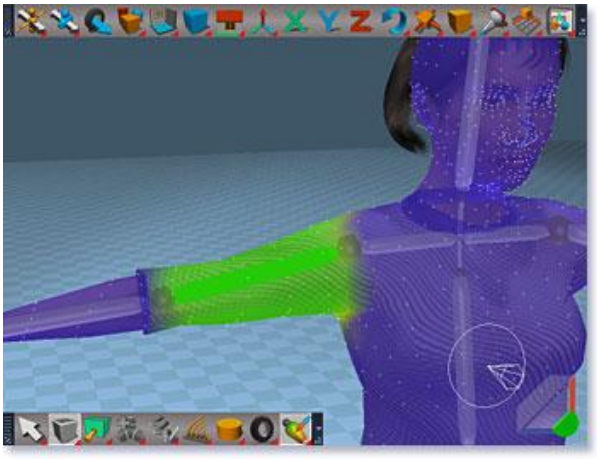


As seen in the above image on the left a brush tool appears, and the vertices on the mesh become highlighted with colors. Green means the vertex is closely associated with that bone, so will be pulled strongly along with it. Yellow means it is influenced by the bone, but not so strongly. Blue means it is not influenced by the bone at all.

As is common around areas where limbs attach to the main body, some of the main body vertices are being influenced by the bone in the limb, as you can see by the yellow area on the character's side.

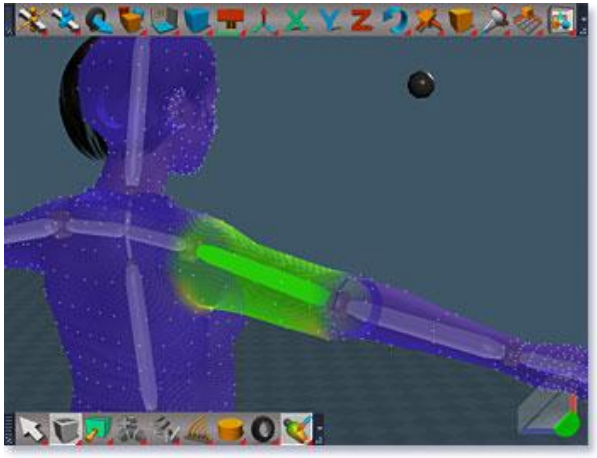
Change the Weight Paint brush so that it will Subtract vertices from being influenced by the bone. You may also want to decrease the default size of the brush for finer control – the panel above on the right shows the settings used throughout this example.





Click with the left mouse button and begin to paint – since you are using the Subtract setting on the brush, the vertices change from Green to Yellow to Blue as they are removed from the influence of this particular bone.

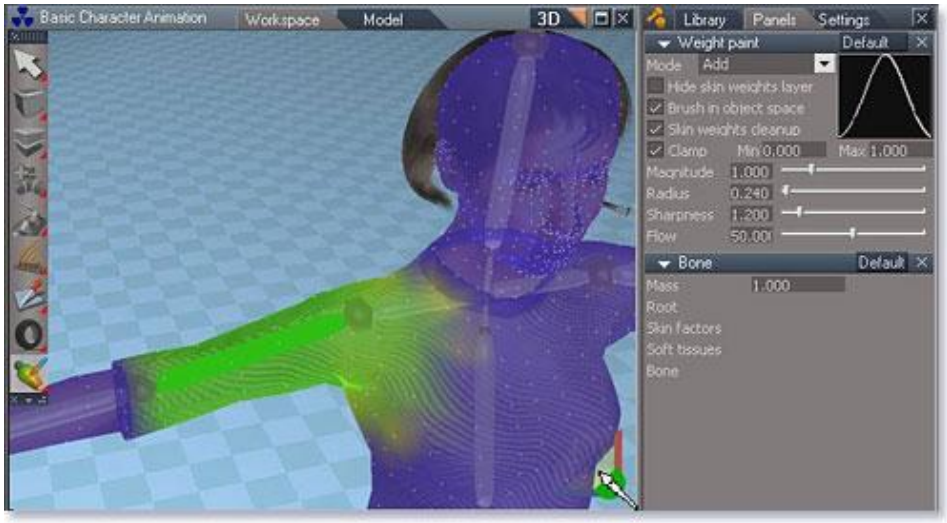
As you keep painting, you will end up with moving from something like the first image in this section, to the picture above on the left, and finally to the one on the right. You can adjust the Magnitude of the brush so that there is a smoother transition between arm and shoulder (more yellow vertices) than seen here, it is up to you depending on how you want the model to bend at this joint.



Now rotate around to the back of the character and ensure the vertices there are correctly assigned. The picture above shows what you will end up with if you subtract vertices from the influence of the bone there too.

You may need to swap the brush to Add mode, and check that the vertices on the side and back of the character are fully assigned to the bone in the spine, now that you have removed them from being assigned to the arm.

You can view weight painting in action in the video below



video link

It can be useful to move the skeleton while weight painting, so you can find and fix any vertices that have not been properly assigned. This can be done using Dynapose (or IK Handles if you have any already set up for the skeleton). One important thing to remember is to record the default pose for your skeleton so you can return it to the default state after moving it to find stray vertices.

This gives the workflow of using Set Default Pose, beginning weight painting and assigning the vertices, moving the skeleton to pose the character and check all vertices move as expected, using weight paint on the skeleton while it is posed so you can easily paint onto vertices that are incorrectly assigned, then clicking Default Pose to return your character to the default state. You can activate weight painting at any time of course, if you later find any stray vertices, but deliberately moving your model to test it and fix issues at this early stage is a good idea, to make working with your model easier when you reach the stage of animating and posing it.

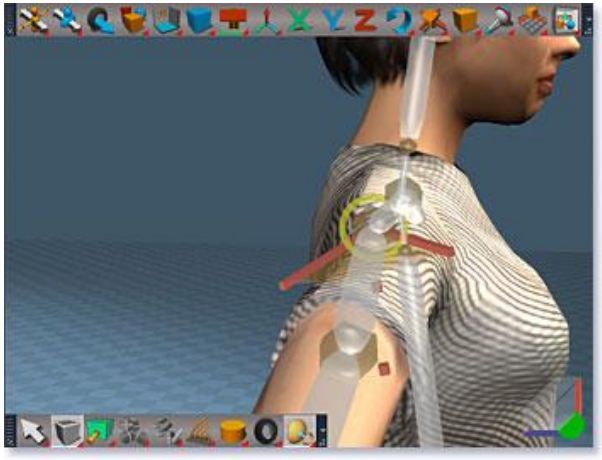
8.3.3 Joint Limits Tutorial

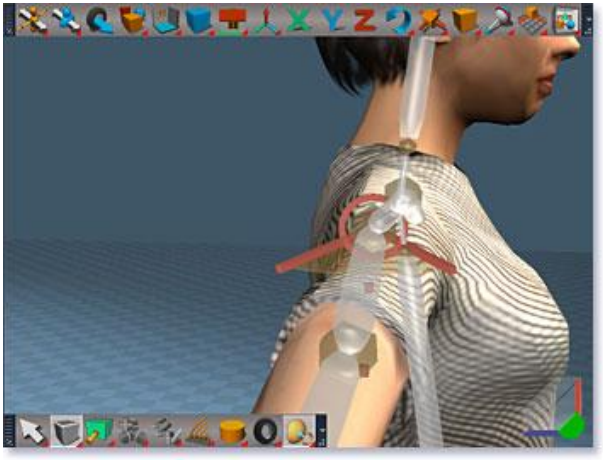
One important thing about a real world skeleton is that certain joints only move so far in certain directions. You can't twist your head around through 360 degrees for example, or lift your leg upward until your toes touch your forehead.

To make your character move realistically in trueSpace, you will need to set limits for the joints that simulate those you would have in the real world. This tutorial will look at doing that for the shoulder joint on the Lucinda model.

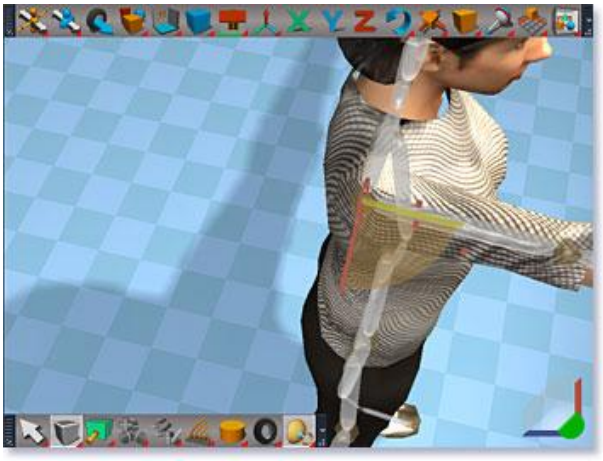
Ensure your character is selected ,and then choose the Joint Limits tool. The skeleton will become visible, and you can click on a joint, in this case the shoulder joint.

When you do, a 3D widget appears that lets you view and adjust the limits for this joint, as seen in the picture on the left.





First, we want to make sure that the movement range of the joint is aligned with the mesh. A quick change of viewpoint will give you the image on the left, and you can see that the forward / backward motion of the joint also moves the arm up and down a little. Using the circular part of the widget, you can adjust that so that the forward / backward motion lies in the expected plane. The adjustment is seen on the image on the right.



Now move to a more overhead view so you can see the range of forward and backward motion for the arm. Clicking on the edges of the visible limit lets you move that edge of the limit, and as you adjust the limit, you will see the mesh move in real-time, as well as seeing the range displayed by the widget change.

In the above images, on the left, you see that the arm is adjusted so that it can no longer bend backward. On the right, you see what happens when you adjust the other limit of the joint, letting the arm move until it is pointing forward. Notice how both arm and widget update immediately as you move the mouse to show the range of motion you have set.

Now you should adjust the up and down range for the shoulder. Move your view to get a good vantage point, and then click on the small cube in the 3D widget. Initially the joint has no up and down motion at all as seen in the left image, until you click and drag on that cube, which gives you an image similar to the one on the right.

