ANZOVIN
studio

THE SETUP MACHINE 3

USERS MANUAL

Maya Versions: Maya 2014 and up

Rig Construction: Raf Anzovin

Programming: Connie Hildreth

Documentation/Beta Testing: Morgan Robinson & Raf Anzovin

INTRODUCTION

The Setup Machine 3 is a plugin for Maya 2014 and later. TSM3 enables you to quickly and easily create a high quality animation rig for any character. Simply add our modular widgets to your character and click 'Rig,' and TSM3 will install a state of the art character animation rig, with stretchable, bendable limbs, anti IK pop, smart IK/FK switching, true mirroring, and more! There are also options to use lower resolution systems, suitable for export to game engines and you can generate a mocap skeleton that drives the animation rig, so you can easily work with mocap data.

As with all of Anzovin Studio's products, we created TSM3 because we wanted to use it ourselves. We hope you'll find it as useful as we do!

-The Anzovin Team

TABLE OF CONTENTS

WHAT'S NEW FOR TSM2 USERS?

TSM3 is a significant upgrade from TSM2. You'll notice that we've introduced the following major features:

- Ability to rebuild rigs
- Game engine compatibility
- Built in mocap support
- True control mirroring
- Heat Map and Geodesic Voxel Weighting options
- Joint and control tags for easy scripting
- New eye and single influence widgets
- Even more control and flexibility in existing systems
- Viewport 2.0 compatibility
- Improved rig performance

Unlike TSM2, TSM3 does not have an Influence Object layer between the joints and the character's skin. Since Maya now supports Dual Quaternion weighting, the influence object layer is no longer beneficial and simply adds unnecessary influences.

WHAT’S NEW FOR TSMG USERS?

TSM3 retains TSMG's game engine compatibility, but offers additional features that bring it up to the standards of TSM2 and beyond. New major features include the following:

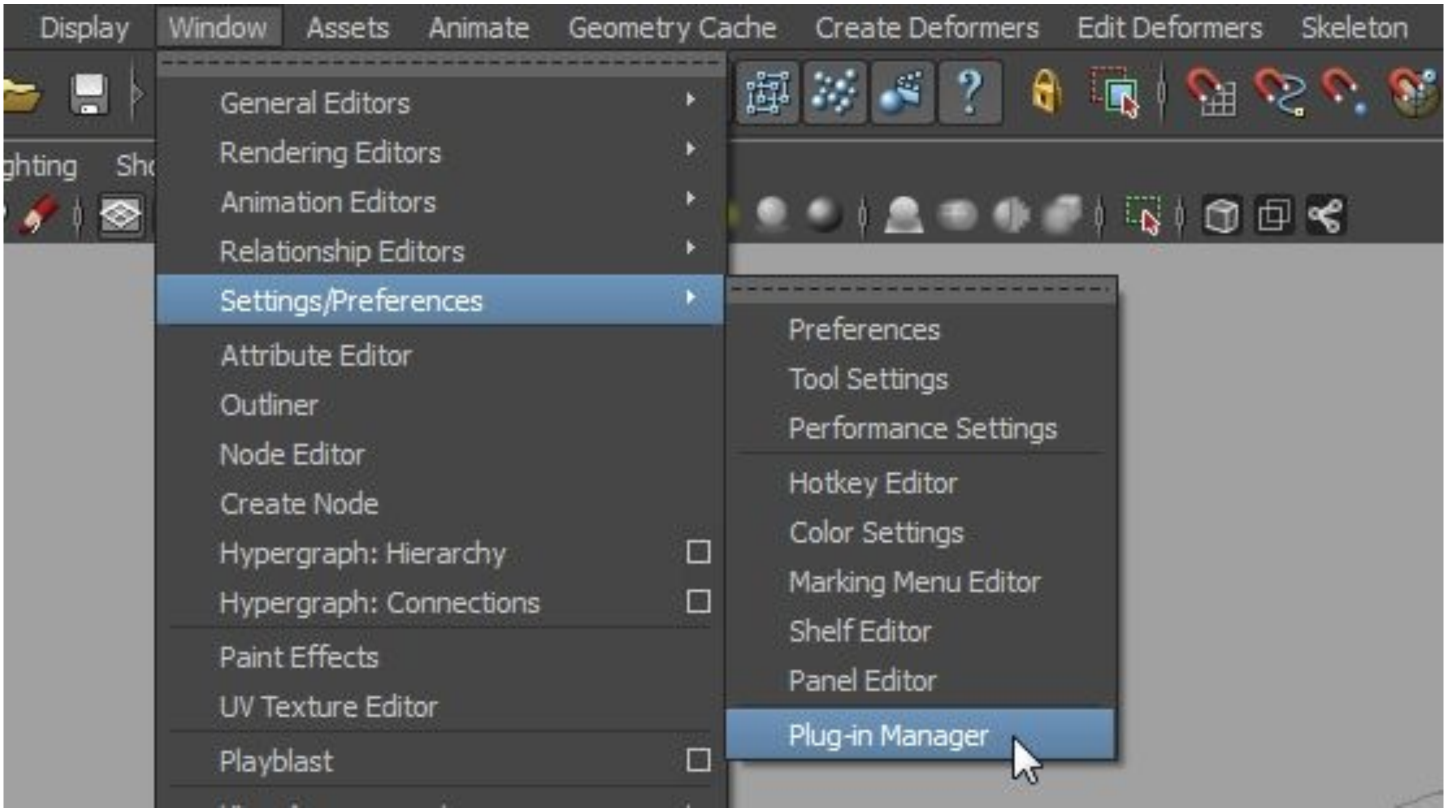
- Built in mocap support
- True control mirroring
- Joint and control tags for easy scripting
- Bendable limbs
- Ability to remove weights by body part
- New eye and single influence widgets
- Even more control and flexibility in existing systems
- Viewport 2.0 compatibility

INSTALLATION

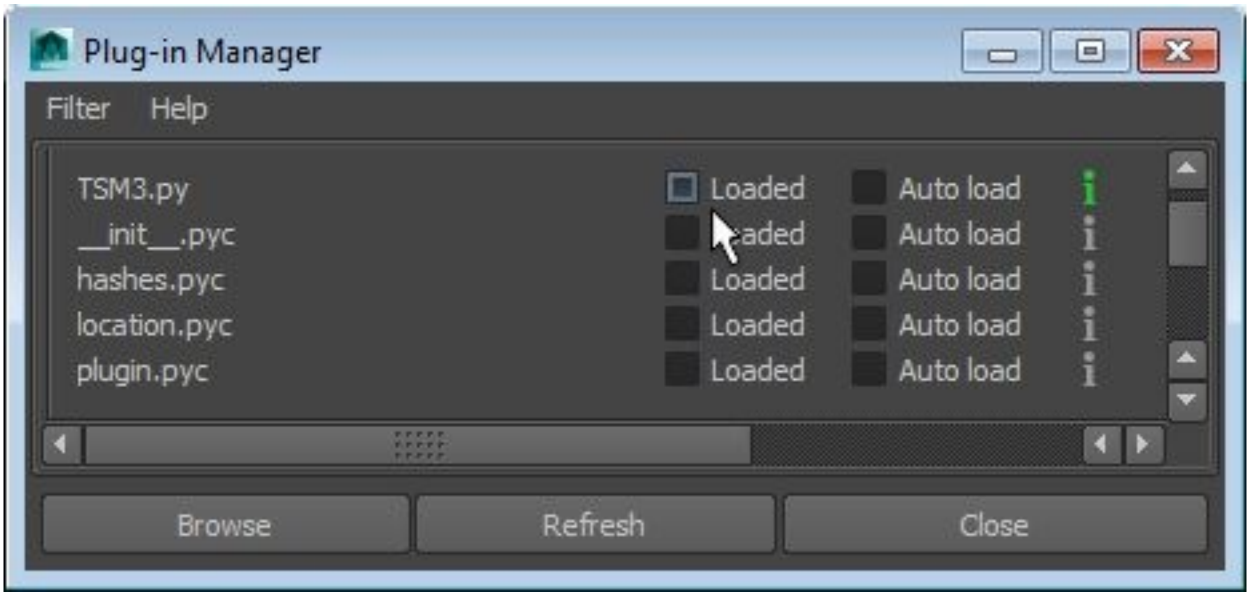
To install TSM3, first ensure that the installer for your OS (either “Install TSM3 for Mac OS X.app” or “Install TSM3 for Windows.exe”) is in the same directory as the InstallerData folder. Do not copy or move the installer to a different directory without copying or moving the InstallerData folder as well!

Next, run the installer. You will be prompted to read and agree to the license agreement, and to choose which Maya installations you'd like to install TSM3 for. Once it is installed, you will also have to load the plug-in for whichever installations of Maya you are using.

