Character Rigging In Maya
using ekCharacter ToolKit.mel

Introduction

These notes are meant to supplement and not replace the class lectures. Practice doesn’t make perfect; PERFECT practice makes perfect. The more you do this, the more you practice it, the better you will get at it. Persistence is everything.

As animators, you either need to have characters rigged for you or you rig them yourself! If you want to tell your own stories, you will have to confront the sometimes intimidating art of rigging characters for animating. Always remember that this isn’t meant to be oppressive; it’s about independence!

Rigging is the sometimes excruciatingly technical art of articulating a character for animation. It involves creating a skeleton structure out of joints and bones that will deform the model. Then, the Technical Director (TD) creates controls that the animator uses to move the character. The animator sets keyframes on these controls, and the animation process continues from there. For the purposes of this class YOU will be the TD.

This process is complicated by the fact that there are as many ways of rigging characters as there are set-up artists and animators. Each TD or set-up artist has ways of creating rigs, each animator has certain expectations for the rigs, and each production has certain needs that the rigger and animator must bow to. And of course, each character has features and needs that will determine what techniques will be needed.

You have the advantage in that you are rigging for your style of animation. Furthermore, you are blessed with being at the beginning of your animation careers, so your expectations of what you want in a rig are fairly simple. Unfortunately, you are at the beginning of your careers, so you will have to quickly learn and master some fairly heavy concepts

The process you will learn is contained in this booklet, and it consists of the following steps:

• Creating and naming the joint hierarchy. We will use CometJointRename.mel to help in this.
• Re-aligning the joints’ Local Rotation Axes using CometJointOrient.mel.
• Create separate leg skeletons for FK and IK.
• Create the IK Controls using Jason Schliefer’s Grouped inverse foot technique. Then we will see how a custom mel script can automate this process.
• Install the “No-Flip” knee controls.
• Create FK Controls on each leg.
• Create a switch control to switch between the two control skeletons.
• Create StretchyIK back.
• Create FK controls for the stretchy back.
• Create separate control skeleton controls for the arm IK and FK.
• Create the FK controls for the collar, shoulder, elbow.
• Create forearm average control for the Forearm and Wrist.
• Create the IK Controls for the arm.
• Create the neck and head controls. Create eye controls if necessary.
• Create Follow control for the head.
• Create Follow controls to allow the IK legs and arms to follow the body or the world.
• Group all controls appropriately for animation.

Yes, this takes EXACTLY as long as it sounds! Which is why I, your soon to be beloved instructor, have written `ekCharacterToolKit` for your use. This Mel script provides a push button interface for using the many and varied scripts designed to automate the rigging process. This document was originally 280 pages of which the first 140 pages was all about rigging the skeleton. It is now quite a bit shorter!

**Automation vs. Customization**

Unfortunately, to take advantage of this wonderful automation, there are a couple of things that you MUST do:

**Spelling and Capitalization:** The power of naming in Maya is enormous! The ancients believed that if you knew someone’s name, you had power over that person. In Maya, if you know the name of something, you can change it through scripting. The scripts I have written rely HEAVILY on proper naming. Therefore you must name things EXACTLY as I require. Failure to do so will cause just about every one of my scripts to fail. The good news is that the scripts I have written by and large name things for you, but there may come times when you will need to type or rename some things yourself.

**Orientation:** As you will see, almost all the joints have their default rotations set a particular way. This default rotation is its Orientation. There are several ways to orient a skeletal joint; for consistency’s sake, I will insist on a certain way of doing things.

To get you into the animation phase as quickly as possible through automating the set up process, we must sacrifice some ability to
customize your characters. That is part of the inevitable trade-off in this process. To make it practical to set up multiple characters, you will not learn to rig from scratch, you will learn to use the tools I have written. The good news is that they are powerful, flexible and will not limit you in terms of your animation abilities.

With this in mind, let’s begin to create the skeleton.

**The Maya System of Joints and Bones**

Maya joints are a type of deformer that are designed to transfer rotational values to points on a model based on each joints influence on those points. Joints are most often arranged in a hierarchy—meaning that one joint is a child or farther down on the hierarchy than the parent.

Each child of a parent exists in the *local space* of that parent. This means that the translate values of the child are listed as distances from the parent. This is important to understand where joints are concerned. Zeroing out a joint’s translate attributes will cause the joint to snap back to the location of the parent. The ultimate parent is the World space as defined by the world origin at 0, 0, 0.

The three joints shown below were created with the X axis pointing down the bone, the Y axis pointing up and the Z axis pointing forward in space. Knowing this is important to understanding what comes next.

Pressing the W key, while LM clicking with the mouse will allow you to change the space in which the translate tool will work. You have three choices:

- **World**—Will move the joint (and any other object) in the World space along the World XYZ axes.
- **Local**—Will move the joint along the axes defined by the parent. Note the middle image above: the x axis is point in the same direction as the parent.
- **Object**—Will move the joint along the joint’s own axis as defined by its orientation.

Joints also have *Orientation* which will effect the direction in which they point. For our purposes, we will always work with the X axis pointing down the joint.

**Arbitrary Point #1**—Work with the X axis pointing down the joint. It is possible to have differing orientations for different parts of
the body. You may one day decide that it is easier to have Y pointing up the spine joints and X pointing down the arm joints. However for learning purposes (and because many of the scripts we will use to make these tasks easier expect the X axis to be pointing down the joint), it is much easier to learn with this consistent principle.

Installing ekCharacterToolKit.mel

Before we begin rigging, we must install ekCharacterToolKit.mel and all of the scripts that this UI script runs. To install all the scripts, do the following:

1. If you are in Maya, quit the program.
2. Copy the folder ekCharacterToolKit Files to your desktop.
3. In a Finder window, navigate to the following folder:

   User/Library/Preferences/Autodesk/Maya/8.5/scripts

   Where User is the name of the account you have. NOTE: it is crucial that you use the scripts folder in the 8.5 folder and NOT the scripts folder in the maya folder. If you make this mistake, the scripts will not load at startup and you will have to go back and move all these scripts to the right folder.