Note that the arm moves in real-time again so you can see how it looks, and also that the 3D widget updates. It now shows a range through which the joint can rotate back and forth, and up and down. There are now four edges that control that range of motion.

To adjust how far down the joint can bend, grab the lower edge of the joint limit as seen in the picture on the left. Click and drag down and again both arm and widget update to show the range of motion you are allowing for this joint.

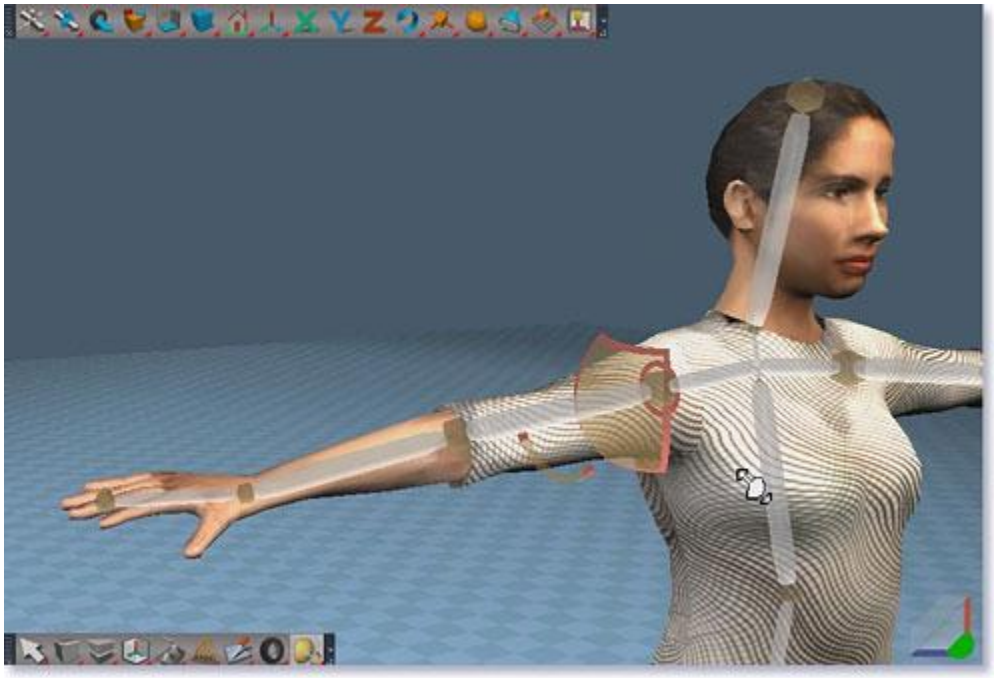
The shoulder is a complex joint, and we are not done yet! It has a little rotation around its own axis too, so let's add that to our joint limits. The picture above shows the cube highlighted for setting these final limits – again the default is that there is no rotation allowed, so just a cube is shown and no range of possible motion.

After adjusting the rotation in one direction, our widget updates to show a range of rotation using an arc with a control handle at both ends, as seen on the left above. Grabbing the other control handle lets you adjust the rotation limit in the other direction, as seen on the right.

And now your joint is set to behave as you would expect it too! The image on the right shows a close-up of the limits set for the shoulder joint, and you can see how the widget gives you a clear indication of the range of motion that this joint will allow.

Many other joints will only require 1 or 2 degrees of freedom, and will not be as complex to set up as the shoulder joint – once you can tackle a joint like the shoulder joint, the rest will seem simple!

You can review how to set Joint Limits in the video below:



video link

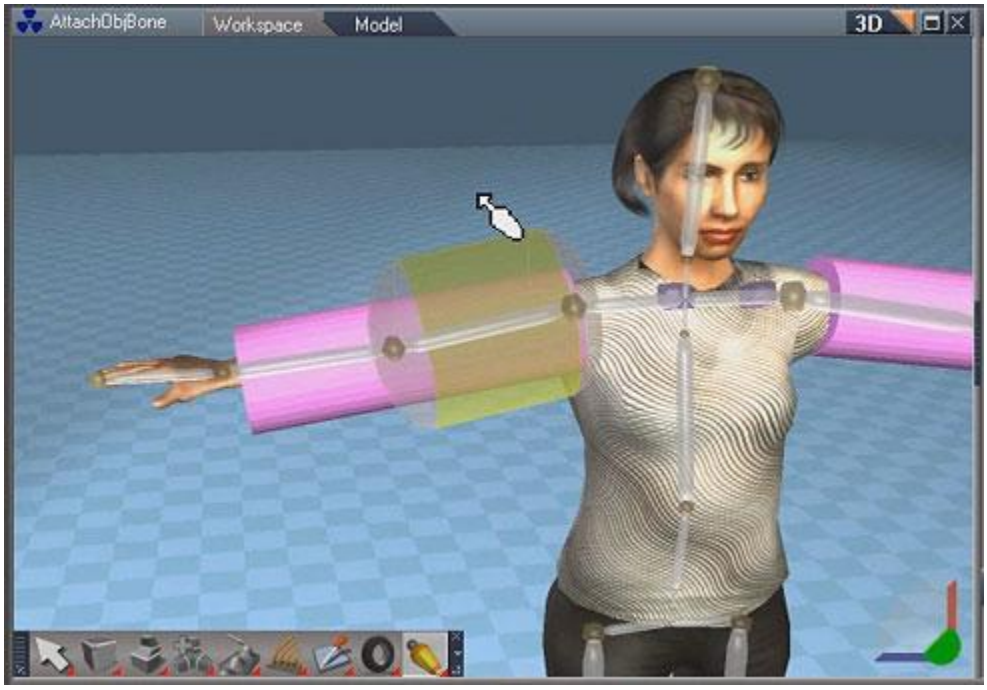
8.3.4 Attach Skin to Bone

As well as attaching skin to the whole skeleton, it is possible to attach skin to one bone or a subset of bones. This is useful for attaching clothing or items that are not part of the whole body.

Begin by selecting the Attach Skin To Bone tool, and then select the bone that you want to attach the object too. Next, click on the object, and wait until it has been attached. If you want to attach the same object to another bone, then simply select the next bone, and click on the same object again. Repeat this process until you have the object controlled by all the relevant bones. Also note that you can attach more than one object to the same bone too.

Once attached you will most likely need to use Bone Envelopes or Weight Painting to assign the vertices in the object to the relevant bones.

You can view the process of attaching objects to individual bones in the following video tutorial:



video link

8.3.5 IK Handles and Locks, Basic Tutorial

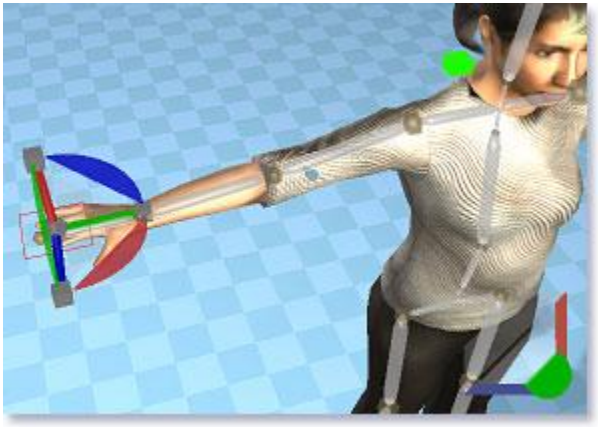
IK Handles let you set up specific rigs to more easily pose and control your characters. There are endless possibilities in how you might choose to set those up, and this tutorial will take you through the basics of how to set up an IK Handle and use it with a single Lock.

We are going to make an IK Handle that lets you move the arm of a character while keeping the rest of the body completely still. This could be useful when you want to tweak or adjust a pose by fine tuning the arm without repositioning the rest of the body.

The first step is to add a Full Lock onto the shoulder of the character. As with all Locks, you can enable and disable this yourself for use with the Dynapose tool – select Dynapose, click on the Lock to enable it (it will turn green), and then drag the hand of the character, and the arm will move while the rest of the body stays still.

Using Dynapose with a lock so only the arm moves – note the lock is green showing it is active in Dynapose.

This is fine, but what if we want quick access to that sort of posing, without having to enable and disable the Lock ourselves? We want to define a way of moving the arm where the shoulder Lock is always on, and we can do that with an IK Handle.

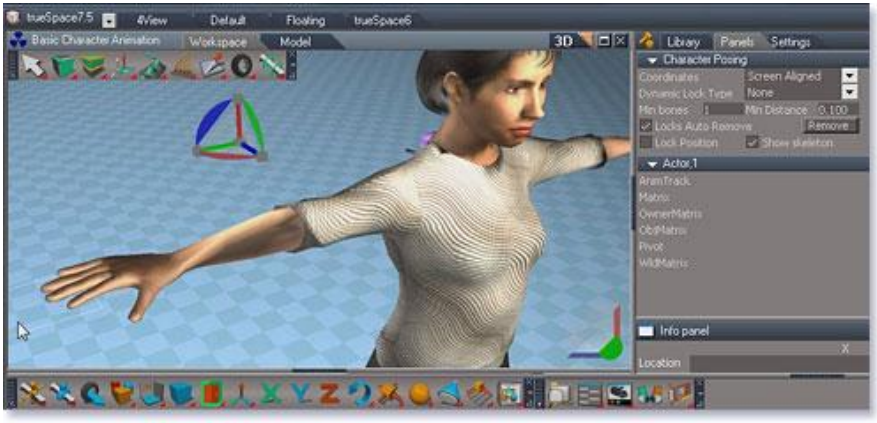


Use the Add Handle tool and add a Handle to the bone in the character's hand. With the Handle still selected, click on the Full Lock in the shoulder to enable it. This activates the Lock for this Handle – every time you use this Handle to pose the character, this Lock will be treated as active, as seen in the image above (the IK Handle has been changed to red from the default yellow, for easier viewing in the manual).

You can try this for yourself. Exit the character editing mode so that you are back to having your character selected. Now try dragging on the IK Handle – first, note that Dynapose does not need to be active for this to work; next notice that even though the Lock is not currently enabled (as seen in the image above, note that the lock is not green this time, showing it is currently disabled), when you use the IK Handle, you move the arm just as if the Lock was enabled.