In Maya, go to Windows > Settings/Preferences > Plug-in Manager:



This will bring up the plug-in manager:



TSM3.py	□ Loaded	□ Auto load	i
__init__.pyc	□ Loaded	□ Auto load	i
hashes.pyc	□ Loaded	□ Auto load	i
location.pyc	□ Loaded	□ Auto load	i
plugin.pyc	□ Loaded	□ Auto load	i

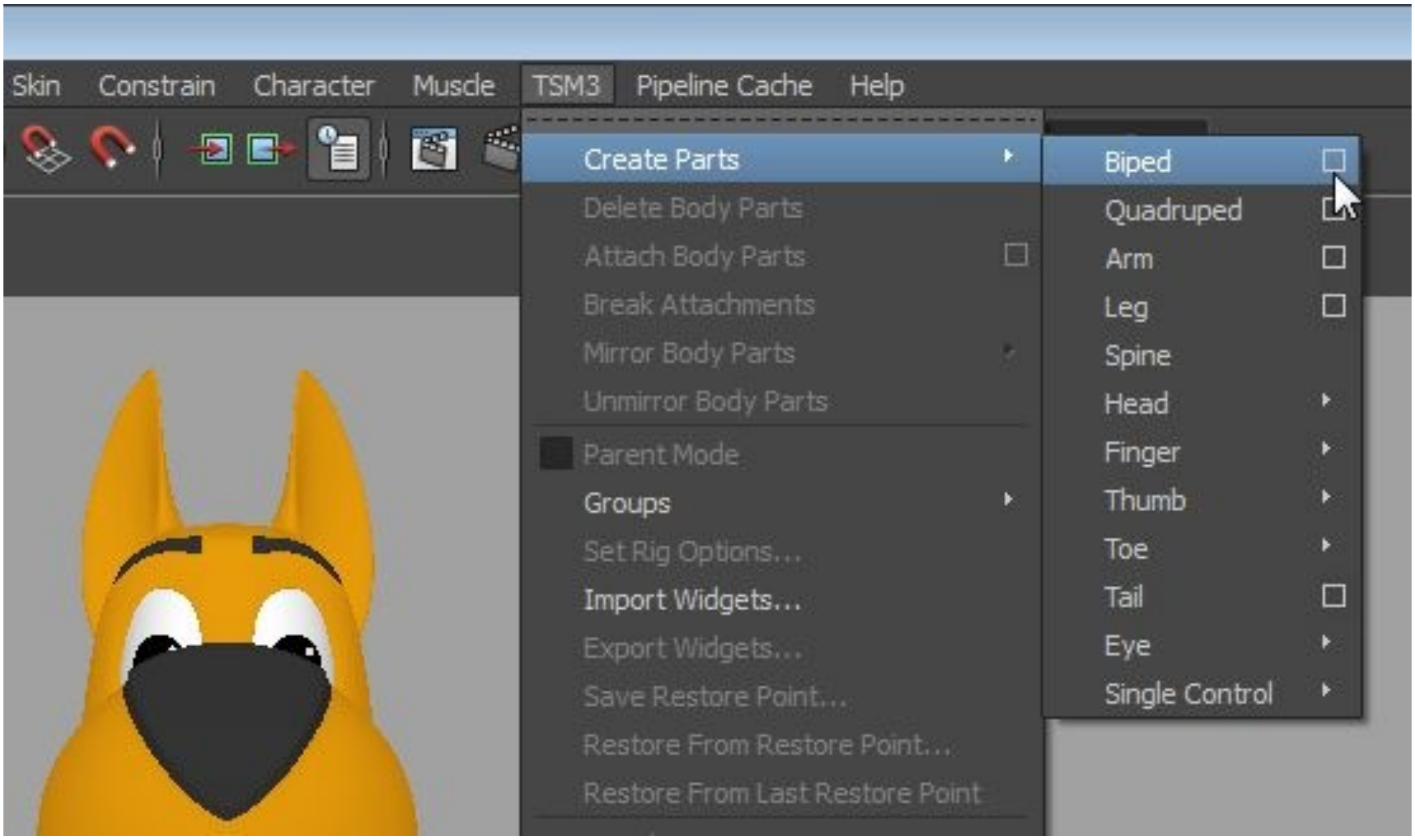
Check both 'loaded' and 'auto load' next to TSM3.py.

At this point, you will be prompted to enter your serial number. If you have purchased TSM3 and have not received a serial number within 24 hours, please contact sales@anzovin.com

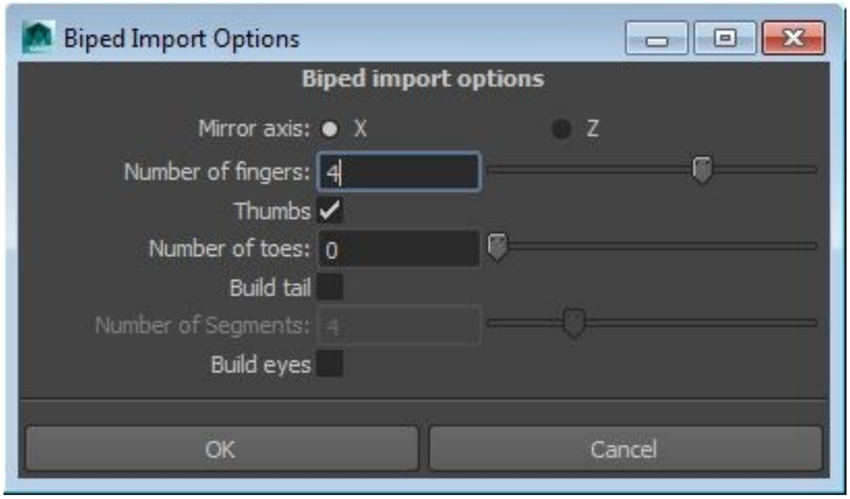
GETTING STARTED

To use TSM3, you'll need to start with a character. This character should consist of one or more polygonal meshes. To teach you the basics, we'll be using a standard humanoid, but TSM3 is certainly capable of rigging more exotic characters — see **Build Your Own** for more info.

STEP 1: CREATE AND POSITION WIDGETS



The first step is to get some widgets into your scene. Widgets are a simple representation of the joints of your rig. By positioning them to fit your character, you'll enable TSM3 to position the actual rig. To create the widgets we'll be using to rig Dennis, our example character, go to the TSM3 menu, then Create Parts > Biped. Select the box to pop up an options dialog:



Biped Import Options

Biped import options

Mirror axis: ● X ○ Z

Number of fingers: 4

Thumbs ☑

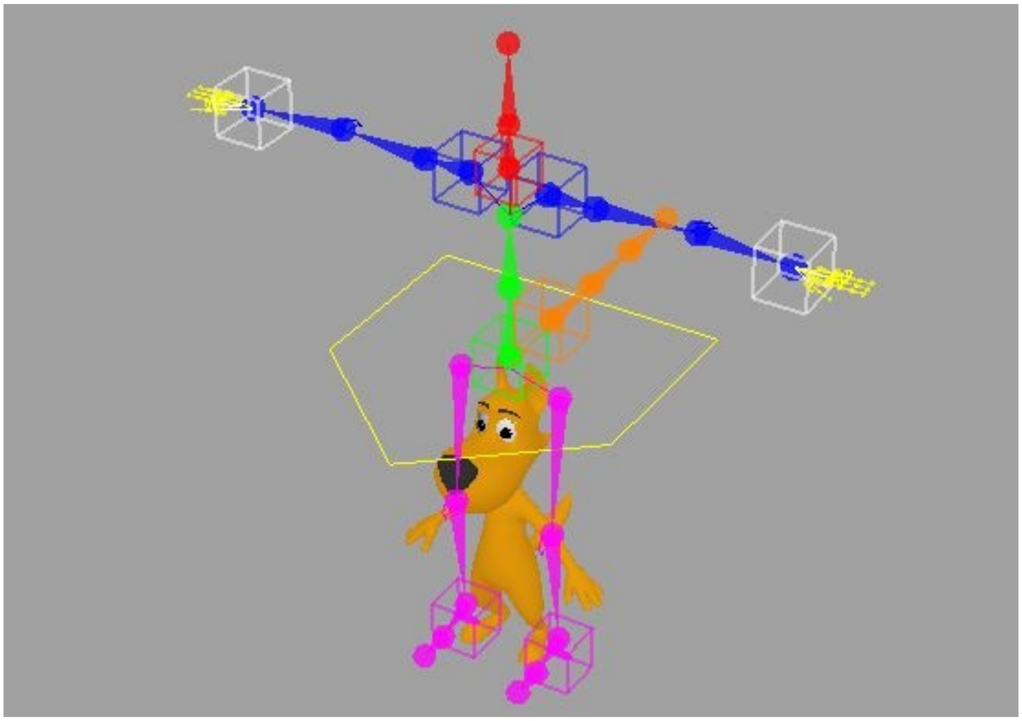
Number of toes: 0

Build tail □

Number of Segments: 4

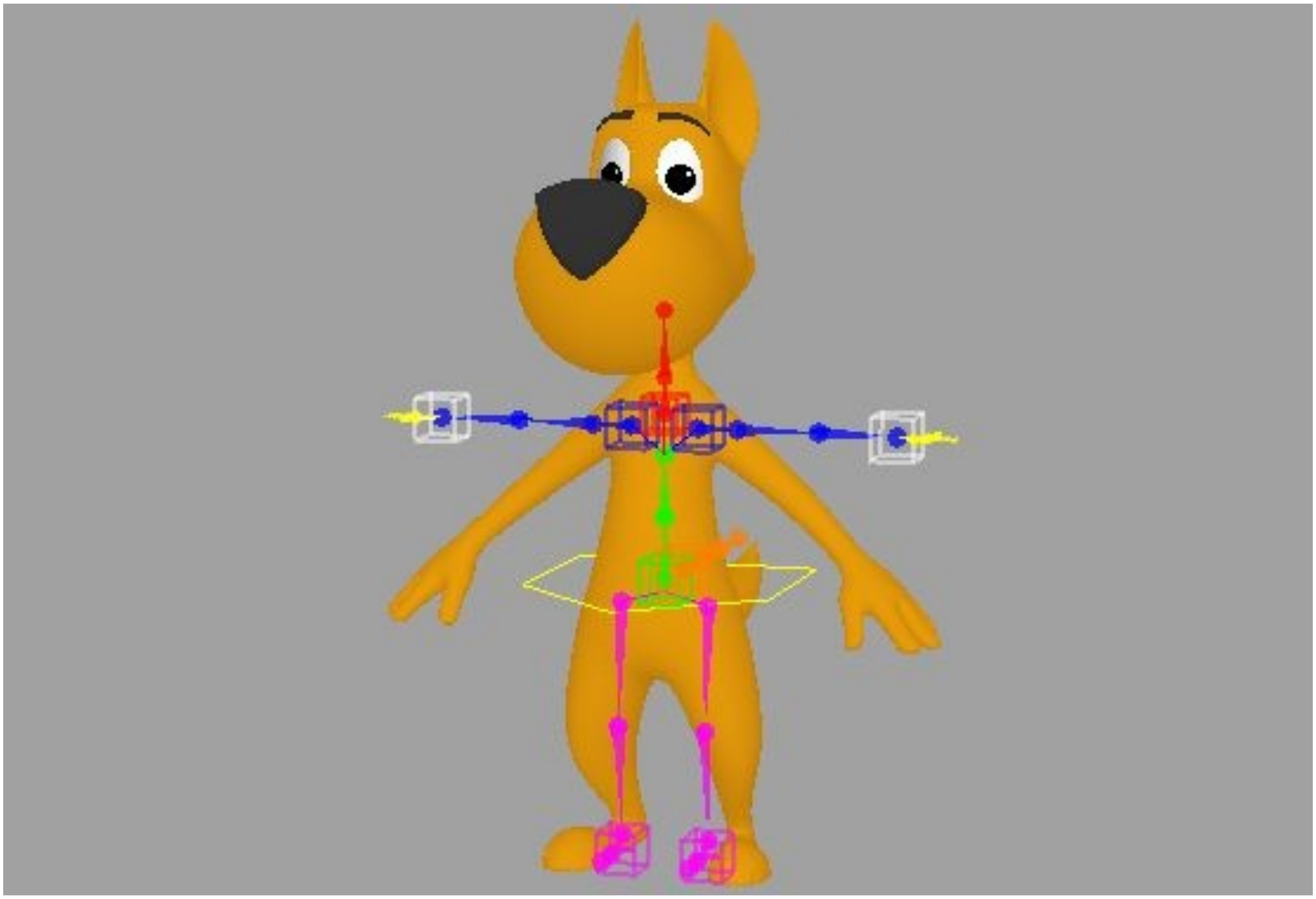
Build eyes □

Here you can choose your mirror axis, number of fingers and toes, etc. Dennis has only three fingers, but he does have a tail. It's fairly short, so after you check 'Build tail,' adjust the number of segments to three. Then click OK.



You'll get a set of brightly colored widgets in the shape of a biped. Since Dennis is smaller than the default biped scale, you'll need to adjust the widgets to fit. Start by selecting the green cube shaped control at the base of the spine, then scale and position it to get a general fit.

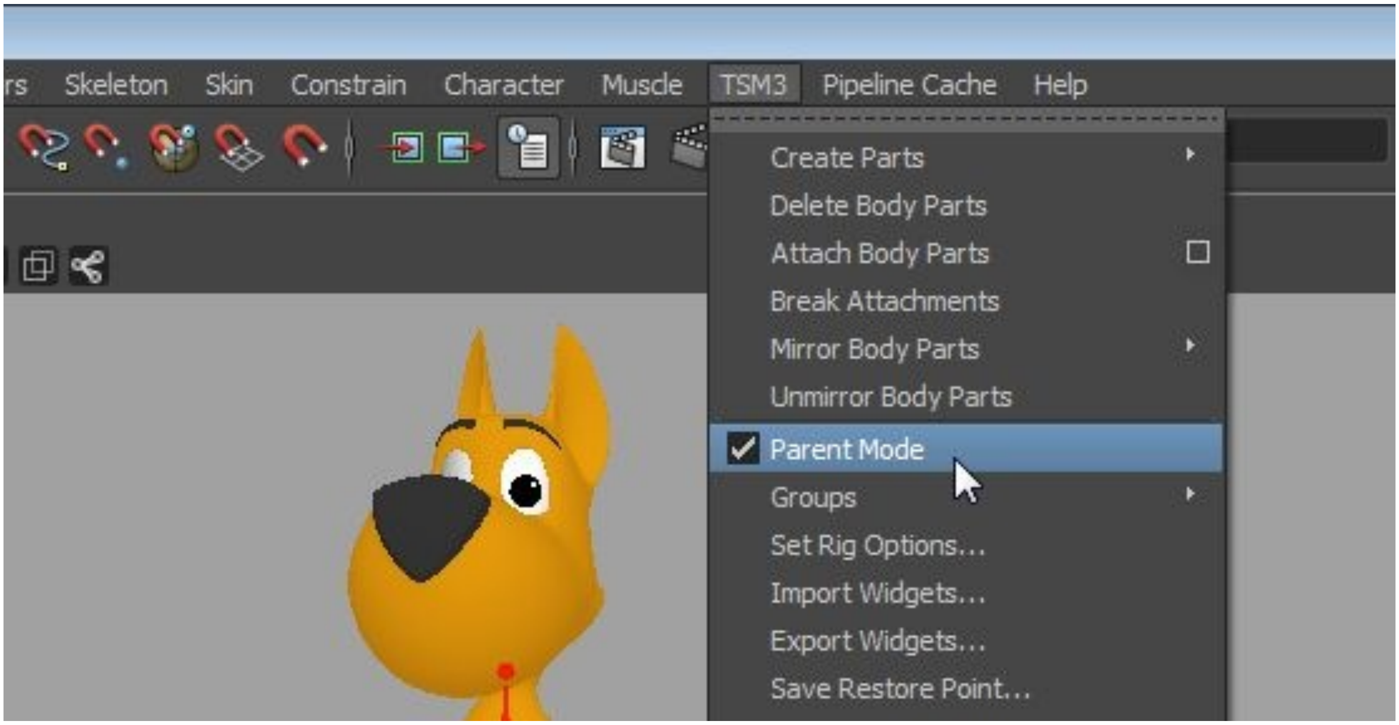
You'll also want to position the upper body control, which is the yellow house-shaped control near the center of the rig. Once the character is rigged, this will be the control for everything from the pelvis up, so it usually makes sense to position it around the pelvis:



Now, you can continue positioning widgets, using the square base controls to position an entire system, and then the smaller round controls for fine tuning. You'll find that the widgets mirror in both directions, so you can work on either side.

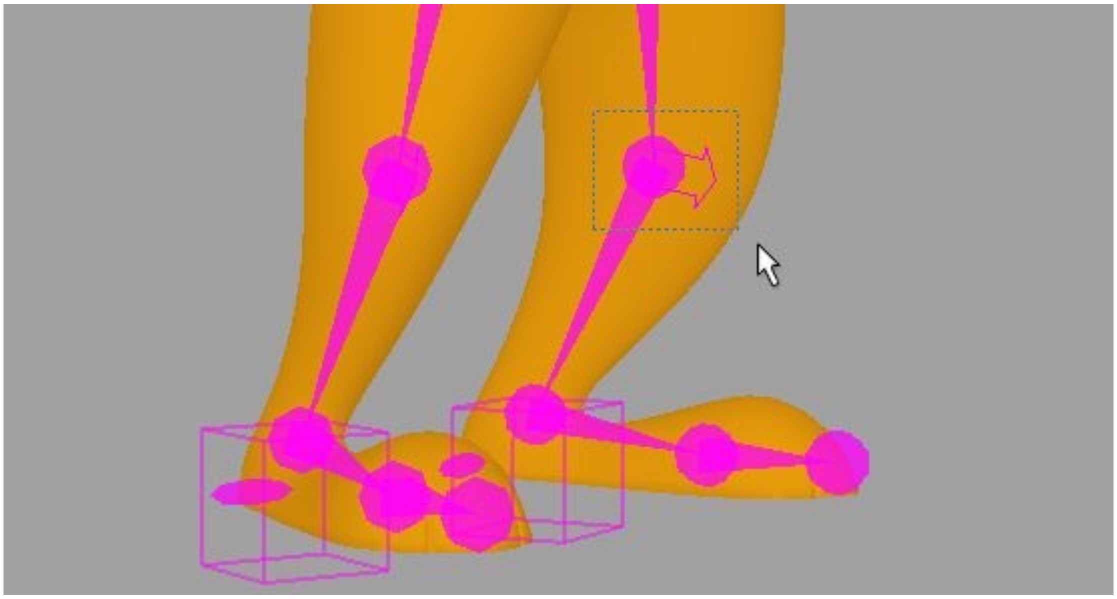
By default, 'parent mode' is on. This means that if you move or scale a widget that has other widgets attached, its 'children' will be moved or scaled as well. However, you can switch this off by unchecking TSM3 > [X] Parent Mode.