4. Copy the following files to this scripts folder:

   ekCharacterToolKit.mel
   cometJointOrient2.mel   ekRigFKIKLeg.mel
   cometRename.mel         ekCreateGroup.mel
   ekfingerRiggerUI.mel    ekRigFollow.mel
   ekMakeDEFSet.mel        ekRigIKEyes.mel
   ekRigArm.mel           ekMakeFaceCTRL.mel
   ekRigBack.mel          ekCleanCTRL.mel
   ekRigLeg.mel           ekMakeCTRLSet.mel
   ekRigNeckHead.mel      ekMakeWeightDance.mel
   ekRigSpine.mel         FS_faceCurves.mel
   ekFinishSpine.mel      FS_addInfluences.mel

   This should represent the entire contents of the folder.
5. Start Maya. When Maya starts it loads or “sources” all the .mel files in the scripts folder.

**Creating and Naming the Joint Heirarchy**

Our example character is named “Oswaldo”. Open OswaldoGeo.mb to see him in all his glory.

Let’s create the leg skeleton as follows.

1. In the side view, hold down the command key and press both the LM and MM buttons. Zoom in to center on the leg.

2. Press 4 to enter wireframe key. (Press Shift 4 to enter wireframe mode in all windows.

3. Go to **Skeleton > Joint Tool**. Press the Reset Tool button to set the tool the way we want it, then set both the short and long bone radii to 1. This keeps the joints the same size.
4. Draw a joint chain that looks like the one at left. Name the joints as shown. Click Hold and drag to move them around before setting the next joint.

If you need to move the joint you just set and have released the mouse button, MM drag the joint into position.

If the joints are too large to place properly, go to the Animation Preferences window, click on the Kinematics category at left and change the Joint Size value. (.38 shown here).

Here we are using a two knee joint configuration which makes weighting the knee a little easier.

Arbitrary Point #2

There MUST be a slight bend in the chain from the Hip to the Ankle. A completely straight chain will not bend with the IK. Also, you should be diligent in placing joints in the middle of geometry that will deform as seen in the knee and ankle areas. This is contrary to actual anatomy, but is crucial to getting smooth deformations.
Now let's get to the interface:

In the command line (in the lower left corner of the interface) of the main Maya window type:

```
source ekCharacterToolKit.mel; ekCharacterToolKit;
```

Then press the Enter or Return keys

At right is the ekCharacterToolKit interface. It simply represents the tools and commands we will use in rigging our characters. While I have not tested it on quadruped characters, there is no real reason why it should work.

As with any custom tool—you will need to learn how and when to use each command. There are certain things that you MUST do for each command to work. Part of the purpose of this document is to teach you how to use each one.

These commands are listed in order of use. Most often, you will create and name a joint hierarchy or skeleton first, so the most useful tool for that purpose is listed first.

Now that you have the leg and foot joints created and basically named, we can speed things up with the Comet Rename script. Press that button and…
...you will see the window at left. This tool renames objects and joints. More importantly, it also adds prefixes and suffixes and searches and replaces text strings in object names.

We will use it to add the character prefix and the side suffix.

We will do so as follows:

1. Select the Hip joint by clicking on the top level joint or by pressing the up arrow key several times. The arrow keys let you navigate up and down a hierarchy.

2. Press the “Select Hierarchy” button at the bottom of the cometRename window. This selects the entire joint hierarchy. Note how the joints change color with the last joint in the hierarchy turning the color green while all other joints are white. Noting the colors in the selection will help you determine what is selected first (white) and what is selected last (green).

3. With all the joints selected, type “Osw_” in the Prefix: field.

4. Click the Add Prefix button. Note how the names of the joints change in the Channel Box.

5. Type “_L” in the Suffix: field and press Add Suffix. This appends the side suffix to the joint’s name.

6. Click on the PerspOutliner button in the toolbar.
7. In the now visible Outliner, shift-click on the + next to the Osw_Hip_L joint to open the hierarchy. Now your Outliner window should look like this.

Clearly, CometRename can save you a lot of typing.

When you create joints from one of the orthogonal planes (Front, Side, Top), you are making them on the XYZ plane that is perpendicular to that window. For example, when you made the joint chain originally, you were creating the joints on the YX plane directly in the middle of the character. Let’s line this joint chain as follows:

1. Select the Osw_Hip_L joint.
2. Press and hold the space bar to bring up the Hotbox.
3. Click and hold on Autodesk to bring up the view marking menue and go to the front view.
4. Press the w key to invoke the Translate tool.

5. Drag the Osw_Hip_L joint to line up with the middle of the left leg as shown at left.

**Arbitrary point #3:**
When modeling your characters, it will behoove you to model your characters with legs parallel to the YX plane because this will make lining up your leg joints much easier. Making characters that are bow-legged or pigeon-toed, while possible, are unnecessarily complicated at the beginning of your character design efforts.

Next, we will mirror these joints as follows:

1. Go to Skeleton>Mirror Joints. Click YZ for the Mirror Across: option, choose Behavior for the Mirror Function. Type “_L” for the Search For: field and type “_R” for the Replace With: field.

2. Click Mirror, and your joints will mirror across the YZ plane and be suffixed with the _R designator.
Creating The Spine

The Spine of any character is the most important set of joints in any rig. A flexible, supple spine allows the animator to create interesting action lines to his or her poses. This is the beginning of any superior animation.

In the images at left, you can see an anatomical drawing of the spine juxtaposed with a preliminary placement of the spine with our character. This juxtaposition reveals

**Arbitrary Point #4:** Unlike a “real” spine, you should place the joints in the middle of the character. This will aid the deformation of the torso and abdomen.

Let’s create our spine as follows:

1. Select **Skeleton>Joint Tool** from the Skeleton menu.

2. In the side view begin at the bottom and create a series of joints like you see at left. Try to create a spine of around 12 joints. This will allow suppleness while not being too difficult to weight later on. You can click and drag to place joints. You can also MM click on the most previously placed joint to move it into position.

3. Press Return to accept the chain.

I can almost guarantee you that your joint chain will not be perfect. Mine never is. So we will adjust joints within the chain as follows.
1. Select the joint you want to move.
2. Press the w key to invoke the translate tool.
3. LM drag the joint and all its children into place.

BUT WAIT! What if I want to move the joint, but not the children?

4. Before moving the joint, press the Home key on the keyboard to enter Move Pivot mode. The movement cursor will change.
5. Move the joint as needed. The children will stay in place.

Practice a little “digital chiropractic” and align the spine as you see in the next image. Try to keep a smooth line to the spine hierarchy.

Press the Home Key again to leave Move Pivot mode.

Now we will name the spine appropriately.
Renaming the Spine

Rename the Spine joints as shown either using cometRename or by manually renaming each joint.