If you activate Dynapose, you will find that you can drag the arm without using the IK Handle, and it moves normally with the whole character following the motion; drag the arm using the IK Handle, and again the movement is the same as when the Lock is enabled.

You can see the steps involved in creating an IK Handle and setting it to work with a particular Lock in the video below, which also covers the difference between a regular and a firm IK Handle:



video link

This barely scratches the surface of what you can do with Locks and IK Handles, but it does give you a simple illustration of the principles at work. Next we will look at using multiple Locks with one Handle to create even more specific and complex ways of posing our character.

Note that you might want to assign a Tag to any Locks you are using. While not necessary for the functionality seen in this tutorial, it can be useful later if you choose to store Limbs in a Limb Library. By setting a Tag for a Lock, that Lock can then be referenced by an IK Handle in a Limb even if the Lock is outside that Limb.

8.3.6 IK Handles and Locks, Advanced Tutorial – IK Groups

In the previous tutorial, you saw how an IK Handle lets you manipulate a character, and can have particular Locks associated with it. In that example, the IK Handle moved only the arm, and left the rest of the body frozen as the movement from the IK Handle was terminated with a Full Lock, preventing both rotation and movement of any joint after that Full Lock.

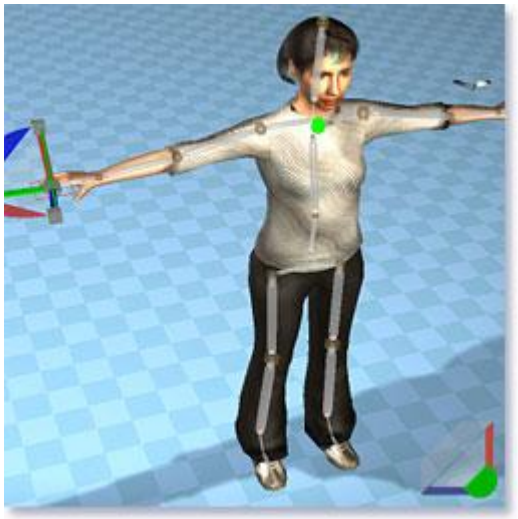
We can create more complex behaviors than that though, if we add in more Locks to define some IK groups. And IK group is similar to defining a set of joints as being “one big bone” – that is, the whole group may rotate at the ends, but the joints inside the IK group cannot rotate independently.

An important thing to note is that IK groups are created using Rotation Locks only.

If you use multiple Position Locks associated with an IK Handle, then this creates multiple nails that freeze those points of the skeleton in their location. There are no IK groups defined in between the Position Locks, they are only locking the position of the joint or bone they are on.

Multiple Full Locks have no effect, as a Full Lock prevents all rotation and movement from that point on in the IK structure, so a Full Lock terminates the effect of the IK Handle.

To illustrate how IK groups work, we are going to make an IK Handle which keeps the character’s back straight when we move the arm. Normally, the joints in the back will bend independently causing the back to bend, but by creating an IK group that contains all those joints, we can prevent the joints from bending independently. This will keep the back straight, effectively making it act as if the back was created from just one bone with no independent joints in it.

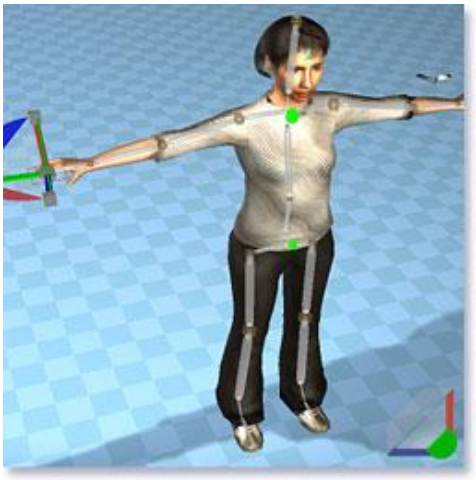


The IK Handle with associated Rotation Lock.

First, add a Rotation Lock onto the shoulders, and then add an IK Handle onto the bone in the hand of the character. Enable that Rotation Lock with that IK Handle, and you will have something like the image above. This ends the first IK group, giving you a chain of joints running from the hand to the shoulder, and all those joints are free to move and rotate.

Moving the Handle – note how the arm bends, the character stays straight, and the whole character moves forward.

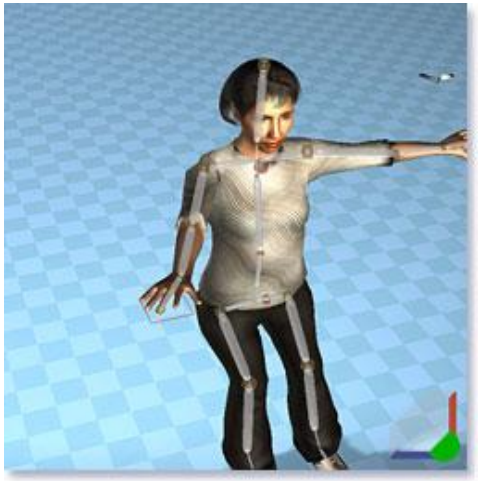
With just this one Rotation Lock in place, all the other joints in the skeleton after the Lock are prevented from rotating, but are still free to move. Try using the IK handle at this point in the set up, and you will see the arm bends and moves freely, and the rest of the body stays rigid (no rotation) but does move to follow the movement of the arm, as seen in the image above.



The second Rotation Lock creates an IK group from the shoulders to the hips.

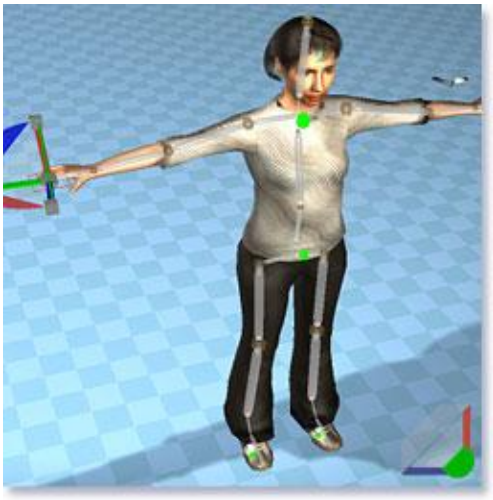
Now add another Rotation Lock on the hips. Note that if you have exited the editing mode for the IK Handle, adding another Lock to the Handle just needs you to edit the IK Handle information again by using the Shape Skeleton or Add Handle tool, and then click on the existing Handle you wish to edit. Your scene should look something similar to the image above.

Adding this second Rotation Lock to the IK Handle ends the second IK group, which extends from the shoulders to the hips. This group is locked from rotating, because of the Rotation Lock that starts it up at the shoulder, but with this new Rotation Lock in place, the legs become free to rotate again – the second Rotation Lock ends the effect of the rotation being blocked.



The legs are now no longer Rotation Locked and are free to bend again when the Handle is used.

Try the IK Handle at this point in the set up. Now you will find that the legs bend as you move the character around, while previously they were rigid, but the back still remains rigid and does not bend, thanks to the effect of the Rotation Lock. Also, note how the head and other arm do not bend, they too are still affected by the first Rotation Lock (since there is no second Rotation Lock along those branches to end the effect of that first Lock). The end movement might be something like the above picture.



Two Full Locks are added to the feet.

Finally, add a Full Lock onto each foot. These Full Locks now create ends for two other IK groups, one for each leg, running from the hips down to the feet.

Now the feet stay locked in place.

Now when you use the IK handle, the feet stay locked in place, completing the effect we wanted for this IK Handle.

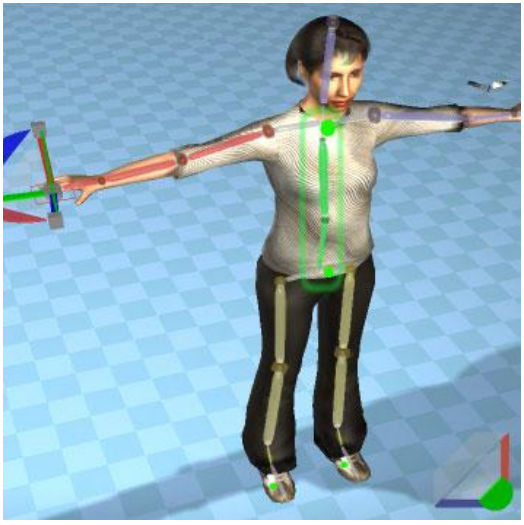


Image highlighting the IK groups by colorizing them.

Let's review what you are seeing when you move the character using the IK handle – the above diagram has the various groups of bones and joints colorized to highlight them.

The first IK group runs from the Handle on the hand, up to the shoulder (colored red in the above image). All the joints in this group are free to move and rotate.

The next IK group runs from the shoulder, down the back, to the hips (colored green in the image above and circled). This is our main IK group, defined by the two Rotation Locks. All the joints in this group are allowed to move, but not allowed to rotate, keeping the back straight.

Then we have two more identical groups, running from the hips down to the feet (colored light yellow in the above image). These groups are free to move and rotate (since the last Rotation Lock at the hips ended the effect of the Rotation Lock at the shoulder). The groups are terminated by the Full Locks on the feet, which keep those bones pinned in place and unable to rotate.

Finally, we have the remaining groups (colored blue in the above image) which are still Rotation Locked from the first Rotation Lock. These groups run to the end of those chains since they do not have a Rotation Lock to terminate them.

When editing the IK Handle, you can choose to reverse the groups that are locked and which are free to move by using the "Invert IK Groups" parameter. When checked, then the joints and bones that were Rotation Locked will become unlocked and free to rotate, and those that were free to rotate will become locked.

In the example you have just set up, if you check “Invert IK Groups” on the IK Handle, then the joints from the hand to the shoulder will be rotation locked and so the arm will not bend. The back (and the head and other arm) which were previously locked from bending will now be free to bend. The joints in the legs were previously free to rotate, so those now become locked and the legs will not bend. The Full Locks on the feet keep the feet pinned in place still.