This allows you to reposition an arm, for instance, while leaving the fingers in place.



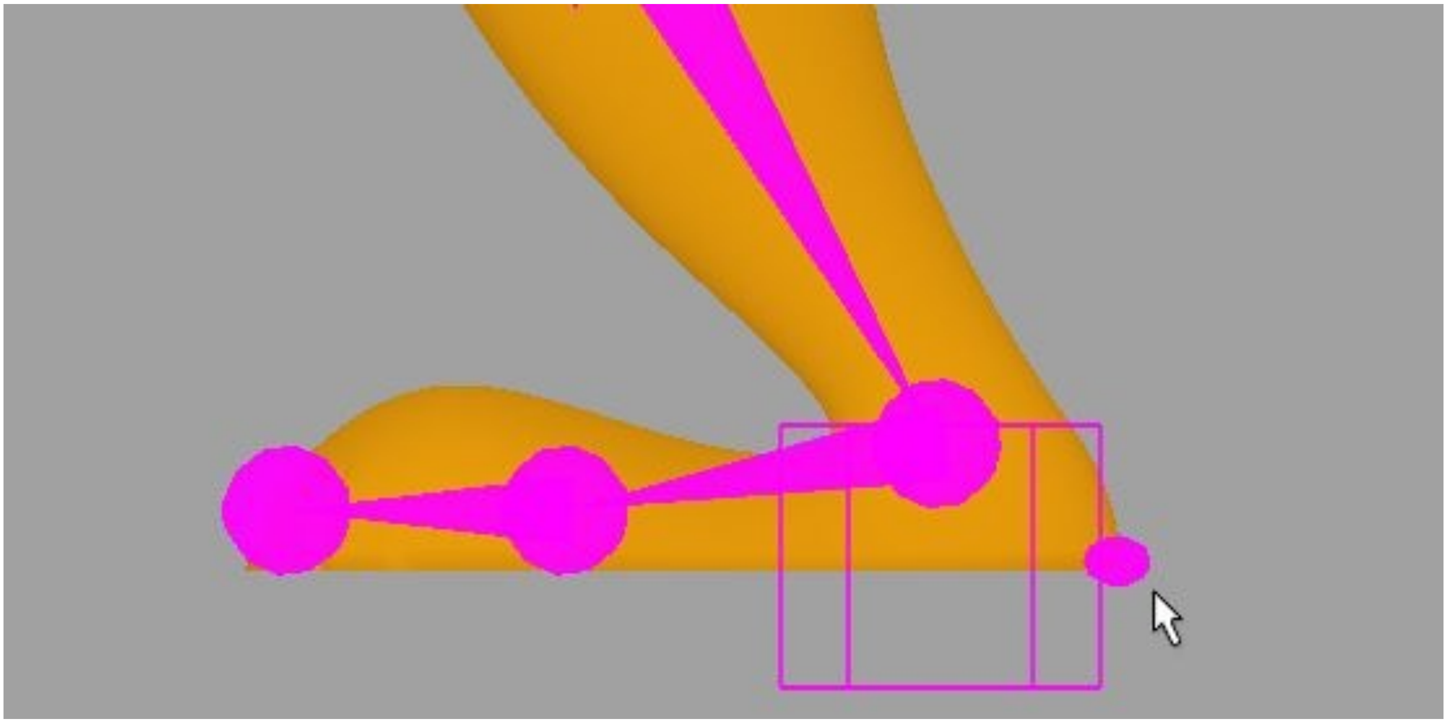
Most of the positioning you're doing should be self explanatory – the round widgets go where you want your joints to end up. However, there are a few special cases to consider.

Knee and Elbow Direction



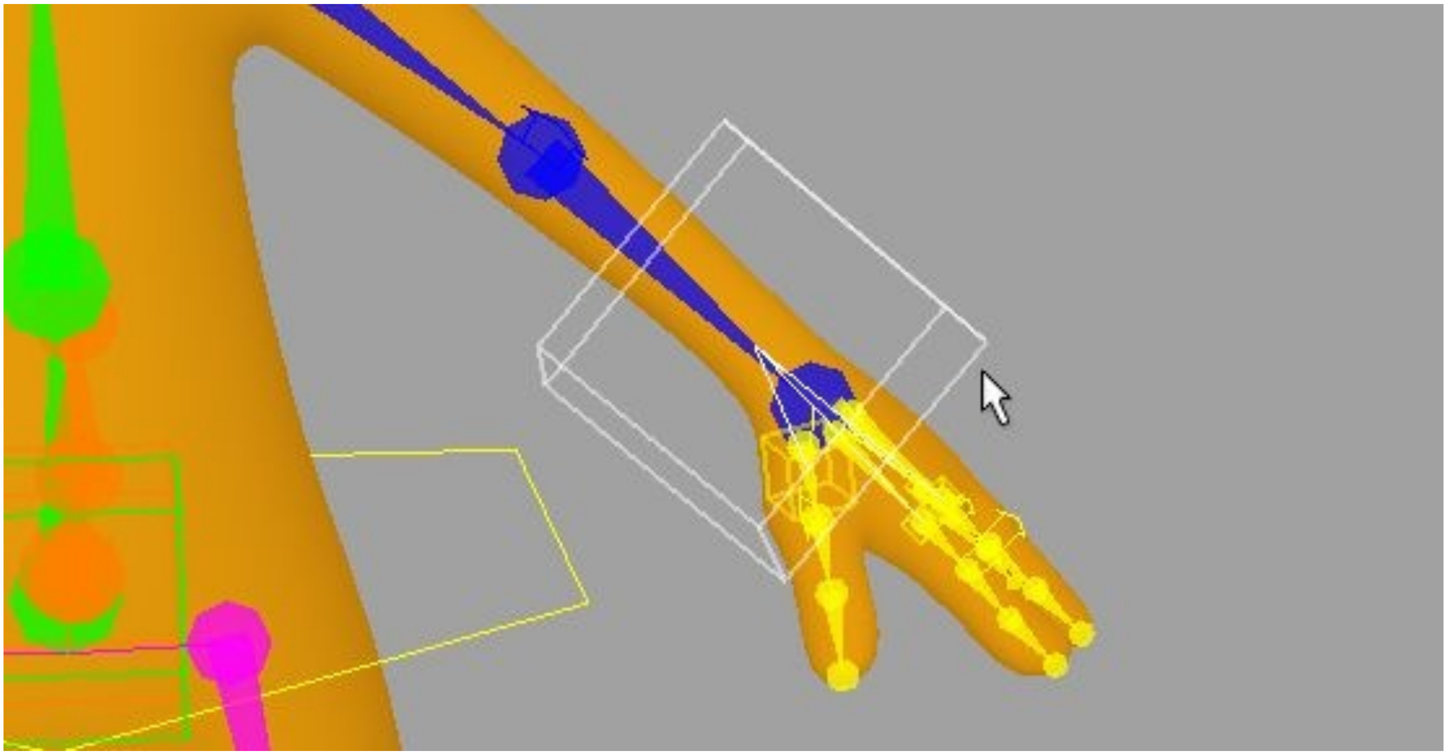
As you're positioning the leg widgets, pay close attention to the arrow that points out of the knee widget. This arrow indicates the direction that the knee will bend in. It's affected by the position of the hip, knee, and ankle leg widgets, so be sure to position them so that the knee arrow points in the correct direction. The same is true for the elbow arrow in the arm.

The Foot Control



The small widget below the ankle will determine the placement of the foot control, and should be positioned at the bottom and back of the character's foot, right at the heel, as if resting on the ground.