Manually rename the joint by clicking on the joint, then changing its name in the Channel Box (shown below) or outliner.

```
<table>
<thead>
<tr>
<th>Channels</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CSpine</td>
</tr>
<tr>
<td>Translate X</td>
<td>0.253</td>
</tr>
<tr>
<td>Translate Y</td>
<td>-0.02</td>
</tr>
<tr>
<td>Translate Z</td>
<td>0</td>
</tr>
<tr>
<td>Rotate X</td>
<td>0</td>
</tr>
<tr>
<td>Rotate Y</td>
<td>0</td>
</tr>
<tr>
<td>Rotate Z</td>
<td>0</td>
</tr>
<tr>
<td>Scale X</td>
<td>1</td>
</tr>
<tr>
<td>Scale Y</td>
<td>1</td>
</tr>
<tr>
<td>Scale Z</td>
<td>1</td>
</tr>
<tr>
<td>Visibility</td>
<td>on</td>
</tr>
<tr>
<td>Radius</td>
<td>1</td>
</tr>
</tbody>
</table>
```

Adding the Neck and Head Joints

We will move a little faster when adding the Neck and Head joints.

1. Go to **Skeleton>Joint Tool**.
2. Click on the CSpine joint to begin adding joints to the hierarchy.
3. Add joints as shown at left.
4. After placing the HeadTop joint, press Return to accept the chain.
5. Rename each joint as shown.
Adding the Jaw and Mouth Joints

The Face/Jaw construction is very easy; we will not use many joints because we will do facial poses using influence curves and blendshapes. Let’s create the Jaw Face joints as follows.

1. Go to **Skeleton>Joint Tool**.
2. Click on the Head joint to begin adding joints.
3. Create a chain as shown at left. Press return to accept the chain.
4. Name the joints as indicated. NOTE! It is CRUCIAL that you spell and capitalize these joints exactly as indicated.
5. Deselect the chain by clicking anywhere in the background.
6. Go to **Skeleton>Joint Tool**.
7. Click on the LowerJawBase joint to begin adding joints.
8. Add a joint at the corner of the mouth. The easiest way to do this is to add it from the side view and move it into position from the front view. You do this by MM click-hold and dragging it in the front view.
9. Name it “Mouth_L”.
10. Go to **Skeleton>Mirror Joint** to create a mirrored copy called Mouth_R.

These joints will figure very prominently when we rig the facial curves.
Creating the Arm Joints

Arbitrary Point #5: When modeling, create the arms of your characters facing straight out from the body in what is called the crucifix pose. This allows you to place joints much more easily.

Let’s place the arm joints as follows:

1. Make sure you are in the front view. (Hold Spacebar, click on Autodesk in the middle of the Hotbox and choose Front).

2. Go to Skeleton>Joint Tool.

3. Hold down the x key and place joints at the locations indicated. The x key invokes Maya’s Grid Snap feature which places items on each grid location. You are working in the front view, so the joints will be place flat on the YX plane. We will fine-tune their placement now. Press return to accept the chain.

4. Name them as indicated at the bottom of this image (remember, spelling counts!)

4. Press the Home key on the keyboard to enter Move Pivot mode.
5. Dragging them along the X axis ONLY, move them into the position shown below:

![Image of character with joints moved along X axis](image)

6. When finished, press the Home Key again to leave Move Pivot mode.

7. Use the Hotbox to switch to the Top view.

This is where each character will be a bit different. A character with a very muscular chest, back and shoulder will need to have the joint placed slightly differently then Oswaldo. What we want to do is place the Collar joint midway between the chest and back while the shoulder joint needs to rest in the center of the shoulder mass. Careless placement will lead to a big headache later. Lets place these joints as follows:

1. Select the Collar joint.
2. Press w to invoke the Translate tool.
3. Using ONLY the Z axis (the blue arrow) move the collar bone back into the chest a little ways.
   
   (Note: The bottom image shows the perspective view set to show wireframe on shaded and to XRay. Set this in the panel menu bar in the Shading menu. I used the Hotbox to change the side view to the persp view.)

Now if your character called for it, you would do the same thing for the shoulder joint. But we lucked out because Oswaldo has a very easy area to place the joints into. Lets place the rest of the arm joints as follows:
1. Select the Shoulder joint.

2. Press the e key to invoke the rotate tool.

3. Click on the green Y axis and rotate it slightly counter-clockwise so that the elbow joint rests right in the middle of the elbow area of the model. Yes, this will knock the forearm, wrist and palm joints out of alignment, but don’t worry, we’ll fix that now.

4. Press the down arrow to drop the selection down the hierarchy to the Elbow joint.

5. Now rotate the Elbow joint slightly in Y clockwise until the wrist joint rests in the center of the wrist area of the model.

Now, I know that you, dear reader will of course do this perfectly each and every time you attempt this, and that all of your joints will line up perfectly. But if, by some wild cosmic accident, they should not line up, then do the following.

1. Hold down the w key and LM click in the top viewport. You will see a marking menu like that shown at left. This chooses what space you will use to move objects.

2. Choose Object. This will move the joints along the aligned axes.

3. Move the joints ONLY by translating them in X.

**Arbitrary Point #6:** The Elbow, Forearm, Wrist and Palm joints MUST remain in a straight line.

Now lets do the fingers!
1. Select **Skeleton > Joint Tool**.

2. Click on the Palm joint to start the chain.

3. In the top view, create a four joint middle finger chain as shown at left. When done, press Return.

4. Name the joints as shown.

5. Select the Mid joint.

6. Go to the **Edit > Duplicate**. Set the dialog as shown at left. It basically will make 1 copy of the joint which will be the child of Palm.

7. Click Apply.

8. Move the duplicate Finger to the Ring finger position as shown.
9. Rotate the finger chain (e key) so the chain aligns with the finger mesh.

10. Translate (w key) the joints in X to line up with the mesh knuckles.

11. Lastly, use cometRename or manually rename these joints to Ring, Ring2, Ring3 and RingTip respectively.

12. Repeat for the Pinky, Index and Thumb joints. Name them appropriately as well.
13. Use the joint tool to click on the palm joint and then click somewhere in the heel of the hand to create a joint as shown. Name this joint Cup.

14. Press q to enter the Select tool.

15. LM click on the Pink joint.

16. Hold down the shift key and click on the cup joint to add to the selection.

17. Go to Edit > Parent or press p to make Pink the child of Cup.

Now let’s align the joints into the fingers of the hand.

1. Starting with the index finger, use the Rotate Tool (e key) to rotate the Index joint so that its child joints sit as exactly as possible in the center of the finger.