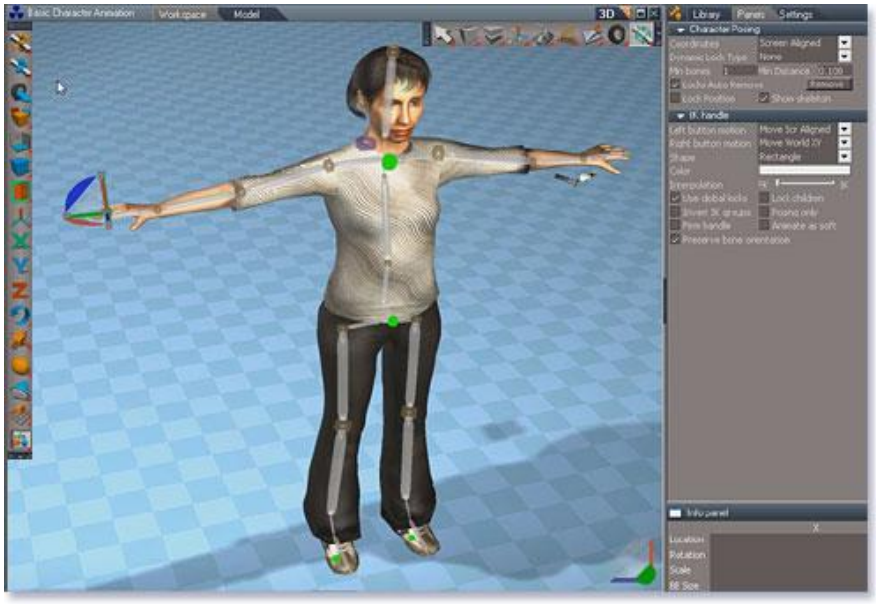
With that checked, the IK Handle will now keep the arm rigid, and make the character bend at the waits, with the head and other arm following that movement. The legs and feet will remain totally static.

As you can see, you can create very varied results for an IK Handle depending on how you set it up, allowing you to have whatever kind of control you want over your skeleton. You can of course have more than one IK Handle, each set up to use different combinations of Locks, so that you can instantly access different ways of manipulating your character.

You can use the Co-ordinate Lock settings for Rotation Locks to disable for each axis (x, y and z) separately. When you disable all axes (all Co-ordinate Locks are checked), then the Rotation Lock will behave only as a start / stop marker for defining IK groups. This means that if you check all the Co-ordinate Locks for the Rotation Lock at the shoulder blade in our example, then you will enable rotation of the body of the character, but it will rotate as one group, rather than each bone within that group rotating separately.

You can also selectively enable rotation, so that bending in certain directions is allowed – for example in the image seen above, the top Rotation Lock on the shoulder has the Y direction unlocked, allowing the character to bend from side to side down their back when the IK Handle is used, but preventing other rotations for the back.

You can view the process of setting up an IK Chain in the video below:



video link

8.3.7 Controlling Skeleton Visibility

To make it easier to work – or if you want to create a final render from the real-time view – it is useful to be able to control whether or not you can see the skeleton.

Let's start by controlling the visibility when you are not using Dynapose to work with the character. This affects the visibility of the skeleton when using the real-time view to produce renders, for example, and can avoid the situation where you can see joints or bones "sticking out" of the character.

Notice how you can see the IK Handle on the left of the image, and the ends of the skeleton.

Ensure you are viewing the Panels tab in the stack , and then right click on the Build Skeleton or Shape Skeleton tools, and. This opens the Character Editor Preferences panel. Select the Default aspect of this panel. In here, turning Joints & Bones to Hide will prevent the skeleton from showing.

Setting Rig to 'Hide' affects bones, joints, handles and locks.

You can also control visibility for Locks, and for the whole Rig (hides Joints, Bones, IK Handles and Locks all at once)., and for the IK Handles.

If you click on Dynapose, then the skeleton will become visible again. This means you can see and use Locks, IK Handles, etc, while posing the character, but keeps them hidden while you are not posing the character. You may also want to hide the skeleton while posing the character though, so you can concentrate on how the character looks without the distraction of seeing the skeleton.

This is a separate view control, found in the Panels tab of the stack whenever Dynapose is active (you can right click on the Dynapose tool to open the panel if you have closed it previously). Here there is a checkbox for Show Skeleton – if this is checked, then the skeleton and rig (IK Handles and Locks included) will be visible while Dynapose is active, whether or not they are visible when Dynapose is not active. This is an ideal setting for keeping the skeleton hidden until you need it – hide the Rig in the Character Editor Preferences panel, and ensure Show Skeleton is checked in the Character Posing panel.

Note that if you enable showing individual features such as Locks or IK Handles using the Character Editor Preferences panel, then these are always visible whether Dynapose is active or not, even when Show Skeleton is not checked under the Dynapose Character Posing panel.

You can watch how to show and hide the skeleton in the video below:

video link

8.3.8 Working with Limbs

A limb is any branch of the skeleton that comes off of the main body, obvious examples being an arm or a leg. In the case of an arm or a leg, note that the limb would include any fingers and toes on it, so the limb can itself contain branches. If you choose, you could also treat a finger as a limb.

Working with limbs is very easy in trueSpace with very little set up require – you simply need to name the joints where limbs are to be attached to specify what sort of limb should go onto that joint, and you need to ensure to define whether that joint is for a Left or Right limb.

Once created, limbs can be dragged from one part of the skeleton to another, or to and from a limbs library. Note that a limb will copy all Joint Limit information for the joints it contains. It will also copy all IK Handles and Locks that are attached to it. IK Handles stored with the Limb can also reference Locks outside the Limb, so long as the external Lock has a matching Tag (for example, a Lock with a Tag of “Shoulder” could be referenced by an IK Handle in an Arm).

Let us explore some uses of limbs below.

8.3.8.1 Mirroring a Skeleton

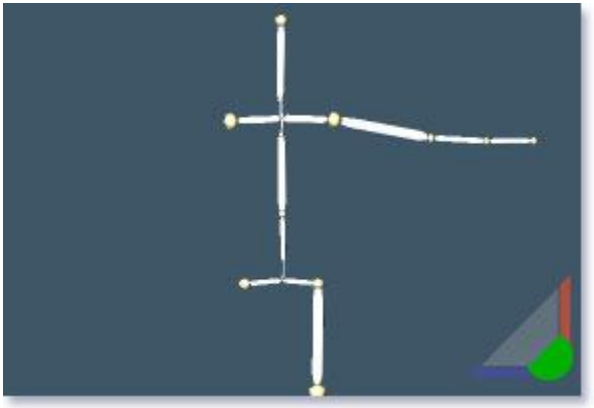
One of the primary uses of limbs is to allow you to mirror a skeleton when building it.

Begin creating a skeleton as you have done in the previous tutorials, but this time, create only the left arm and left leg. You will need to create the right shoulder joint and the right hip, coming off the spine of the character, to define the position where the other arm and leg will attach to, but do not create anything more than that.

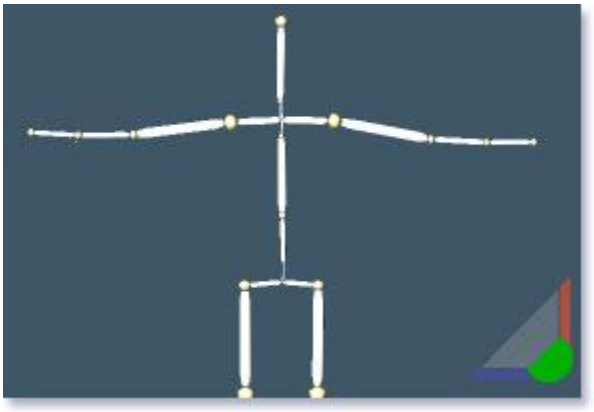
Now set up your Joint Limits for your arm and leg. You will have to manually set up the Joint Limits for the two shoulders and the two hips – since these joints are contained in the base skeleton and not in the limb, the Joint Limits for those will not be copied along with the limb.

You might also want to create any Locks that you think will be useful for this limb, though these are also easy to add later so are less important than Joint Limits.

Be sure to select the shoulder and hip joints, and in the Joint panel specify which is Right and which is Left in the Symmetry parameter. This is important to ensure that your limb is mirrored correctly.



Once done, you will have a skeleton similar to the one seen above. Hold CTRL and click on the shoulder joint for the left arm, and then drag over to the right shoulder joint. Release the mouse button over that joint, and the arm is instantly copied. Because you set the Symmetry parameter for the shoulder joints, the arm will be mirrored correctly.



Now repeat the same process for the leg, and you have a complete skeleton as seen above. By using this drag & drop process, you can drastically cut the amount of work required to set up your skeleton, and ensure absolute accuracy in the parameters for each limb.

The important thing to remember is to set the Left and Right symmetry appropriately for the shoulder joints, and to set the Joint Limits for those joints manually to ensure they match. Note that it is not necessary to add a Limb Tag to the joints for this procedure, although it is highly recommended that you always do add a Limb Tag, to ensure your skeleton is as flexible as possible for future use and re-use.