Using Groups



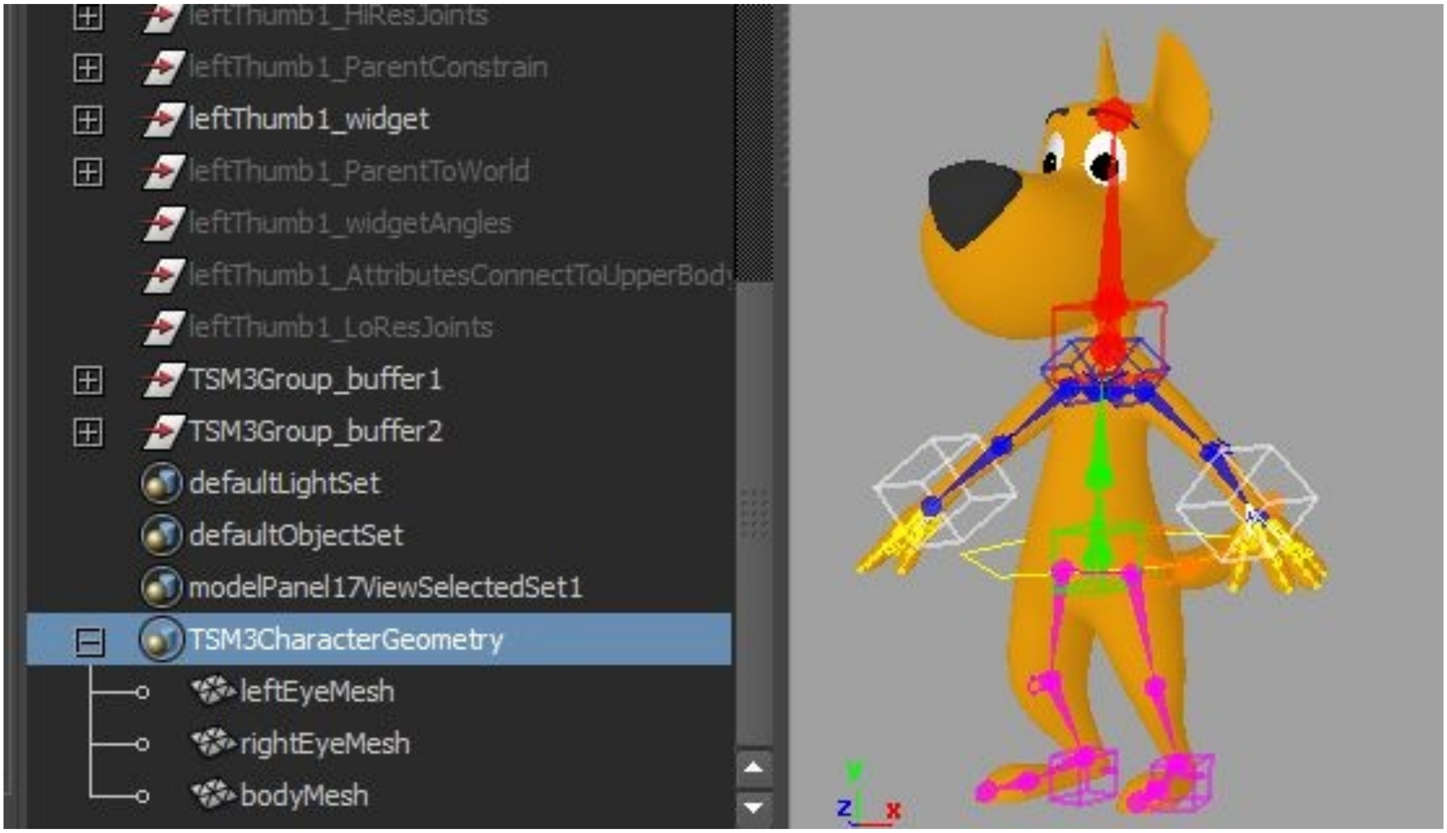
You'll notice that there's a white cube positioned near each of the hands. This is a group, specifically for the fingers. Groups are a convenient way to reposition multiple widgets.

The Groups submenu in the TSM3 menu enables you to create groups, add parts to a group (first select the parts, then the group), remove all parts from a group, and delete groups. If you add mirrored widgets to a group, a mirrored group will be created for the widgets on the opposite side.

Note: The widgets you choose to add to a group must either be unattached, or attached to the same widget.

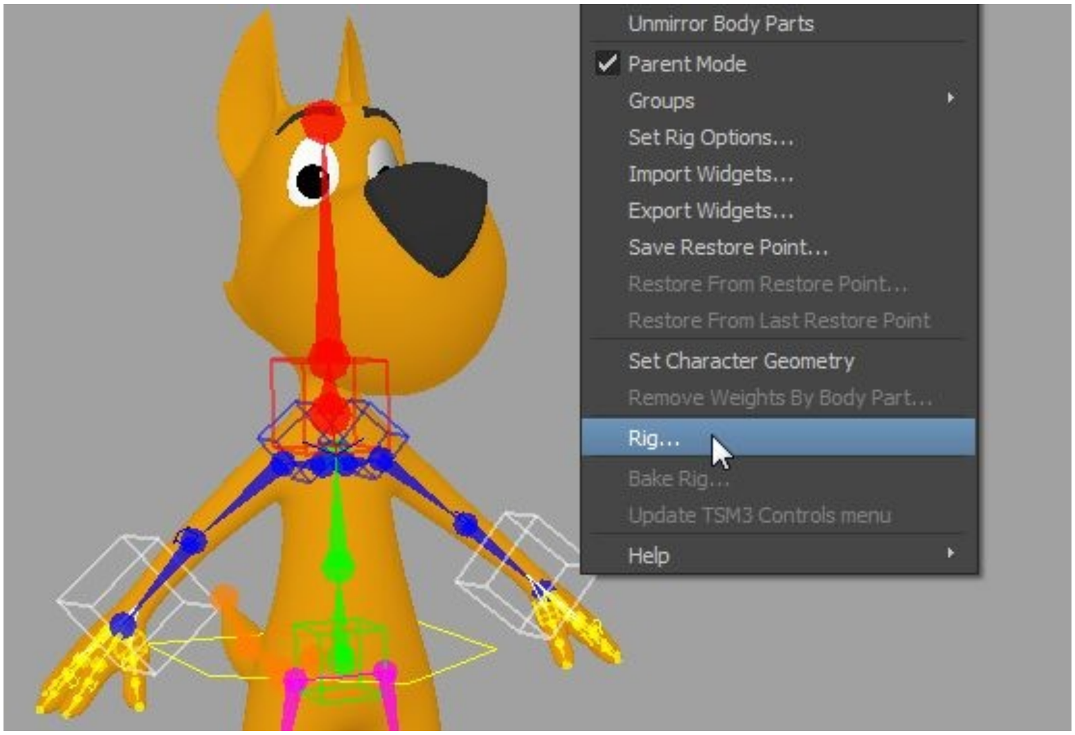
STEP 2: SET GEOMETRY

Once you've got your widgets placed, you'll need to tell TSM3 what geometry it should rig. Simply select your character's geometry and select TSM3 > Set Character Geometry:

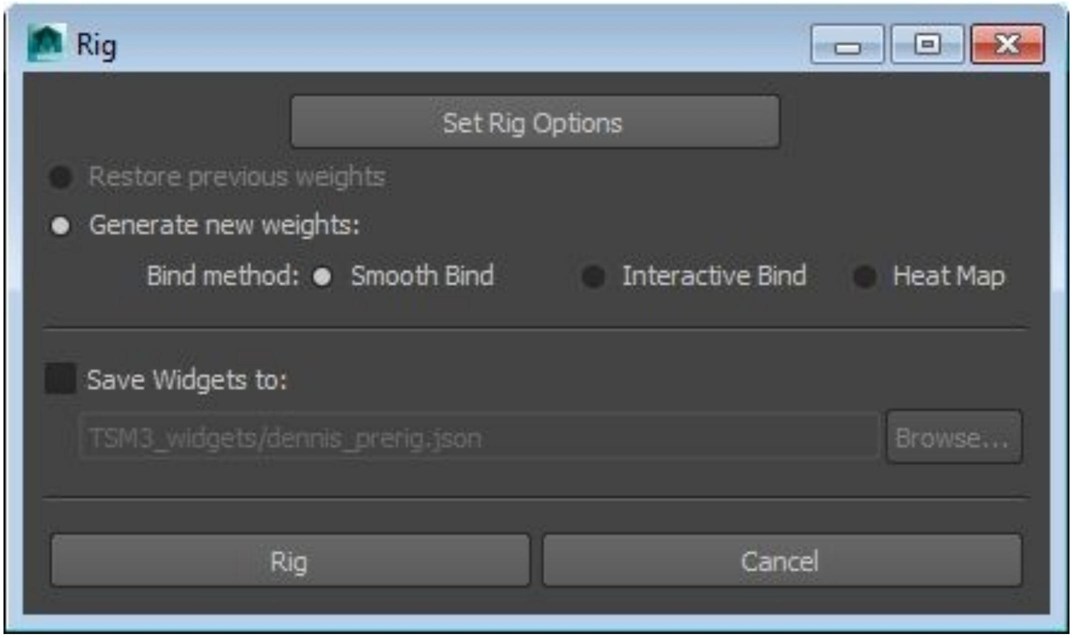


This will create a standard Maya set called **TSM3CharacterGeometry**, from which you can add or remove objects at will. Anything in the set will be weighted to the TSM3 skeleton when rigged.

STEP 3: RIG



Select TSM3 > Rig...



Rig

Set Rig Options

○ Restore previous weights

● Generate new weights:

Bind method: ● Smooth Bind ○ Interactive Bind ○ Heat Map

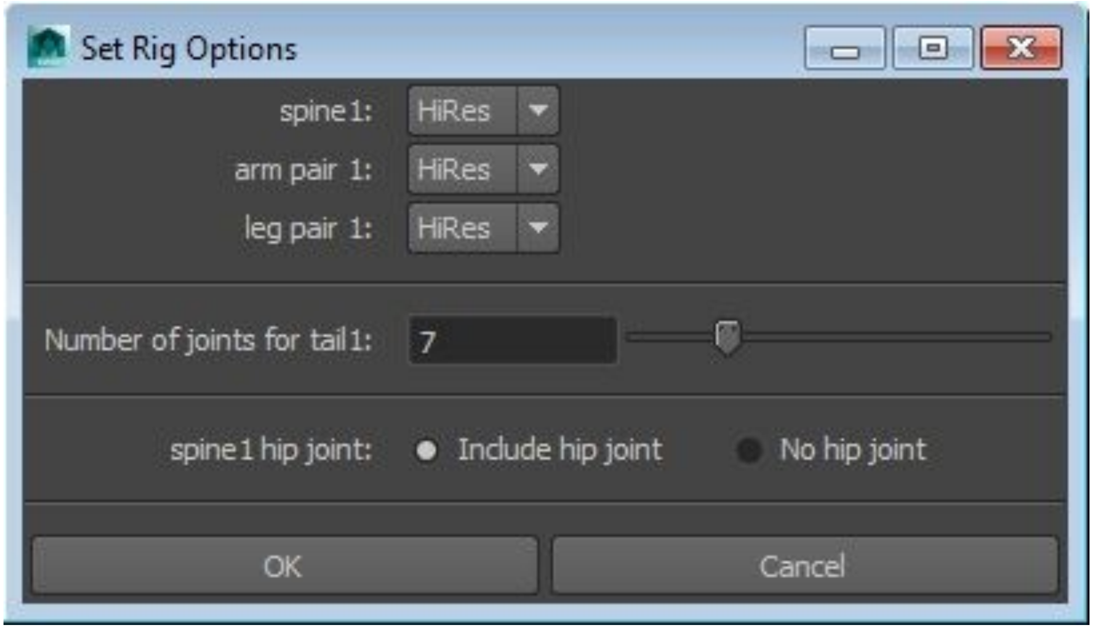
☐ Save Widgets to:

TSM3_widgets/dennis_prerig.json Browse...