2. Use the up and down arrows to quickly navigate up and down the hierarchy.
3. Repeat for each finger

If necessary, you can move each joint up or down in Y or back and forth in X. BUT you should never move them out of alignment by moving them along the Z axis. You won’t be happy!

4. Repeat this process for the Thumb joints.

Now, we want to assign a side to the arm and mirror the arm joints. Do this as follows:

1. Select the Collar joint.
2. From the ekCharacterToolKit window, press the Comet Rename button.
3. Click on the Select Hierarchy button in the Rename window.
4. Type _L in the Suffix: field and press Add Suffix to append the name of each joint in the chain with the “_L” suffix.
**Finishing the Skeleton**

All that is left for this phase is to mirror the arm, set up the root and make sure everything is named properly. Let’s do that as follows:

1. Click in the background to deselect everything.
2. Select Collar_L.
3. Go to Skeleton > Mirror Joint. Use the settings at left and press the Mirror button.

At this point, if your character was asymmetrical in the upper body, you would take time to move and/or rotate the right side joints into place.

Now, to parent the arms to the CSpine joint, let’s do the following:

1. Select Collar_R.
2. Hold down the Shift key and click on Collar_L.
3. Keeping the shift key held down, click on CSpine.
4. Press the p key to parent the arm joints to the CSpine joint. (Alternately, you can go to Edit > Parent.)

To finish up the back, we must create a root joint and parent the spine and legs to it. This joint is the top of the hierarchy for the entire skeleton, so it is important. Also, it is not oriented like all the other joints in that it is aligned to the world rather than having its X axis pointing at its child (it has three children, so that would be impossible anyway. Let’s create the root as follows:
1. Go to **Skeleton > Joint Tool** or press the Joint Tool button in the Animation Shelf.

2. Place 1 joint as shown at left in the front and side views. Press the Enter Key

3. Name the joint “Root”.

4. Select the Spine joint.

5. Hold down the shift key and select Osw_Hip_L.

6. Shift-select Osw_Hip_R.

7. Shift-select Root.


You now have a full hierarchy for your skeleton. Now we will make sure to name everything properly. I jumped the gun on naming the legs. (actually, I did it on purpose to let you see how to remove text using cometRename, but shhhh, don’t tell anyone!). We need to remove the “Osw_” prefix we added earlier. Then add it back in to the whole skeleton. Don’t worry, it is easy. Lets do it as follows:
1. Select the Root joint.

2. Bring up Comet Rename. (Either click on the button if ekCharacterToolKit is up, or type “source ekCharacterToolKit.mel; ekCharacterToolKit;” sans quotes, in the Command line. Then click the Comet Rename button.

3. Press the Select Hierarchy button to select all the joints in the skeleton.


5. Press Search and Replace. This will remove the prefix Osw_ from the legs.

Now let’s add it back to the whole skeleton with the DEF designation.

6. In the Prefix: field type “Osw_DEF” sans quotes. Make sure the DEF is capitalized and spelled correctly. 😊

7. Press the Add Prefix button.

You now have the entire skeleton hierarchy renamed with the Osw_DEF<function>_<side> nomenclature. This designates the entire skeleton as a deformation skeleton. The scripts we will use later use that DEF text as a tag to identify the purpose of a particular joint. VERY cool!
A quick glance through the Outliner hierarchy reveals the power of cometRename! Imagine having to go through each and every joint and rename it by hand.

Now, on to Joint Orientation!
Orienting the Skeleton

When placing joints, you almost always wind up translating them to move them into position. This disrupts the orientation of the parent to the child by moving the child so the parent no longer points at the child. This results in unpredictable rotations as shown below.

Rotating the parent joint around the X axis twistst the joint causing the middle joint to rotate predictably with and carry the end joint in an arc.

Moving the middle joint back in Z and over in X along the middle joint's X axis...

...causes the whole joint assembly to rotate completely out of whack!

Therefore: Joint orientation is crucial.

Each joint has what is called a Local Rotation Axis. This axis will be known henceforth as the LRA. It reflects how the joint is oriented. You can see them by choosing **Display > Component Display > Local Rotation Axes.**
You may edit them manually by going into component editing mode (press F8) and activate them by LM clicking and holding on the ? mark in the Component Selection Mask toolbar.

This will allow you to change the orientation of each joints’ LRA thus changing the directions of each joint’s rotation.

Our task is to set the LRAs of each joint in our skeleton so as to facilitate creating controls for our rotations. There are over 75 joints in our skeleton. Manually rotating each joint’s LRA is not only inefficient, it is also VERY inaccurate.
Fortunately, there is a tool that is the gold standard for changing joint orients. It is called cometJointOrient.mel. You can access it from the ekCharacterToolKit interface. It is the right under Comet Rename and is called Comet JointOrient.

Note: If you download this script from highend3D or from comet-cartoons.com, there will not be a Select Hierarchy button. I added this command as a time saving help to this script.

Orienting the Spine Joints

We will use this script to orient the joints in the skeleton starting with the spine. But first, we need to select the joints we want to change. To do this, we will use Maya’s quick select field located at top right. LM click on the sel button and choose Quick Selection from the popup field. Then type “*Spine*” as shown below:
Pressing Enter selects all the joints with the string Spine in the name. This includes the CSpine Joint.

1. In the cometJointOrient interface, press the Show Axis Button.

2. You should see all the axes revealed as shown at left.

Note that the bottom 4 joints are reversed when compared to the top 7. We need to change this.

3. Set the cometJointOrient as shown at left.

4. Press Orient Joints, and you will see the joints align. The Aim Axis specifies the axis that will point down the joint. We want to set that to X. The Up Axis specifies which axis would point upward. Set this to Y. But keep in mind that these joints are not oriented to point the Y axis along the World Y axis, we will set the World Up Dir (ection) to force the Y axis to point the way we want. Press the Z button next to the World Up Dir field, and uncheck Auto-Guess Up Direction. This will cause the Y axis to point toward the front. If we were to enter –1 in the Z field, the Y axis would point toward the back.
Switch to the Rotate tool (press e) to rotate the spine forward. You will see all the joints bending forward predictably. This is good. Press z to undo this motion.

Lastly, press the Hide Axes button to hide the LRAs of the Spine joints. There, that wasn’t so bad, was it?