8.3.8.2 Saving Limbs to Libraries

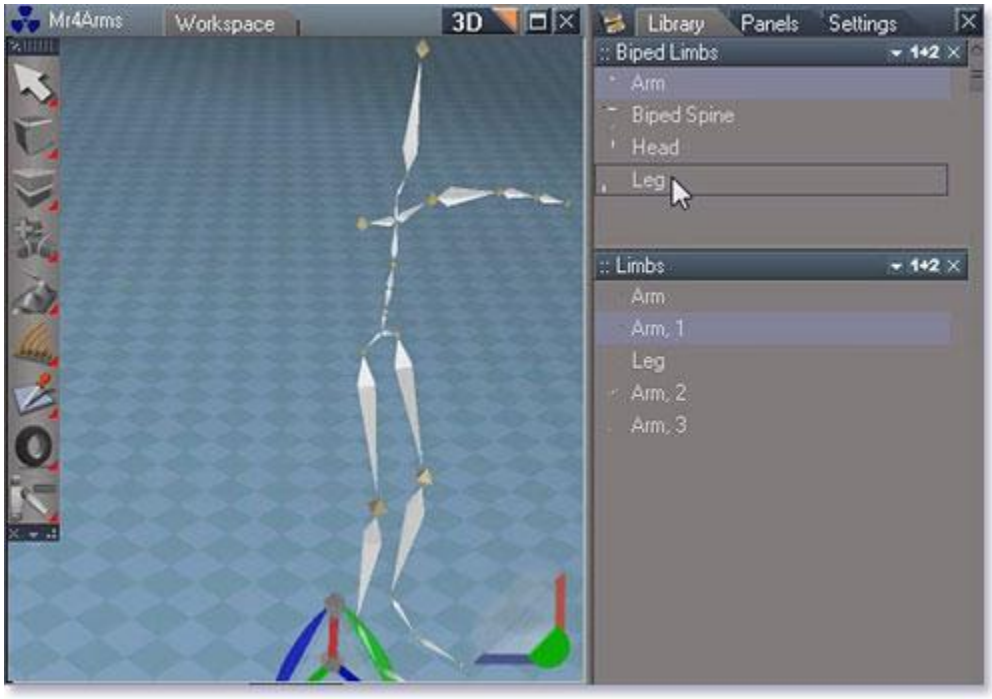
You can go beyond simple copying of a limb from one side of a skeleton to another and instead you can save limbs to a library. This lets you create a new skeleton, create the joints where limbs are to attach, name those joints with an appropriate tag, and then connect limbs from your library onto your skeleton.

This means you could have a library of 10 different legs, and with a double click you could change the legs on the skeleton until you get the ones you want. Your skeleton could have one leg, two, four, or more – the double click will attach a leg to each joint in the skeleton that is expecting to have a leg!

To illustrate, use the skeleton you created in the previous tutorial. Giving your joints a Limb Tag was optional last time, but take a moment to ensure that you have named both shoulder joints as “Arm” (remember you do not name them “Left Arm” and “Right Arm” as the Left and Right is specified in the Symmetry property for the joint).

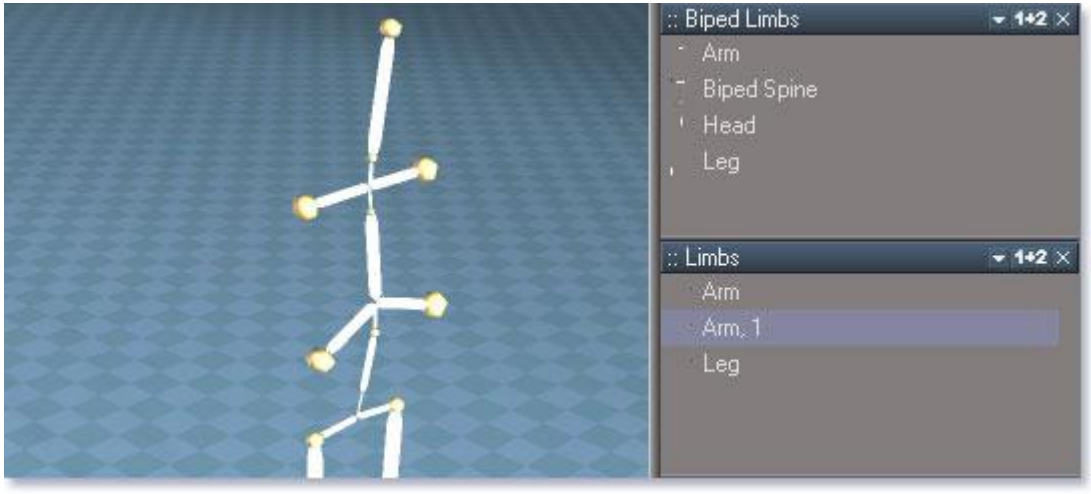
Now create a new Library that should be set to a type of Limb, and make sure the library is open in the Stack. Then in Build Skeleton mode, CTRL and click on the shoulder joint for your arm, and drag it to your library.

Now create a skeleton which has no arms at all, and just the two shoulder joints. Each should have the appropriate Symmetry value, and should have a Limb Tag of “Arm”. Now simply double click your arm in the Limbs library, and you will see that your skeleton instantly gets both a left and right arm, mirrored appropriately.



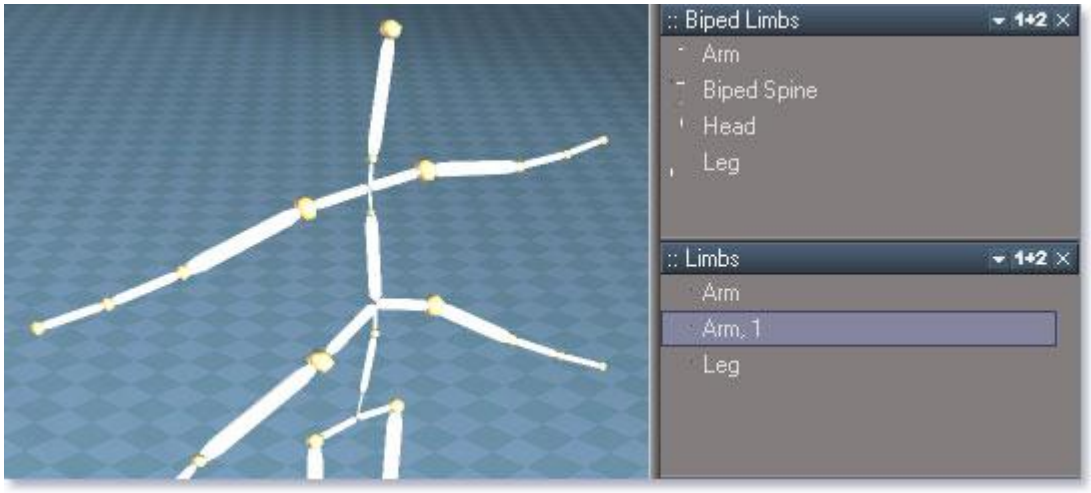
video link

You can watch the Limb Libraries in action in the above video.



A skeleton with four joints created for arms

Now try creating a skeleton with joint locations for four arms. Again, a double click instantly adds all four arms. Using the Build Skeleton tool, edit one of the arms to make it different, and then drag it into the Limbs library. Now double click your new arm, and the other three arms on your object will update to match.



A double click in the Limbs library adds all four arms at once

In this way, you can easily and instantly swap between different limbs for your skeleton, making it easy to "mix and match' the limbs and letting you instantly reuse any favorite set-ups you have.

Important Note – Don't forget that the Limb Tag is case sensitive, so that "Arm" is different from "arm"!

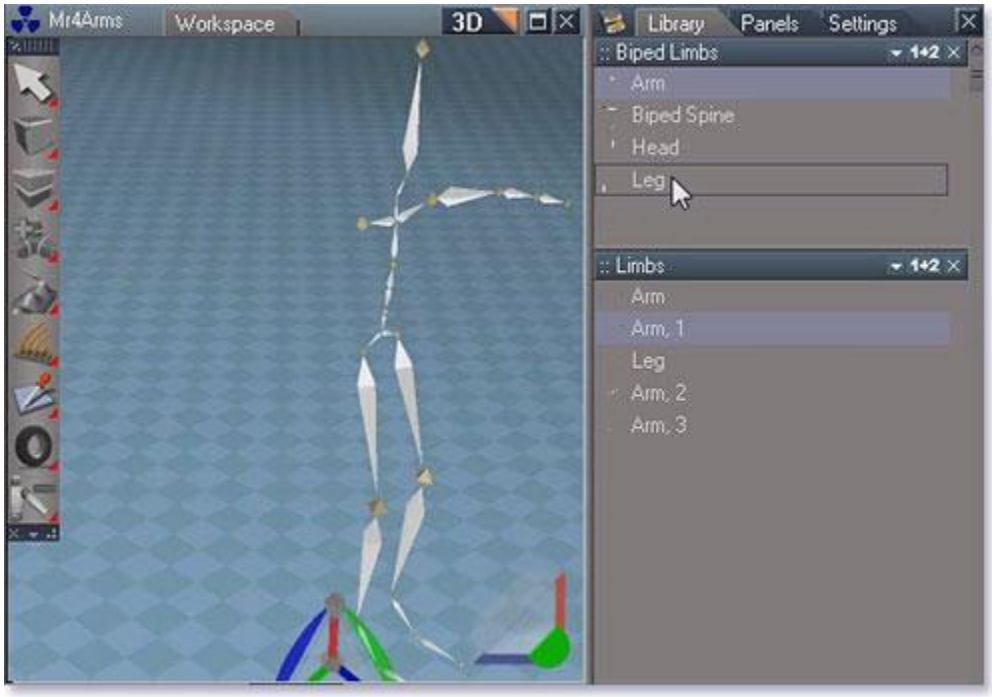
You can see how to use Tags with the Limb Libraries in the video below:

video link

8.3.8.3 More about Limbs and Libraries

You can manually drag a Limb from a library onto any joint on a skeleton, regardless of the Limb Tag that is applied to that joint and the Limb Tag stored in that Limb. Limb Tags are used for double clicking a Limb in the library and having it applied to all relevant joints in the selected skeleton, but the manual method lets you reuse any Limb wherever you like on any skeleton.

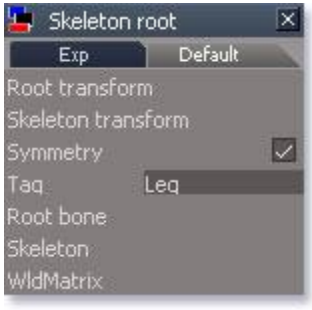
You can also drag a Limb into the workspace, to edit it without attaching it to a skeleton. This could let you adjust a leg or an arm independently, and then save it back to the library, ready for use on your skeletons.



video link

8.3.8.4 Editing the Tag for a Limb

You can rename the tag a limb uses. Load the limb into the scene by dragging it from the library into the workspace. Now in the Link Editor, go inside the Skeleton object for that Limb, and open the Skeleton Root object.