Rig Cancel

This pops up a dialog with some options.

The **Set Rig Options** button will allow you to set the resolution of various systems, and also how the spine hip joint is handled:



Set Rig Options

spine1: HiRes

arm pair 1: HiRes

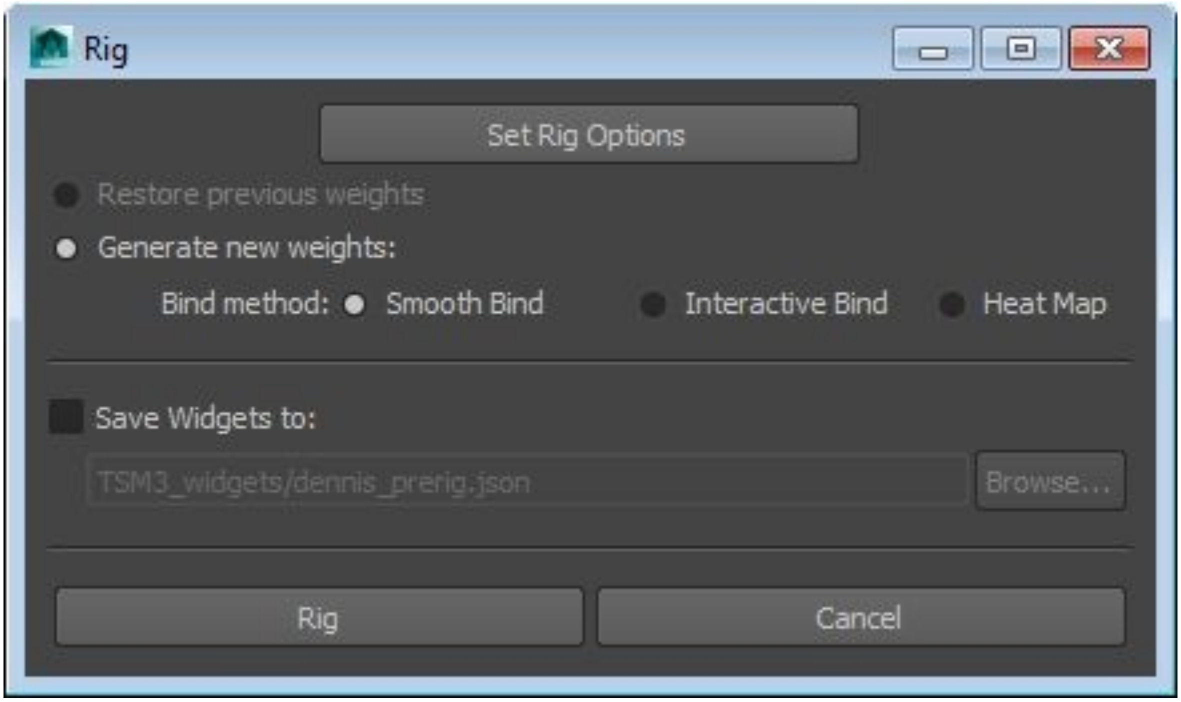
leg pair 1: HiRes

Number of joints for tail1: 7

spine1 hip joint: ● Include hip joint ○ No hip joint

By default, all systems will be set to HiRres. For the spine, this means five joints, while the legs and arms have six each. At the LoRes setting, the spine has three joints, the legs and arms have four. For now, we'll leave them set to HiRes. We'll also leave the tail at the default number of joints.

The spine hip joint, which adds a joint between the base of the spine and the legs, is included by default. However, if you your engine doesn't support this configuration, you will need to select "No hip joint" in order to have the required hierarchy.



○ Restore previous weights

● Generate new weights:

Bind method: ● Smooth Bind ○ Interactive Bind ○ Heat Map

□ Save Widgets to:

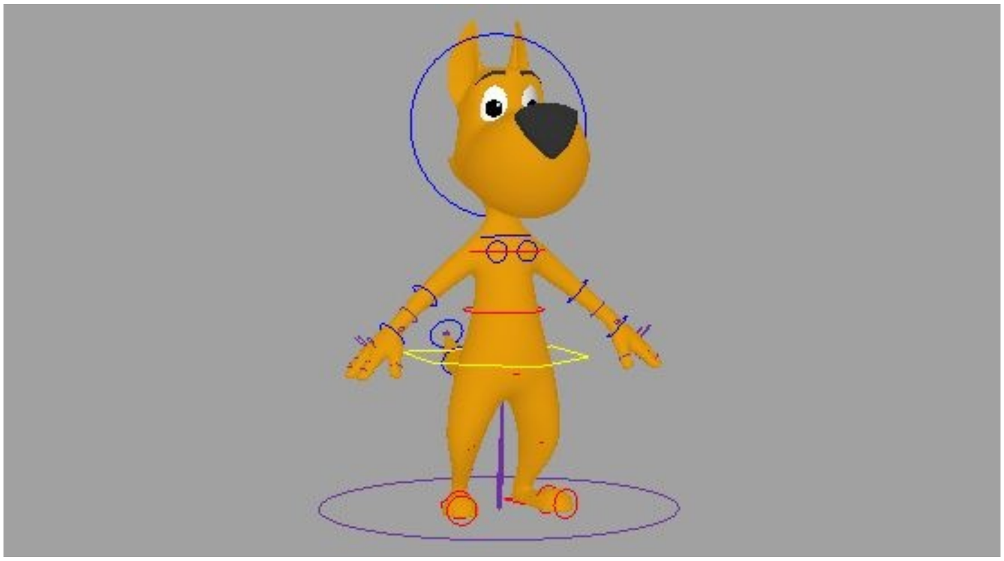
TSM3_widgets/dennis_prerig.json

Once you've set your rig options, the next choice is whether you want to generate new weights, using your choice of weighting method, or whether you want to restore previous weights. Restoring previous weights will come into play if you end up rolling back to the widget stage via the Restore from Restore Point functionality.

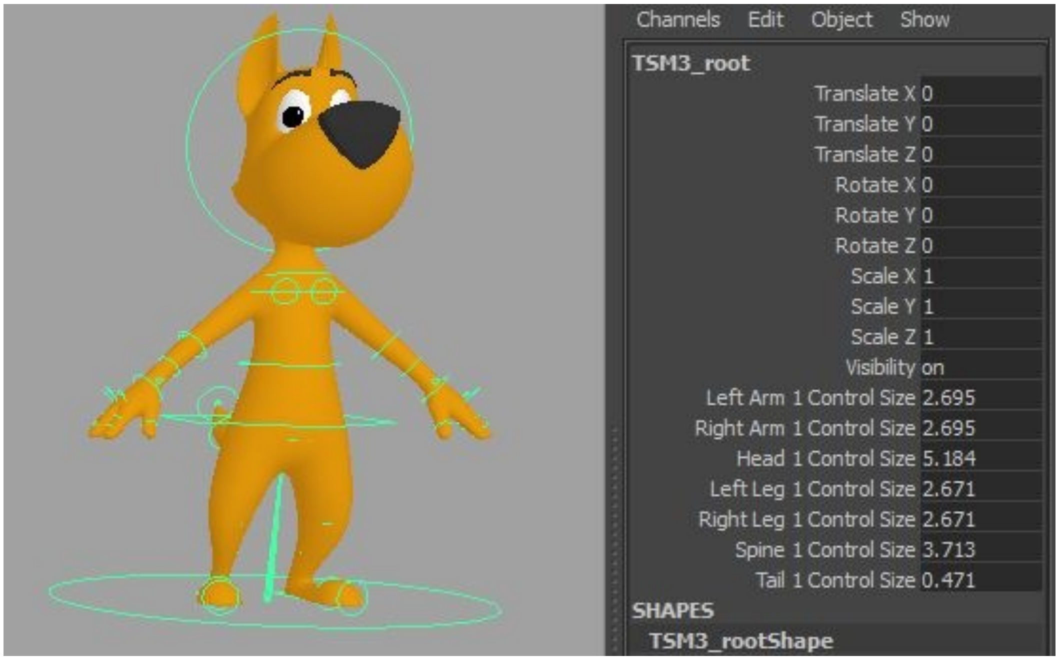
Note: The Heat Map and Geodesic Voxel bind methods only work correctly with geometry that contains influences within it. They will not yield good results on something like Dennis's eye geometry, which are single sided pieces.

Another option is to have TSM3 automatically save your widgets to a .json file. You can then use the Import Widgets feature to bring them into another scene which is particularly useful if you’re rigging multiple similar characters. Finally, click Rig to start the rigging process!