**Orienting the Neck, Head and Jaw Joints**

Once you begin to get the feel of doing this, it becomes much easier. So let’s orient the head, neck and jaw joints as follows:

1. Select the neck joint. (That is the one right above CSpine).
2. Press the Select Hierarchy button.
3. Set the cometJointOrient window
   Aim Axis: X
   Up Axis: Y
   World Up Dir 0.0 0.0 1.0
   Auto-Guess unchecked
4. Press Orient Joints and you should see them oriented as shown at left.
Hmm, this worked for the Neck and Head joints, but the face and jaw joints have the Y and Z axes facing the wrong direction. Let’s change that as follows:

5. Next Select the FaceBase joint.
6. Press Select Hierarchy.
7. This time set the cometJointOrient window as shown. Basically, just check the Reverse box next to the Up Axis. This will reverse the direction of the Y axis. Taking the Z axis with it.
8. Press Orient Joints and the Y axis will flip flipping the Z axis as well!
9. Select the Neck joint and choose select hierarchy.
10. Click the Hide Axis button
That’s much better! Actually, this step will allow one of the later scripts to automatically rig the jaw to open and close, so it is quite important.
Orienting the Arms

One of the most important things to remember when rigging is that all your controls should have your Translate, Rotate and Scale values zeroed out. With joints, you can’t have the Translates zeroed out, but Rotates must be. The good news is that cometJointOrient zeros Rotations out easily, and that is what we must do for the arm. Let’s do that as follows:

1. Select Osw_DEFCollar_L.
2. Hold down the shift key and select the Shoulder, Elbow, Forearm, Wrist and Palm joints.
3. In cometJointOrient, press the Show Axis button.
4. Using the settings shown at left, orient the joints properly. The only difference here is that we have set the World Up Dir to be 0.0 1.0 0.0 by clicking on the Y button on that line.
5. Press Hide Axis to hide the LRA.
6. Repeat steps 1-5 for the Right arm, but this time, click the Reverse box for both the Aim Axis and Up Axis lines. This will orient the arms properly for this side.

This should be getting a little easier. Now let’s move on to the fingers.
Orienting the Fingers

Since we are already on this side, and the cometJointOrient box is set correctly, let’s start on the right hand. Orient the finger joints like so:

1. Select the Thumb; then shift select the Index, Mid, Ring and Cup (not Pinky) finger joints.

2. Press the Select Hierarchy button.

3. Press the Show Axis button

4. Click on the Orient Joints button in the interface to remove all rotations from the joints.

5. Click the Hide Axis button.

6. Repeat steps 1-5 for the left hand but use the settings shown (Uncheck the Reverse boxes)
Orienting The Legs

1. Select both Hip joints.
2. Click the Select Hierarchy and Show Axis buttons.
3. Press the Z button in the World Up Dir line.

You are done! Save your file!
**Good News! Bad News!**

Let’s get to the bad news first: from laying the first joint to finishing the Orientation process should take you no more than 30-45 minutes if you are concentrating. Yup! That is less than an hour.

The good news is that as you practice this part of the setup process, you will get better at it. You will get faster and more accurate the more times you try it.

More good news: This will be the most time consuming part of the process! For the most part, I have reduced the rest of the setup process to simply selecting the right joints and pressing the right buttons! Ultimately, it is the time you take to accurately complete the first parts of this process that make the speed of the last parts possible.

Even more good news: prior to the advent of ekCharacterToolKit, the rest of the setup process would take 6-8 weeks! It was HARD work! So now let’s see how you learn to take advantage of these tools!

**Rigging Using ekCharacterToolKit.mel**

One big unknown I faced when writing these tools is that I do not know the extent of your animation knowledge, nor do I know how you have been trained to animate. There are many styles of rigs. Often, an animator learns to animate on one type of rig, but then must make a switch. For example, some rigs require the animator to key joints directly in a Forward Kinematics style of animating. Others use IK and key the IKHandle of that piece of software directly.

Some rigs are extraordinarily complex with controls for everything! I have always wondered how animators actually animate with such rigs. For the purposes of learning to animate bipedal characters, I believe such complex rigs are counter productive. That said, I do follow certain principles:

- **Key as few joints as possible**—In fact, only the Collar joints are deformation joints that are also named and animated as control joints. All other joints in the DEF skeleton are controlled by some other control. This controlling of animation attributes is called *abstraction*.

  **Abstraction**—This term refers to controlling as many objects and attributes with as few animated controls as possible. One of the best examples is the face. Animating the face requires moving blend shapes, joint animation and cluster influences. These deformers can be and are collected in one controlling object that contains attributes that drive these deformers. Creating a smile can require multiple deformers that are driven by a single attribute called “Smile”. This is the essence of abstraction.
Circles for FK controls; Squarish shapes for IK controls - The shape of the controller should tell the animator what kind of control they are selecting. NURBS curves are not renderable and can be hard cornered or smooth lines on screen that the animator can select easily.
Creating the `<Character>` Group

The top node of the rig is the Group node. It consists of a simple NURBS circle that has the root joint parented under it. It is used to change the direction of a walk, moving the animation to a new location and even scaling the entire rig. The attributes are not keyed unless absolutely necessary. Let’s create the Group node as follows:

1. If it is not up already, bring up ekCharacterToolKit as we did earlier.
2. Select the DEFRoot joint.
3. Press the “Create Group node” button.
4. There is a Selection Confirmation window, click Okay. If you hadn’t selected the Root node, you could press Cancel and get out of it, but you did.

There you go! Did I lose anyone?
Rigging the Legs with an FK/IK Switch

This script accepts the selection of the Hip, Knee, Knee2, Ankle, Foot and Toe joints and rigs both an FK and IK leg allowing you to switch between the two types of rig. You have to do each leg. Rig the legs as follows:

1. Select the Osw_DEFHip_L joint.
2. Type “select –hi;” sans quotes in the command line and press enter. Alternately, shift select in order the Knee, Knee2, Ankle, Foot and Toe joints.
3. Click the Rig FK/IK Leg button.
4. The script will ask you if it is the left or right leg. Click the Left button.

Okay, this doesn’t look right, but it is! The leg joint is pointing the wrong way. It is the result of the “No-Flip Knee” technique that this script sets up. We’ll fix it as follows:
5. In the Channel Box, you will notice an attribute on the selected Osw_AnsleCTRL_L called “TWoffset”. Click on the name as shown at left.

6. In the perspective viewport, MM click hold and drag to the right to change the attribute. You will see the magenta-colored knee swing around until it lines up with the other joints. It will do so when TWoffset reaches 90.