Expand the parameters for that object, and you will see something similar to the above image. Here you can see the Tag for this particular Limb (in the example shown, it is “Leg”). You can change this here to give it a new name, for example if you might rename a Leg to an AlienLeg so that you can attach it from the Limb Library onto joints that have the AlienLeg tag, allowing you to build a collection of legs that are specifically designed for your alien characters.

8.3.8.5 Limb Libraries and Animation

It is possible to swap Limbs even on an animated skeleton. So long as the joint names and hierarchy are the same in each one, you can switch out an arm for another and still have the animation preserved.

If you want to do this, naming joints may help, otherwise when you create a new limb then the start joint name may be different each time, and this would prevent the animation from carrying across. The limb can have new joints, even placed in between joints that were previously animated. However, new branches along the length of the limb can change how the limb is encapsulated and could cause the animation not to be preserved. You can however add new branches on the end of a limb.

As an example workflow, you could start by working with a basic skeleton that had no hands on the ends of the arms. Once you were happy with blocking in the animation for the limbs, you could swap the arms for new ones that do include hands, and go on to work on the animation for the hands. This lets you start simple, with a more basic skeleton, and move on to a more advanced skeleton when required without losing any animation you have already developed.

Note that while you can replace an arm on the skeleton and the skeleton retains the animation, you cannot directly save an arm with animation to a Limb Library and then reload the arm with the animation still attached onto another skeleton. Of course, you can easily save the animation clip to a clip library, and the arm to a limb library, and recombine the animation with any appropriate arm on any skeleton!

8.3.8.6 Limbs for Skeletons with Skin Already Attached

You cannot replace Limbs on a skeleton that already has skin attached. Doing so will result in bones that cannot have vertices assigned to them. You can detach the skin and the skeleton (select the character, use the Attach Skin To Skeleton tool, and click on the attached skin to detach it), and then edit the skeleton using the Limbs Library, and then reattach the skin.

Note that this will reset all the Weight Painting when you reattach the skin. Animation will be preserved however (within the limits of carrying animation across from one Limb to another).

8.4 Character Editor Panels and Settings

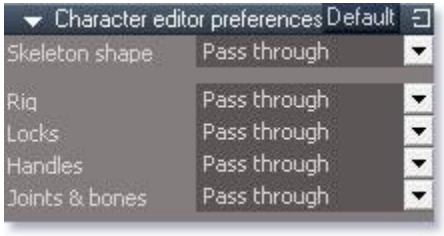
Many of the tools have associated panels that give you many options on how to work with the Character Editor. This section will give you a guide to each panel.

8.4.1 Character Editor Preferences

This panel opens in the Panels view in the stack when you right click on the Build Skeleton icon, or when you select the Shape Skeleton tool. It has several different aspects that affect the overall way character editing works, affecting both the UI and the tools.

Note that these settings affect all skeletons (though individual skeletons can have their own settings that override these defaults). The aspects are listed below:

Default



This panel controls how the various parts of the skeleton and rigging are displayed. The Skeleton Shape options let you choose the way the skeleton is rendered when being built / shaped, or being moved using Dynapose. You can choose a variety of bone shapes, and whether the skeleton should be wireframe or solid.

The other options let you show or hide the various parts of skeletons in the scene.

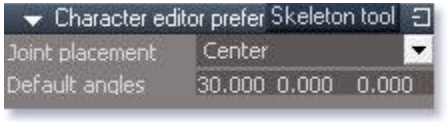
Colors

The Colors aspect controls how your skeleton looks. This is purely for your convenience in building and working with the skeleton and does not affect a rendered animation. As well as color, there are sliders to control transparency. All skeletons use these colors by default, but they can be overridden for individual skeletons if desired.

Most parameters are self explanatory. Those that may need more explanation are:

- **Active Lock**: When a lock is enabled, or selected for editing, this color is used to highlight the lock.
- **Mixed Lock**: It is possible to select multiple IK Handles for editing at once (see later for more information about IK Handles). If all IK Handles have a particular lock enabled, then it is shown as Active. However, a lock may be enabled for one IK Handle, but not for another – in this case, it is a "Mixed Lock" as it is not enabled for all selected IK Handles, and is displayed using the color set here.
- **Highlights**: Anything that is highlighted with a mouse over is displayed in this color, for example an IK Handle when you mouse over it for use or editing.
- **Attached Obj**: Once a mesh is attached to the skeleton, it is displayed using this color.

Skeleton Tool



This controls how the skeleton is built with two important parameters. The Joint Placement parameter has three settings:

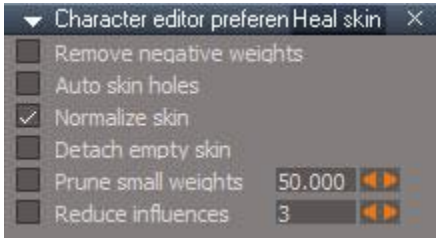
"**Center**" which will place joints inside an object. This is ideal for most uses, such as building a skeleton inside a character mesh. This is the most common option when constructing a skeleton for a character.

"**Ground**" which will place joints always on the ground, ideal for building a skeleton in an empty scene if you are using Perspective view (as an alternative, to build an upright skeleton for example, you could switch to building in an orthogonal view such as Front).

"**Surface**" which will place joints on the outside of an object, such as following the surface of a sphere or any other shape, useful for creating skeletons that will pull and distort the shape of a mesh rather than controlling the mesh from the inside like a real world skeleton.

The Default Angles parameter controls the joint limits that are set by default when you add a new joint. These limits are for x, y and z rotation and values vary from 0 (no rotation allowed in that axis) to 360 (full rotation allowed for that axis). You can always adjust the limits for each joint later of course, and you should use these parameters to select the defaults you find most useful.

Heal Skin



This aspect displays settings that affect how the Heal Skin tool will function.

- **Remove Negative Weights** - removes all negative skin weights. Negative skin weights have reversed effects of "pushing" vertices away from the movement of the bone, rather than pulling them to the movement of the bone.

- **Auto Skin Holes** - if there is a hole (vertices that are not attached to any bone), then skin weights are extrapolated from the surrounding weight painted geometry. Note that vertices are not assigned to the nearest bone, but instead are assigned to the same bone as nearby vertices. If a bone has no vertices assigned to it, then the Auto Skin Holes option will not assign vertices to that bone.

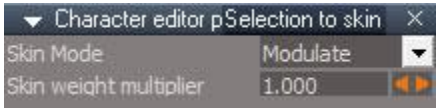
- **Normalize Skin** – Makes sure that the total value for influences for any point on the surface is equal to one. This means if a vertex is affected by more than one bone that the sum total of all influences from all bones will be normalized to add up to a total of 1. It also means that if a vertex is not assigned to a bone (ie has a zero skin weight), then it will be assigned to the nearest bone. Also, any vertices that have partial weights of between zero and 1 and which are assigned to only one bone will have their weight set to 1.

- **Detach Empty Skin** – If a vertex has a zero skin weight, then it is detached from the skeleton.

- **Prune Small Weights** – This parameter has a value associated. If the skin weight is less than this value, then the skin weight will be set to 0 (if you set a high value so that all weights will be pruned to zero, and check Detach Empty Skin, then the Heal Skin tool will detach the skeleton and skin).

- **Reduce Influences** – Each skin vertex can be attached to and influenced by multiple bones. This parameter specifies the maximal number of bones that can influence a vertex. If there are more bones influencing the vertex than this maximal number, then the ones with the smallest weights on that vertex are removed, thus reducing the number of bones that influence that vertex.

Selection To Skin



You can use Point Edit selection tools to make selections, then convert these into skin weights using the Convert Soft Selection To Weight Paint tool. This panel controls how that conversion is done with the two parameters.

Skin Mode controls how the soft selection is combined with the original skin weights, to give the new final skin weights. The different modes let you select different ways of combining the soft selection using the Point Edit tools into the skin weights.

- **Max –** Takes the largest value between the soft selection and the skin weights, and uses that as the new skin weight. This mode means your selection using the Point Edit tools will effectively "add to" the original skin weights. This means that if a point was not selected in Point Edit mode, it will retain its original skin weight.

- **Min** – Takes the smallest value between the soft selection and the skin weights, and uses that as the new skin weight. This means that if a point was not selected in Point Edit mode, it will have a zero skin weight. If a point as a lower strength of selection in Point Edit mode, then this will replace the original skin weight also. Finally, if a point is selected in Point Edit, but has an original zero skin weight, it will still have a zero skin weight.

- **Modulate** – This takes the Point Edit strength, and multiplies the original skin weight by that value. Effectively this makes the Point Edit selection a mask that weakens the original skin weight strengths. For example, if a vertex has a Point Edit selection strength of 1, then the original skin weight strength is unchanged (note it is not increased to 1; the original weight paint is multiplied by 1, so remains unchanged). If the vertex has a Point Edit selection strength of 0.5, then the original skin weight strength will be reduced to half its original strength. Of course with a Point Edit selection strength of 0, then the final skin weight will also be zero.

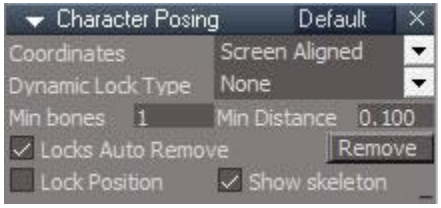
- **Replace** – The Point Edit selection strength overwrites the skin weight completely. The original skin weights are entirely replaced by the Point Edit selection. If a point is not selected in Point Edit, then it will have a final skin weight of zero, regardless of whether it originally had a skin weight (compare with Update).

- **Set** – The skin weights for the selected geometry are set to the value defined by the Weight Multiplier parameter.

- **Update** - This one replaces the selected vertices. For example, if you have already skin weights defined, then those vertices in your selection will be updated, while those outside your selection will remain untouched and will not lose their existing weights (compare with Replace).

Skin Weight Multiplier specifies the value that is used for multiplying selection weights when converting to skin weights. For example, a Skin Weight Multiplier of 0.3 means that a fully selected vertex will become a vertex with skin weighting of 0.3, and so on. The exact use depends on the mode of operation as set in the Mode parameter.