And here are the results: A fully rigged character! You'll find that TSM3 has automatically weighted it for you, although you'll probably need to tweak the weights a bit, using Maya's standard weight painting tools.



Another thing you may need to do is adjust control sizes. In Dennis' case, some of the controls are too small, causing them to disappear into his body. To correct this, select the TSM3_root control – the circle at the base of the character and go to channel control:



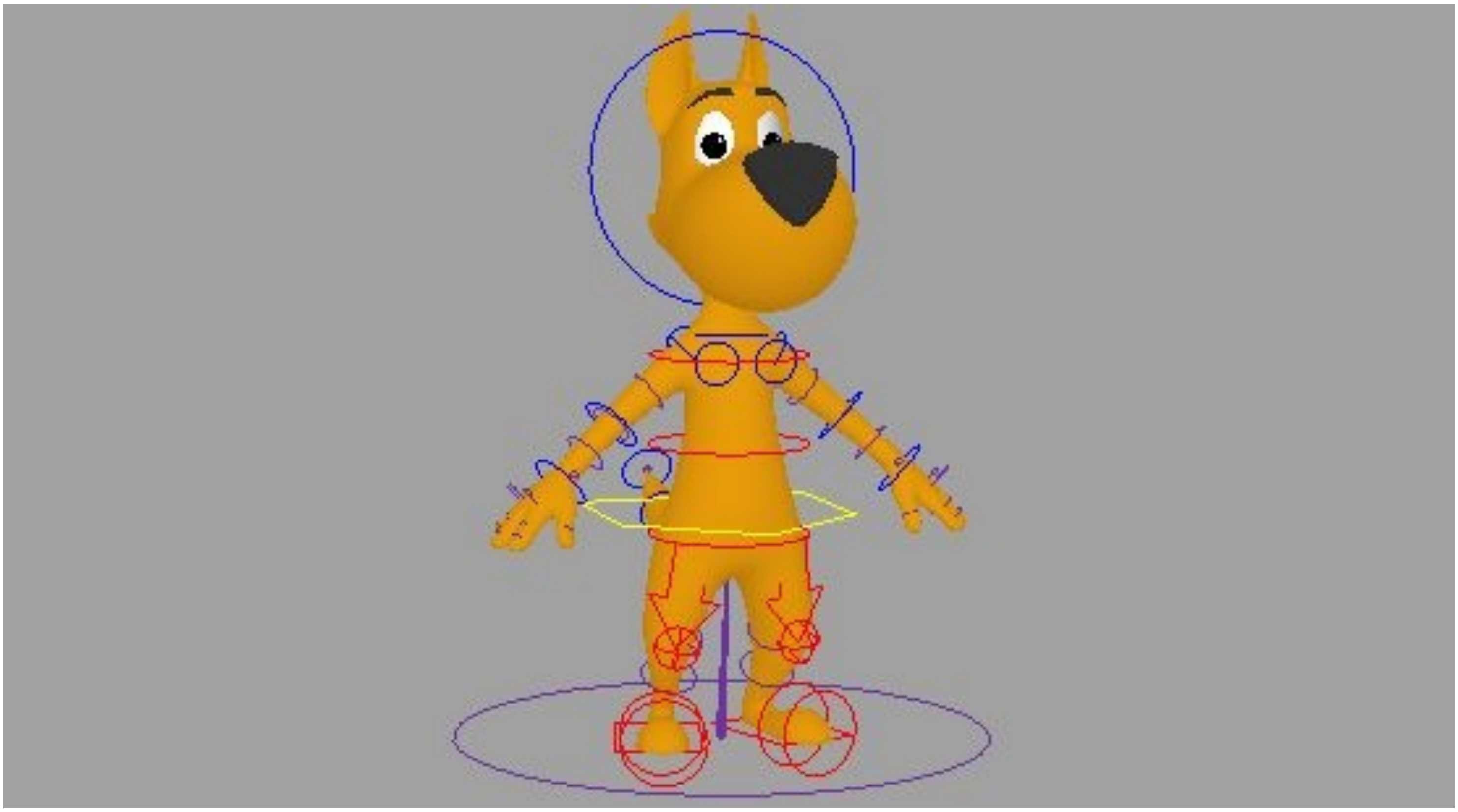
TSM3_root

Translate X 0
Translate Y 0
Translate Z 0
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 1
Scale Y 1
Scale Z 1
Visibility on
Left Arm 1 Control Size 2.695
Right Arm 1 Control Size 2.695
Head 1 Control Size 5.184
Left Leg 1 Control Size 2.671
Right Leg 1 Control Size 2.671
Spine 1 Control Size 3.713
Tail 1 Control Size 0.471

SHAPES

TSM3_rootShape

From here you can adjust the control sizes to fit your character better:

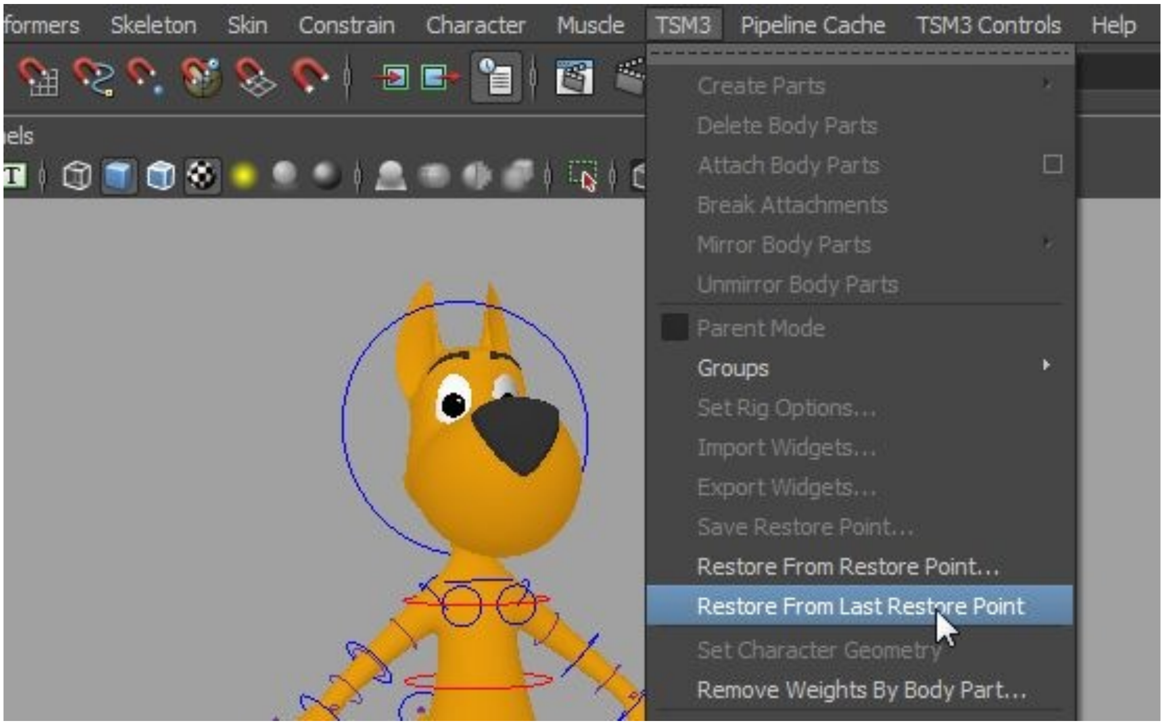


You may also want to adjust the sizes of individual controls, or adjust their appearance in more specific ways. Since the controls are NURBs curves, you can use Maya's existing curve editing tools to edit them.

Note: Any changes made to these controls will be lost if the rig is reverted to a restore point.

STEP 4: USING RESTORE

As you're working on the weights and trying out various poses with the rig, you may decide that one or more of your widgets wasn't positioned exactly right. Fortunately, TSM3 makes it easy to deal with this. Simply go to TSM3 > Restore From Last Restore Point:



This will revert the rig to its widget form. You can then reposition the widgets and re-rig.