7. Lastly, click back on TWoffset to select the attribute.

8. RM click on TWoffset again and choose “Lock and Hide Selected” from the popup menu.

9. Repeat steps 1-8 for the right leg.
The FKIKLegSwitchCTRLs have 3 attributes. The first two turn the visibility of the FK and IK controls off. The third switches between the two rigs. 0 sets the rig to FK while 1 sets the rig to IK.

99 times out of 100 we will animate using IK controls, so 1 is the default for this control. For now, go ahead and hide the FK controls by setting FKVis to 0.

Later on in the setup process, we will be using the FK controls to set up a “Weighting Dance” animation, but we can get them out of the way for now.

Also, note that there is a “Rig IK Leg” button right above the “Rig FK/IK Leg” button. This rigs a single knee IK leg with a Pole Vector controller. The “No-Flip” knee created by the latter is a better and more easily controlled solution for the leg.
Rigging the Arms

Rigging the arms is probably the simplest of all, but as always, it must be done exactly as the script needs to make it work. Fortunately, this is what this manual is for! Lets rig the arms as follows.

1. Select the Osw_DEFShoulder_L joint.
2. Shift select in this EXACT order: Osw_DEFElbow_L Osw_DEFForearm_L Osw_DEFWrist_L
3. Click the Rig Arm button. You will get a confirm screen telling you to select the right joints and to make sure they are named correctly, Click OK.
4. You will be asked which arm. Click on Left in this case.
5. You will then be asked to Link the Wrist and the Palm joints. This will split the rotations between the two joints to smooth out the difficult deformations in the wrist area. If you have aligned the LRAs of the wrist and palm joints, click Yes. If not, click No. If you are unsure, click Yes, then test.
the motion of the wrist/palm hierarchy and undo if it doesn’t work.

Upon clicking Yes, you will see the distinctive round circles that denote FK controls. Much of the time, we will manipulate the arms with FK controls, so FK is the default setting for the FKIKSwitch control.

Take a moment and rotate the Wrist controller in X, Y and Z. You should see the palm joint rotate also. If not, or if it rotates in an unexpected fashion, immediately undo until the FK controls vanish, reselect the arm joints and rerun the script. This time click No when the script asks you if you want to link the Wrist and Palm joints.

We now want to adjust the Control Vertices of the NURBS curves to suit us better. I generally like to change the shape of the Collar controller to indicate that it is different from the shoulder controller. Let’s do this as follows.
1. Select the Osw_DEFCollarCTRL_L control. NOTE: This control is the only joint that is both a deformation AND a control.

   You will see the entire arm light up.

2. Use the Right Mouse button (RMB) or hold down the CMD button and click on the NURBS circle. From the pop up menu that results, choose Control Vertices.

   This will allow you to edit the shape node of the control. You can change the shape of the controller while the transform node remains in place.

3. Uncheck the Joint pickmask button above the shelf tabs. This will prevent you from accidentally selecting joints.
4. Using the selection tool (press Q) LM drag a rectangular marquee around the Collar control CVs, this will select them.

5. Using the rotate tool (press E), rotate the CVs 90 degrees along the Y axis. (That is the green ring). (Left image)

6. Use the move tool (W) to move the circle above and outside of the shoulder/arm area. (Middle image)

7. Select and move individual CVs until you get a shape like the one shown in the right image. Be careful not to accidentally select the CVs of the shoulder control. (But if you do, it doesn’t effect the control itself.)

8. Last, press F8 to go from object mode to component mode. Repeat this entire procedure for the right arm.
Rigging The Spine

The spine represents the most complex part of the figure to rig, even with ekCharacterToolKit. The reason for this harkens back to the aforementioned automated ease-of-use vs. ability-to-customize-to-meet your-needs. It is possible to create a script that will give you a perfectly rigged spine with all the accompanying controls, but the controls might not be placed appropriately for your character. On the far other hand, rigging the spine from scratch would give you absolute control over your spine, but would take most of the semester to learn. The two stage process that rigging the spine represents in this script is what I hope is a happy medium. So, read all the instructions carefully, take your time, and let’s rig the spine as follows:

1. Make absolutely sure that the Select by Object Type: Joints button is active. You have to be able to select joints here!

2. In the “sel” field, LM click and hold on the inverted triangle and choose Quick Selection from the pop-up menu.

3. Type “*DEFSpine*” sans quotes in the field. This should correspond to how you named your spine earlier. Click enter to select all the spine joints except the CSpine and Root joints. These should NOT be selected.

An alternate method of select is to carefully shift select up from the first spine joint to just below the CSpine joint in the perspective window.
Even MORE alternatively, you can open the hierarchy in the Outliner and select the joints directly. It is imperative that you select from DEFSpin to DEFSpine in order, otherwise the script won’t work.

4. In the ekCharacterToolKit interface, click on the “Rig Spine” button. You will see a confirmation screen reminding you not to select the root and Cspine joint. Click OK.

You will then see this window which I included as a reminder of what to do next. And that is what we will do next, so click OK.

It doesn’t look like much happened, but the spine has been rigged with a Stretchy Spline IK solver. The joints will stretch and rotate to follow the curve which you can barely see in the perspective window. The guts of this rig are somewhat complicated. What we will do now is create a control structure that is easy to select and manipulate.

1. In the lower right of the interface, click on the Create New Layer button as shown. This will create a layer probably named “layer1”.

2. Double click on layer1 and rename it to “ControlsLyr”
3. Select the Osw_DEFRoot joint.

4. RMB click on the ControlsLyr and choose Add Selected Objects. This places the entire joint hierarchy on the ControlsLyr.

5. Click twice in the middle box next to ControlsLyr to set it to Reference. This will make all its contents unselectable, but still usable for our purposes.

6. In the side view, go to Skeleton>Joint Tool. Create a 4 joint structure as shown at left. Hold down the v key to snap place these joints right on the underlying spine.

   Place these joints where you think your character should be able to bend. For example, UpBack should be placed where you would expect the rib cage to be able to bend.

7. Name them as shown at left.
8. Create a two joint structure as shown at left. Begin by going to **Skeleton>Joint Tool**. Then clicking on Osw_BackBase to begin from that joint. Then hold down the `v` key and click at the location of Osw_MidBackJNT. Lastly `v` click at the DEFSpine location and press Return to finish the chain.

These joints are going to provide the framework for our spine control structure. But first we have to orient them as follows.

1. Select Osw_BackBase. You will see the entire control structure turn green.

2. Press the Down arrow on the keyboard. This moves down one joint in the hierarchy. Osw_MidBackJNT should be selected. (You can also select it manually.)
3. Select Comet Joint Orient in the ekCharacterToolKit interface.