8.4.2 Character Posing



This is displayed in the Panels tab in the stack when the Dynapose tool is selected (if you have closed the panel, you can reopen it by right clicking the Dynapose tool).

- **Co-ordinates** – this parameter lets you define how your skeleton moves when you click and drag on it. The X, Y and Z movement that your mouse move initiates will depend on what Co-ordinates system you have selected.

To get a feel for the differences, you may want to choose the 4View layout. You will then see the character from all angles, and will be able to see the directions it is moving in more clearly.

"Screen Aligned" is the default setting, and is the most useful for general manipulation from a perspective or camera view. It creates a co-ordinate system for movement that is perpendicular to the ground, and which uses the screen X and Y. The movement you create will change depending on where you are looking from. If you rotate your view to the side, and use the left mouse button, you can drag the foot forward and backward; rotate to a more front view, and the left mouse button will let you move the foot left and right; other views will give other movements.

"Screen" is similar to "Aligned Screen" in that it uses the screen x and y for movement, but it uses "in and out of the screen" as the other axis for movement rather than perpendicular to the ground.

"Joint" will move relative to the orientation of the current joint itself, regardless of the angle you are viewing the model from.

"World" creates movement relative to the world space, regardless of which angle you are looking at the character from. The left mouse button creates movement either left and right or backward and forward in the space, while the right mouse button creates movement up and down in the space. Note this is independent of the character's orientation, so for example if the character is standing upright, a right mouse click on the foot will cause them to raise their leg; if the character is lying on the ground, then a right mouse click on the foot will cause a kicking movement – in both cases the foot is moved upward relative to the world, and not to the character.

- **Dynamic Lock Type** – this allows you to choose how Dynamic Locks work. Dynamic Locks work by freezing parts of the skeleton in place once you pose them for a particular keyframes. For example, let's say you move the left arm into a certain position on keyframes 20. When you then begin to adjust the right arm, trueSpace knows you already set the left arm, and with Dynamic Locks enabled, trueSpace will act as if you had enabled a lock to keep the left arm in place.

  Then if you go on to adjust the left leg on frame 20, trueSpace knows you have adjusted both arms, and again acts as if you had Locks on the arms which you had enabled to keep them in place. This can be useful in making it easier to set poses for characters with no IK Handles.

  Dynamic Locks can be set to None (all parts of the character will move according to the joint limits, unless you manually create and set locks); Position, which will Lock moved parts of the character in position but still allow rotation; and Position / Rotation that locks both position and rotation.

  You can still use other Locks with Dynamic Locks enabled. If you activate a Lock manually, the effect of that Lock will be combined with the effect of the Dynamic Locks.

  The exact effect of Dynamic Locks is also controlled by the next few parameters.

- **Min Bones** – this defines how many bones are cleared of dynamic locks to allow the character to be posed. If set to 1, then only the bone you click and drag on when posing is cleared of Dynamic Locks. If you pick 3, then bones up to 3 away from the bone you click on to pose the skeleton can be cleared of Dynamic Locks. With too large a value, you will find Dynamic Locks have no effect, as all the Dynamic Locks will be cleared, giving the same motion as when Dynamic Locks are disabled.

- **Min Distance** – this works the same way as the Min Bones parameter, only it works in terms of measured distance rather than in terms of the number of bones. This can be useful for working with areas with lots of small bones close together, for example for fingers, toes, or facial animation.

- **Locks Auto Remove** – If checked, then selecting a new current frame in the Animation Editor will automatically remove any Dynamic Locks. If unchecked, Dynamic Locks can only be cleared manually.

- **Remove** – This button manually removes any Dynamic Locks that are currently active. It will not affect the current post of the model of course, but will affect the result when you next manipulate the model into a new pose.

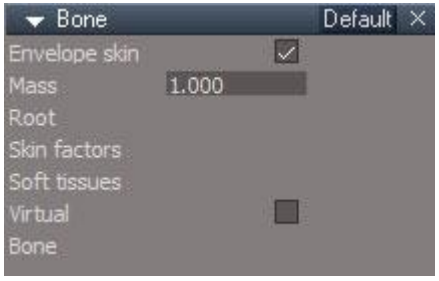
- **Lock Position** – This locks the character in their current position in space. If unchecked, dragging on the arm for instance can have the character zoom off across the scene. When this is checked, however, the arm will move but the character will be "pinned" to their current location.

- **Show Skeleton** – Controls whether the skeleton is shown using X-Ray mode when Dynapose is active (when checked), or if X-Ray mode is not enabled when Dynapose is active (when unchecked).

8.4.3 Shape Skeleton Panels

These panels open in the Panels tab in the stack when you click on particular parts of the skeleton using the Shape Skeleton tool.

Bone



This appears when you click on a bone using the Shape Skeleton tool.

- **Envelope Skin** controls whether or not the bone envelopes have an effect. With this unchecked, only skin weights will have an effect, and any bone envelope will be ignored.

- **Mass** contains a value that affects that bone during physics only, and has no effect on using Dynapose to interact with the skeleton.

- **Virtual** has two effects. First, it sets whether or not a bone should be rendered in the real-time engine – when Virtual is checked, the bone is rendered as a wireframe regardless of skeleton settings. This is useful for small bones that may otherwise clutter the view, such as with small bones for facial animation.

  Next, if a bone is set to Virtual before using Attach Skin To Skeleton, then the bone will be ignored during the Attach process – no vertices will be assigned to it, and so it will have no influence on the skin. This lets you set up the skeleton to use some bones as a way of simply connecting two parts of the skeleton together while having no effect on the deformations during animation (in a way, it is like having two separate skeletons attached via the Virtual bone). Note that making a bone Virtual after attaching the skin does not change or remove its influences.

The other parameters are for future developments and implementation and have no values displayed at present.

Joint

This panel is displayed when you click on a joint using the Shape Skeleton tool.

- **Limb tag** gives a name to the kind of limb that starts from this joint, for example, Arm or Leg. This is used with the drag & drop and library features for Limbs (for example, you can have joints with a Limb Tag of "Arm" – then double clicking an Arm in a limb library will add it to all four of that tags simultaneously, giving your character four arms). Note that this name is case sensitive.

- **Symmetry** defines whether the limb at this location is a left or right limb. Affects the way a Limb is mirrored when added at this joint.

- **Damping** is for physics only and has no effect on interacting with the skeleton using Dynapose. Valid values are 0 to 1.

8.4.4 Skeleton

This panel is displayed when you first build a skeleton, and before it is attached to a mesh and it lets you override the defaults set in the Character Editor Preferences panels. To use unique colors for this skeleton rather than the ones defined in the Character Editor Preferences panel, uncheck the "Use Character Editor color settings" box and you can then define the colors for this individual skeleton using the parameters below.

The visibility options for this skeleton automatically override any settings in the Character Editor Preferences panel.

8.4.5 Attach Skin to Skeleton

This panel is displayed when you select the Attach Skin To Skeleton tool, and controls how a skin is attached to a skeleton, in particular how vertices in the mesh are assigned to the bones in the skeleton. Note that if you want to make changes to these values, you must do so before you click on the mesh you want to use as a skin.

- **Maximal Number Of Bones** defines the maximum number of bones that a vertex is attached to. It is quite possible (and preferable) for one vertex to be affected by more than one bone. This is set to four by default. Four is optimal for game engines, as you cannot assign a vertex to more than four bones with hardware skinning.

- **Blending / Separation** acts in a similar way to Transition Smoothness (see below), but applies when there are 3 or more bones influencing a vertex. Again, a larger value means a smoother transition region.

- **Transition Smoothness** defines how long the transition region between 2 neighboring bones will be. The larger the value, the larger the region will be. Vertices inside this transition region will be affected by both bones (with varying intensity depending on how close they are to each bone).

8.4.6 IK Handle

This panel gives you the settings that control the behavior of an IK Handle.

- **Left Button Motion** – Define what motion is created when the left mouse button is used to click and drag on the handle
- **Right Button Motion** - Define what motion is created when the right mouse button is used to click and drag on the handle
- **Shape** – Define the shape used to represent the Handle. This has no effect on the functional of the Handle, but is so you can readily identify what the Handle does by choosing a shape which is meaningful for you.
- **Color** – Choose the color for the IK Handle. Again this has no effect on the function, and can be used in any way that is meaningful and useful for you.
- **Speed vs Quality** – This controls the quality of the IK solver, letting you choose between higher precision or a faster response. This mainly applies to long chains, where precision versus response time becomes more of an issue.
- **Interpolation** – This is a slider that lets you choose how the IK Handle works in animations. In full IK (Inverse Kinematics) mode, the limb fully follows the handle. That means if you add 2 key frames, then the IK handle will move straight between the two points and the limb will follow the handle.

  In full FK (Forward Kinematics), the handle is moved as if only FK keyframes were made, so Handle follows the bone it is attached to, and will move on an arc. Values in between the two extremes will blend between those two values.

  Both FK and IK interpolation will run the IK solver to calculate how the character updates and moves in the animation. In full IK mode, the IK Handle position is interpolated between one keyframe and the next, and then the whole skeleton is moved according to the animated movement of that Handle (Locks are applied).

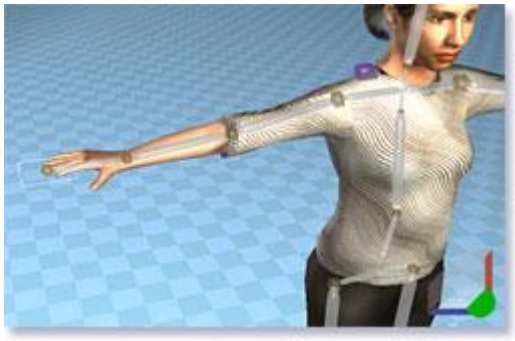
  In FK mode, the skeleton is first interpolated using FK, and here the joint angles are interpolated between one keyframe and the next rather than the IK Handle position. Then the IK Handle positions are calculated and the IK is updated. Handles are then moved to their FK position (as other handles may have moved the FK handle during solving and this ensures that handle is moved back to its correct location).