4. Press the Select Hierarchy button and the Show Axis button.

5. Set the Aim Axis to be X
   Set the Up Axis to be Y
   Set the World Up Dir to be 0.0, 0.0, 1.0
   Uncheck Auto-Guess Up Direction
   Press Orient Joints.

You can see that the upper part of this joint structure is oriented properly. Hide these axes by pressing the Hide Axis button in the Comet Joint Orient interface.

Let’s do the hips as follows:
1. Repeat steps one and two from the previous page.
2. This time, press the right arrow. This switches between children in a hierarchy. Osw_HipsTopJNT should be selected.
3. Select Hierarchy and Show Axis.
4. Orient the joints with the setting shown at left.
5. Click the Hide Axis button. Now your control structure will rotate predictably.

**Binding The IK Spline Curve**

Okay, up to now, I've asked you to do pretty much what you have done before. Placing, naming and orienting joints should be pretty familiar. But now we are going to bind this control structure to the spline curve that drives the bending of the spine. It will involve some numbers in a spreadsheet type format, but what you will have when you are done will be pretty cool. Much of the selection can be done using the outliner, So let’s begin as follows:

1. Click the Persp/Outliner button in the view shortcuts section of the toolbar.
2. Close Osw_Group hierarchy by clicking in the little box with the minus sign to the left of the name.
3. Open fully the Osw_Ba ckBase hierarchy by Shift-clicking on the little box with the plus sign to the left of the name. You should see all the joints listed below.

4. Hold down the shift key and drag select from Osw_MidBackJNT down to Osw_HipsEndJNT. This selects the entire hierarchy.

5. Hold down the command key and select Osw_back_curve. This is the SplineIK curve that controls Oswaldo's spine.

6. Go to Skin>Bind Skin>Smooth Bind Options. Set the options as shown at left. We will discuss these options later when we bind the character to the skeleton.

Click Bind Skin.
Now we need to weight the individual Control Vertices (CVs) of the back_curve so that the curve will deform properly when the joints of the control structure are rotated.

1. Select Osw_back_curve in the outliner.

2. Press the F8 key to enter component selection. In the image at left, I have hidden joints so you can see the CVs better.

CV’s along a curve are indexed beginning with 0 and moving up to the final number. Each of these points is being influenced by one or more of the joints we just created. We need to adjust that influence.

3. Go to Windows>General Editors>Component Editor...

4. Marquee select all the CVs.
You should see all the CVs loaded into the Component Editor. If it is not selected, select the Smooth Skins tab to bring it forward.

The CVs that make up the curve are listed at left. The Joints that are influencing these points are listed across the top. The numbers in the columns represent the amount of influence each joint has. We need to adjust these numbers. Basically, make them look like the table below:

![Component Editor](image)

You actually have more leeway then it seems. But you MUST have CV[0] entirely weighted to Osw_HipsEndJNT AND have CV[4] entirely weighted to Osw_ShldrJNT.

The way it works generally is to specify the weighting of the CV based on its proximity to a particular joint. For example, CV[1] is closer to Osw_HipsTopJNT than it is to Osw_HipsEndJNT, so it should be weighted a little higher than .5, in this case, I chose .6. Given that the total weight needs to be 1, the weight of HipsEnd should be .4.
After all this, finishing the spine will seem anti climactic:

1. Select Osw_BackBase.
2. Click the Finish Spine button in the ekCharacterToolKit interface.

Yes, it is just that simple!
Rigging the Neck and Head

Rigging the neck and head is simplicity in itself. (Well, compared to doing the spine, anyway). Let’s rig them as follows:

1. Start by selecting the joint right above Osw_DEFSpine. This should be Osw_DEFNeck.

2. Shift select each of the joints in the neck and end with Osw_DEFHead as shown at left.

3. Press the Rig Head and Neck button in the ekCharacterToolKit interface.

You have two new controls. Osw_HeadCTRL controls how the head rotates. It has a follow control allowing it to follow either the neck controller (0) or the world (1). When set to world, the neck will rotate independently of the body and remain fixed on a specific point in space.

Osw_NeckCTR controls the rotation of the neck. The custom attribute CurvyAdd adds the rotation of the neck joint to all the joints in the neck allowing the animator to create a very sinuous neckline.
Rigging the Fingers

This is the ekfingerRiggerUI interface properly loaded from the ekCharacterToolKit interface with the joint hierarchies and controller loaded into their proper fields. It is easy to use. Simply select the NURBS curve or other object that will carry the control attributes and press the <-Put Controller button. That loads the name into the field.

Loading the fingers is even easier. Just click on the knuckle joint and press the <-Put <joint> Hierarchy button. This loads the selection and the two child joints into the field. If the child joints are not the rotation joints, you can load individual joints by selecting the individual joints and pressing the appropriate button.

If your hand doesn’t have 5 fingers, simply don’t use the fingers that are missing.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Thumb</th>
<th>Index</th>
<th>Middle</th>
<th>Ring</th>
<th>Pinky</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Osw_FingCTRL_R</td>
<td>Osw_DEFThumb2_R</td>
<td>Osw_DEFThumb3_R</td>
<td>Osw_DEFIndex3_R</td>
<td>Osw_DEFMid_R</td>
</tr>
<tr>
<td></td>
<td>&lt;- Put Controller</td>
<td>&lt;- Put Thumb Knuckle</td>
<td>&lt;- Put Thumb Joint2</td>
<td>&lt;- Put Index Knuckle</td>
<td>&lt;- Put Index Joint3</td>
</tr>
<tr>
<td></td>
<td>Osw_DEFThumb2_R</td>
<td>Osw_DEFIndex2_R</td>
<td>Osw_DEFIndex3_R</td>
<td>Osw_DEFMid2_R</td>
<td>Osw_DEFMid3_R</td>
</tr>
<tr>
<td></td>
<td>&lt;- Put Index Joint2</td>
<td>&lt;- Put Index Joint2</td>
<td>&lt;- Put Index Joint3</td>
<td>&lt;- Put Middle Joint2</td>
<td>&lt;- Put Middle Joint3</td>
</tr>
<tr>
<td></td>
<td>Osw_DEFMid2_R</td>
<td>Osw_DEFMid3_R</td>
<td>Osw_DEFRing2_R</td>
<td>Osw_DEFRing3_R</td>
<td>Osw_DEFRing4_R</td>
</tr>
<tr>
<td></td>
<td>&lt;- Put Middle Joint2</td>
<td>&lt;- Put Middle Joint3</td>
<td>&lt;- Put Ring Knuckle</td>
<td>&lt;- Put Ring Joint2</td>
<td>&lt;- Put Ring Joint3</td>
</tr>
<tr>
<td></td>
<td>Osw_DEFRing2_R</td>
<td>Osw_DEFRing3_R</td>
<td>Osw_DEFRing4_R</td>
<td>Osw_DEFPinkR</td>
<td>Osw_DEFPink2_R</td>
</tr>
<tr>
<td></td>
<td>&lt;- Put Ring Joint2</td>
<td>&lt;- Put Ring Joint3</td>
<td>&lt;- Put Pinky Knuckle</td>
<td>&lt;- Put Pinky Joint2</td>
<td>&lt;- Put Pinky Joint2</td>
</tr>
<tr>
<td></td>
<td>Osw_DEFRing3_R</td>
<td>Osw_DEFPink2_R</td>
<td>Osw_DEFPink3_R</td>
<td></td>
<td>&lt;- Put Pinky Joint3</td>
</tr>
<tr>
<td></td>
<td>&lt;- Put Ring Joint3</td>
<td>&lt;- Put Pinky Joint2</td>
<td>&lt;- Put Pinky Joint3</td>
<td></td>
<td>&lt;- Put Pinky Joint3</td>
</tr>
</tbody>
</table>