  With the Posing Only parameter checked, then the IK Handle is turned off during animation and is not solved at all so that it acts as a pure posing handle.
- **Use Global Locks** – An IK Handle can have Locks specifically assigned to be active when the Handle is used. When this is unchecked, the IK Handle will only use those Locks, and will ignore any other Locks that have been manually activated. When checked, the IK Handle will use any Locks you have enabled during Dynapose, in combination with its own Locks.

Disabling this value lets you ensure the IK Handle always responds the same way, regardless of what Locks you have set for other posing purposes. Enabling this value lets you adjust what the IK Handle does each time you use it by enabling other Locks before using it.

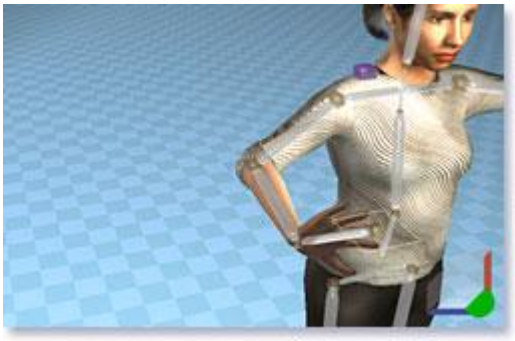
- **Lock Children** – When checked, this creates one FK group from the bone and all child bones (until the first Rotation Lock ends the group). The "Invert IK Groups" parameter will affect what is classed as a child, whether it goes from the parent bone down, or from the leaf bone up.
- **Invert IK Groups** – Rotation Locks can be used to define IK Chains that are controlled by the IK Handle. This checkbox lets you invert the chains that are locked, and which are not locked.
- **Posing only** – The handle is used only for posing and is not evaluated during animation. If you place the character in a pose using a Posing Only IK Handle and record a keyframe, then the result is the same as if you set the pose using the Dynapose tool.
- **Firm Handle** – This changes the primary goal for the IK solver from Locks to Handle. If checked, the Handle will be firmly attached to the bone and will take priority, meaning that you will break any Locks if you move to a pose that the Locks would normally prevent you from reaching. This can be useful for making a character jump for example, with Locks on their feet – the locks will keep their feet on the ground, until that is no longer possible, and then the Locks will be broken and the feet will leave the ground (but will still point toward their Locks, which will remain in place on the ground).

  If unchecked, then Locks take priority and a Handle cannot move the character outside of the poses permitted by the Locks.
- **Animate As Soft** – This is related to the Firm Handle setting. When you animate, you define the end keyframes for each movement. This means that the locks will not get broken very often during animation since they are animated using FK rather than IK. If you want to preserve Locks (for instance, to prevent the feet from sliding), you should use this setting to ensure that the Locks are never broken
- **Preserve Bone Orientation** – The bone to which this Handle is attached will not be rotated by this Handle. For example, if the IK Handle is attached to the bone in the hand, with this checked, moving the Handle will allow the arm to bend, but keep the hand pointing in the same direction; if unchecked, then the hand will bend and rotate. This can also affect the rotation at the elbow in this example, since keeping the hand in the correct orientation will affect how the elbow is bent. See images below.



The starting pose.

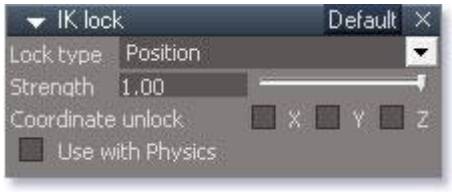
The end pose when Preserve Bone Orientation is checked – note how the hand has not bend from the starting pose.



The end pose when Preserve Bone Orientation is unchecked – note how the hand bends.

8.4.7 IK Lock

Default aspect:

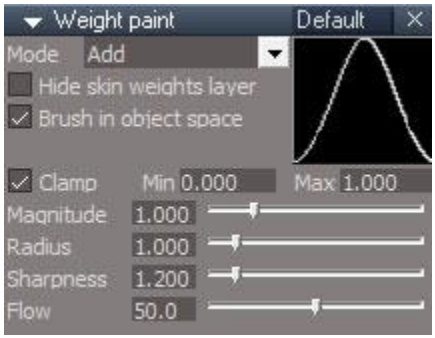


- **Lock Type** – lets you specify whether the lock is position, rotation or both (a full lock). The 3D object representing the lock will change when you adjust this setting.
- **Strength** – lets you set how firmly the lock holds the bone or joint in place, with values ranging from 0 (no limitation of the movement / rotation of the joint or bone) to 1 (complete limitation of the movement / rotation of the joint or bone).
- **Co-ordinate Unlocks** – let you disable the effect of the Lock in a particular direction – when unchecked, the Lock prevents motion in that direction. When checked, it allows motion in that direction.
- **Use With Physics** – If checked, then this lock is used when the character is animated using Physics Simulation.

Tags aspect:

- **Tag** – lets you specify a name for this Lock. This allows Limbs that contain IK Handles to use this Lock even if it is outside of the Limb itself (for example, when an IK Handle on an Arm references a Lock on the shoulder).
- **Symmetry** – Defines how the lock is placed on the bone. Can be very important for use with Limbs, for example when two limbs with different symmetry have an IK Handle using the same Lock (eg an Arm, with an IK Handle on the hand that uses a Lock on the shoulder – this would need symmetry set to “center” so that it works evenly for both the left and right arms).

8.4.8 Weight Paint



Weight Paint lets you paint onto the mesh using a brush to determine which vertices are influenced by which bones and by how much. This panel lets you set the parameters for the Weight Paint tool.

The Weight Paint panel features a variety of aspects, which match the different Modes that can be used for the Weight Paint Brush. These aspects hide parameters that do not apply to particular Modes to make it easier to work with the different Modes. It is important to note though that these aspects do not change the mode itself.

That means that selecting the Smooth aspect in the Panel will display only those parameters that affect the Smooth mode for the Brush, but you still need to manually select the Smooth setting in the Mode parameter.

- **Mode** – This defines how the Brush paints weights onto the surface.
    - **Add** – This mode adds extra weight onto the surface as you paint with the Brush. This increases the influence the selected bone has on those vertices. The Magnitude controls how much of an increase you get. For example, a Magnitude of 0.4 means the Brush will add 0.4 to the weight on the mesh, so if the original weight was 0, the maximum after painting will be 0.4; if the original weight was 0.3, then the maximum after painting will be 0.7; and so on. Releasing the mouse button and then painting again will again add more weight.
    - **Grow** - Grow expands skin weights around the currently weighted area. This means that points adjacent to a weighted area will have weight added to them. However, points that are not adjacent to an already weighted area will not have any weight added. Magnitude does not affect the Grow parameter (the Grow aspect of the Weight Paint panel does not include the Magnitude control for this reason). However it is sensitive to the Flow parameter, the lower the value set in Flow, then the lower the weight value will "spread" from the current area to the surrounding areas.
    - **Set** - This will set the weighting on the surface to the value set in the Magnitude parameter. The Flow parameter still controls how quickly the point reaches the set Magnitude (raise the Flow if you want the Brush to quickly set the weight to the Magnitude). If the current weight on the surface is less than the Magnitude, then the Brush will raise the weight toward the Magnitude. If the current weight on the surface is less than the Magnitude, then the Brush will lower the current weight toward the Magnitude.
    - **Shrink** – The opposite of Grow, this setting reduces the weighted area. Note that it only affects the outer most points on the surface, reducing the weight painting on those (points further inside the weighted area are not affected). The Magnitude parameter has no effect in Shrink mode.

However it is sensitive to the Flow parameter, the lower the value set in Flow, the less the weight will be removed from the points around the edge of the weighted area.

- **Smooth** – Acts similar to a blur tool in a 2D paint application, blending the weight values and producing areas of smoother transition. It blends in both directions, for example on a sharp boundary between heavily weighted and no weight, the heavily weighted points will have their weight reduced, while those points without weighting will have weight applied.
- **Subtract** – This mode reduces the weight on the surface as you paint with the Brush. This decreases the influence the selected bone has on those vertices. The Magnitude controls how much weight is removed from the areas. For example, a Magnitude of 0.4 means the Brush will remove 0.4 from the weight on the mesh, so if the original weight was 0.8, the minimum after painting will be 0.4; if the original weight was 0.6, then the minimum after painting will be 0.2; and so on. Releasing the mouse button and then painting again will again remove more weight.

- **Hide Skin Weights Layer** – This turns off the display of the weight values which is normally shown by coloring the mesh. Check this when you do not want to see the coloring, and just want to see the shape of the mesh. The mesh will then be displayed with its regular texture.
- **Brush In Object Space** – This lets you move the brush to work in the object space rather than in the surface space. This can be useful if the surface has a lot of bumps, wrinkles or other strong changes over a short distance in the surface.
- **Clamp, Min and Max** – If Clamp is checked, then the final value of the Weight Painting is constrained by the Min and Max values (if Clamp is not checked, then Min and Max have no effect). The Min defines the lowest value of association to a bone that can be achieved by the brush in Subtract mode, and the Max defines the highest level of association to a bone that can be achieved in Add mode.

  For example, with a Min of 0.4 and the brush in Subtract mode, you can only reduce the assignment to a bone for a vertex to 0.4. This would mean that you could not make the vertex fully free of the bones influence, as you cannot lower it below the Min value. Similarly with a Max of 0.6 and the brush in Add mode, you cannot assign the vertex to be influenced fully by the bone, as 0.6 is the highest weight you could paint onto the surface.

  To fully detach the vertex from any influence by a bone, set Min to 0 and use Subtract. Note that you can set Min to a negative value to the weight painted on the surface (shown as red). This will make those points be pushed away from the bone rather than pulled toward it.
- **Magnitude** –This controls the overall strength of the brush. Higher values mean a stronger brush that will either assign or remove vertices from bones more fully than lower values.
- **Radius** – This controls the size of the brush, larger values mean a larger brush.
- **Sharpness** – This controls the fall off at the edges of the brush. A low Sharpness gives a smoother blending at the brush edges, while a higher value gives a sharper, more sudden blending.

- **Flow** – Controls how quickly the brush affects the surface. A low value will mean you will need to make repeated strokes over an area to build up the effect of the brush, while a high value will quickly affect the area, perhaps with just a single mouse click.