Rig Fingers | Close
It is CRUCIAL, however, that the joints of the fingers be oriented as shown at left with the X axis pointed down the joint, the Z axis pointed to the back of the character and the Y axis pointed down.

When done, press the Rig Fingers button. When prompted, specify Left or Right hand and the script does the rest!

This is a huge timesaver! But there is one problem, the Thumb’s Z axis is oriented perpendicular to the direction of the joint, we need to move this off axis so that it is more perpendicular to the hand. Let’s do this as follows:

1. Select Osw_DEFThumb1_L.

2. Go into Component Selection mode, RM click on the ? mark and choose Local Rotation Axes. This will display and allow you to manipulate the LRAs for the joint.
3. Rotate the LRA so the Z axis points parallel to the hand and the Y axis points straight down. Now the Thumb controls will rotate the thumb properly.

Repeat these steps for the right hand

You must make sure that for the right hand, the X axis is pointed UP the joint axis instead of down it. Z must be pointed to the back of the character and Y must be pointed down.

Lastly, rotate the LRA so the Thumb rotates properly. Point Z parallel with the hand.

You may repeat this procedure for the Cup joint as well.
Rigging the Cup and Spread Attributes

The interface automates much of the tedious process of rigging fingers, but I left out the cup and spread attributes because learning how to rig them with Set Driven Key is an important thing to learn, let’s do this as follows.

NOTE: These steps use a character called “Smiley”, so all the joints are labeled “Smil” instead of “Osw”. Other than that, everything else is the same!

From the picture at left, we can tell that the axis we will want to drive with the Cup attribute will be the X axis, while the Spread axis will drive the Y axes on the knuckles. Let’s start with the Cup attribute.

Before starting choose **Window > Settings/Preferences > Preferences**... and set the Default In and Out tangents to Clamped in the Animation category.

1. Select Smil.DEFCup_L.
2. In the Channel Box, select rotateX.
3. RM Click on the rotateX attribute and choose Set Driven Key...
4. Select Smil_FingerCTRL_L and choose Load Driver.
5. Choose the Cup attribute on the right side of the window.
6. Press the Key button to set the SDK relationship between the two attributes.

7. Set Smil_FingerCTRL.Cup to –10.
8. Set Smil_DEF.Cup_L.rotateX to 30.
9. Press the Key button to set the key.
10. Set Smil_FingerCTRL.Cup to 10.
12. Press the Key button to set the key, and you are finished with this attribute.

Now let’s set the Spread attribute on multiple joints.

We know that the rotateY attribute will be used to key the Spread attribute, but the index finger will rotate one way while the ring and pinky fingers will rotate the other way. Also, the proper rotations for each finger will depend largely on the design and construction of your character, so we will be doing the next group visually.

1. Select Smil_DEFIndex1_L, Smil_DEFMid1_L, Smil_DEFRing1_L and Smil_DEF.Pinky1_L.
2. Press the Load Driven button in the SDK window.
3. Select all four joints and the rotateY attribute in the Driven field and the Spread attribute in the Driver field.

4. Press the Key button.

5. Click Smil_FingerCTRL_L in the Driver field. This selects that object.


7. Rotate ONLY the knuckle joints to spread the fingers as shown at left.

8. Then, in the SDK window, make sure all four joints are selected along with the rotateY attribute and press the Key button.

9. Set the Spread attribute to 10. You should see the fingers go back into their default positions. If you don’t, select all four finger joints, select the rotateY attribute and set it back to 0. Then select the attribute name RM on it and choose Break Connections. Set the Spread attribute to 0, then start over.
10. But if the fingers DID go back into place, rotate the fingers like that shown at left.

11. Press the Key button in the SDK window, and you should be finished.

Play with the Spread attribute for a bit. You should see the fingers spread and contract as you adjust that attribute.

---

**Rigging Aim Constrained Chest and Back joints**

As an aid in weighting, you will be creating Pect joints and Shoulder blade joints in the upper torso of the rig. These will carry the deformations in the chest area as the arms are rotated forward both at the shoulder and at the collar joint. Let's do this as follows:

1. Select Osw_DEFShoulder_L

2. In the ekCharacterToolKit window, press the Rig Pects or Blades button.

3. You will see a dialogue, click on the Pect button to start the process.
4. Read the dialogue box, then click OK.
   At first it will look strange with a Locator and the joint sticking out from the chest as at left.

5. But simply use the translate tool (W) to move the locator back until the joint sinks below the chest geometry as shown in the two images above.

6. Select the Pect joint that has been created.
7. We now have to change the weight values for the pointConstraint node. Change the Osw_Chest TargLocWO and the Osw_PectInsJNT_L target values to .5 as shown above. This will center the Pect joint between the shoulder and Chest Locator.

The first use of this script automatically creates both the chest and back target locators.

8. Prior to creating the Blade joint, move Osw_BackTargLoc closer to the back of the Geometry as shown at left.

9. Select Osw_DEFShoulder_L.

10. Click on the Rig Pects or Blades button in the interface.

11. This time, choose Blade from in the interface.

12. Click OK and you have the left side done.

Now, you can repeat steps 1-12 for the right side.
Clean Up Controls