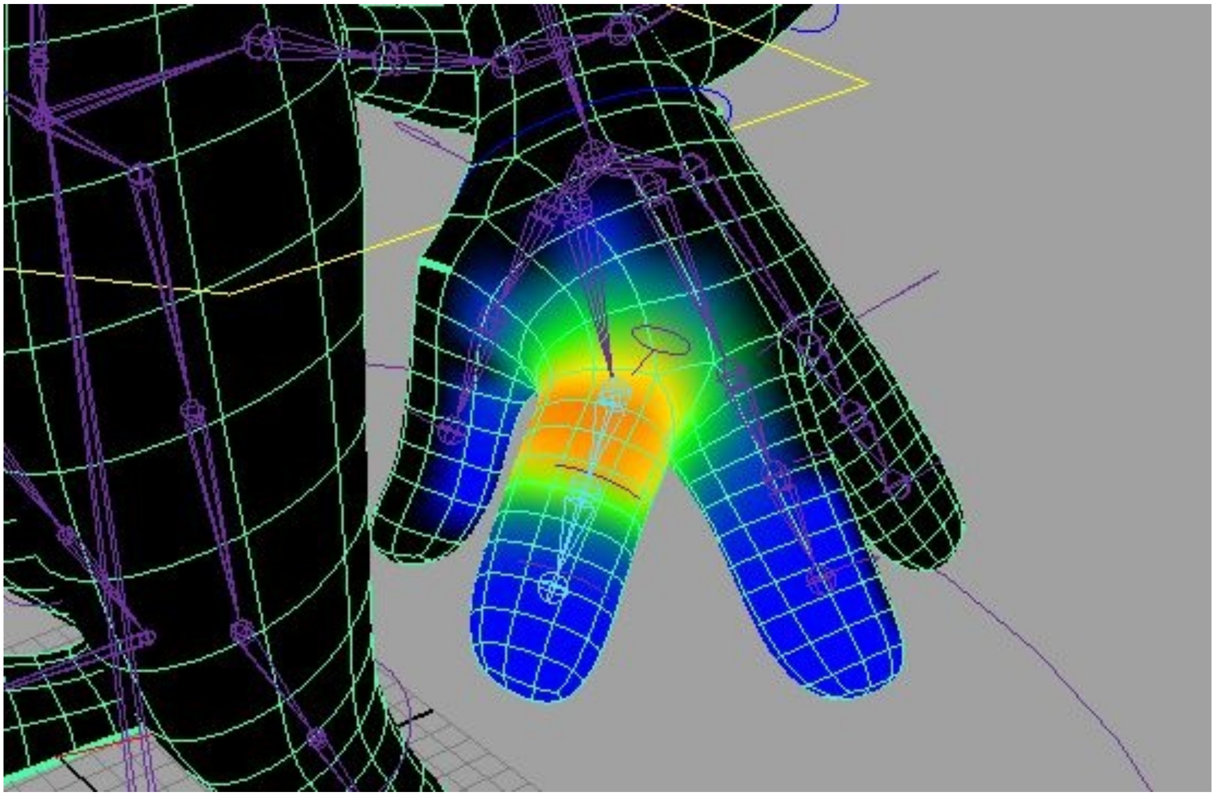
'Restore from Last Restore Point' uses information that TSM3 embeds in your file, while 'Restore from Restore Point' allows you to restore from a .json file. To save a .json file, effectively taking a 'snapshot' of the current state of your widgets, you can use 'Save Restore Point' before you rig.

When you re-rig the character, you'll have the option to restore weights. This option only affects the geometry that was weighted through TSM3 originally, using 'Set Character Geometry.' If you skinned additional geometry after weighting, that will be lost. However, you can save a version of your rigged file and use Maya's weight transfer tools to restore the weights manually.

Note: restoring from a restore point will remove any animation on the rig. To be able to animate a character but still be able to use this functionality, we recommend referencing the character file into a scene, and doing your animation in that scene. If you need to go back and restore and re-rig in the character file, that won't affect the animation file.

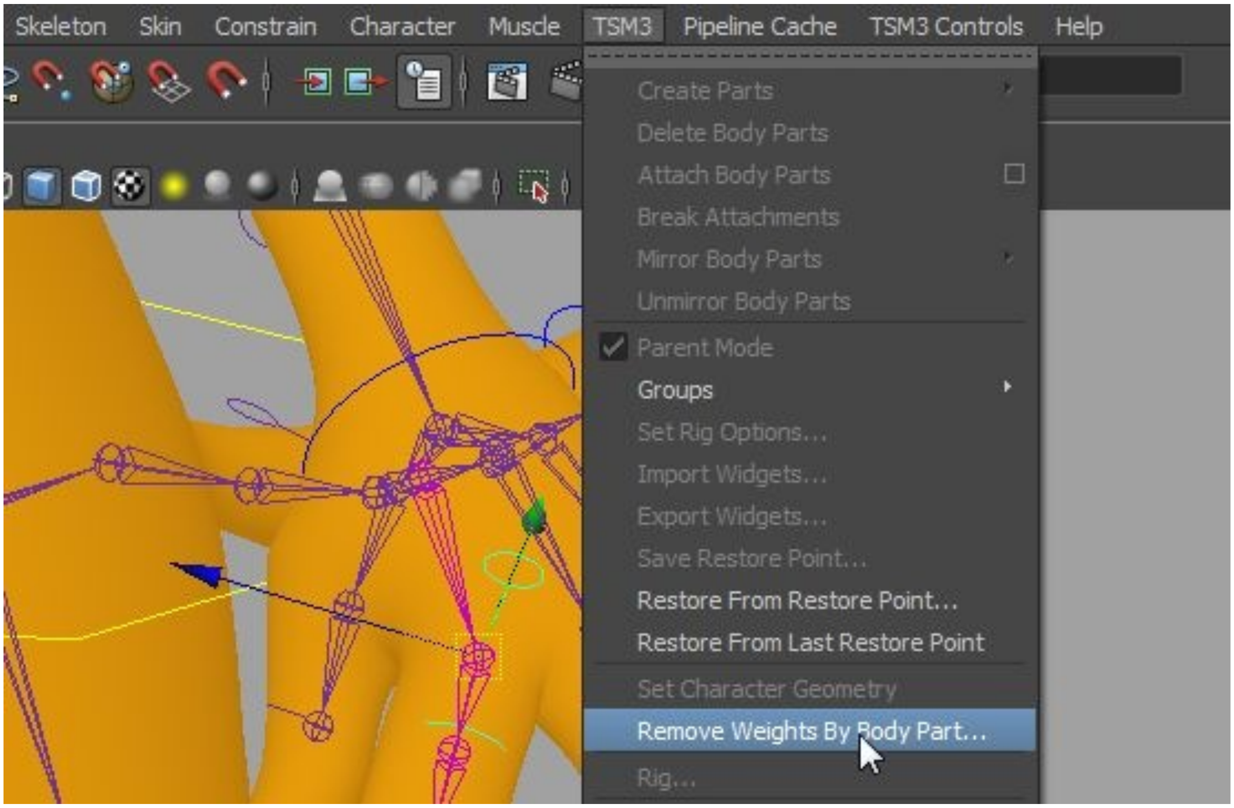
STEP 5: ADJUSTING WEIGHTS

For the most part, you'll use Maya's standard weight painting tools to adjust the weights on your character. However, there's an additional feature in TSM3 that may expedite the process in some areas.

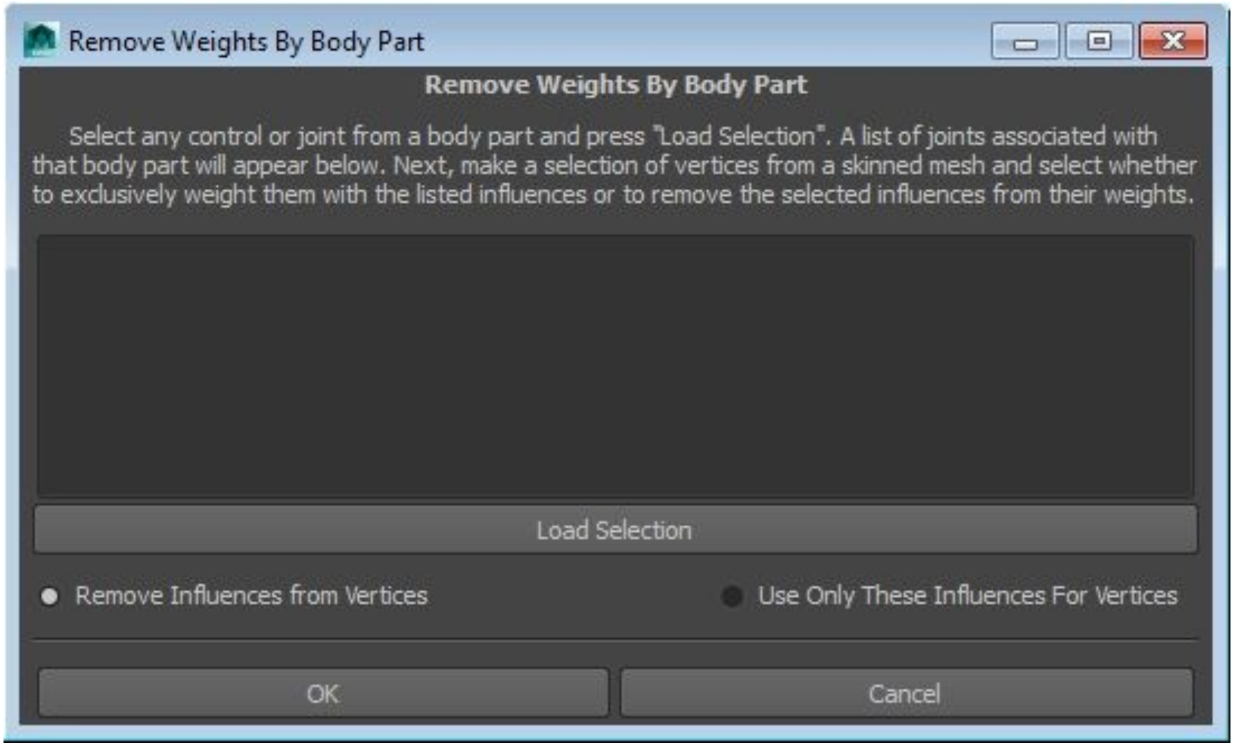


Here you can see that the first finger has some weighting on the second finger. TSM3's 'Remove Weights By Body Part' can help us deal with this.

Select 'Remove Weights By Body Part' from the TSM3 menu:



This will pop up a new window:

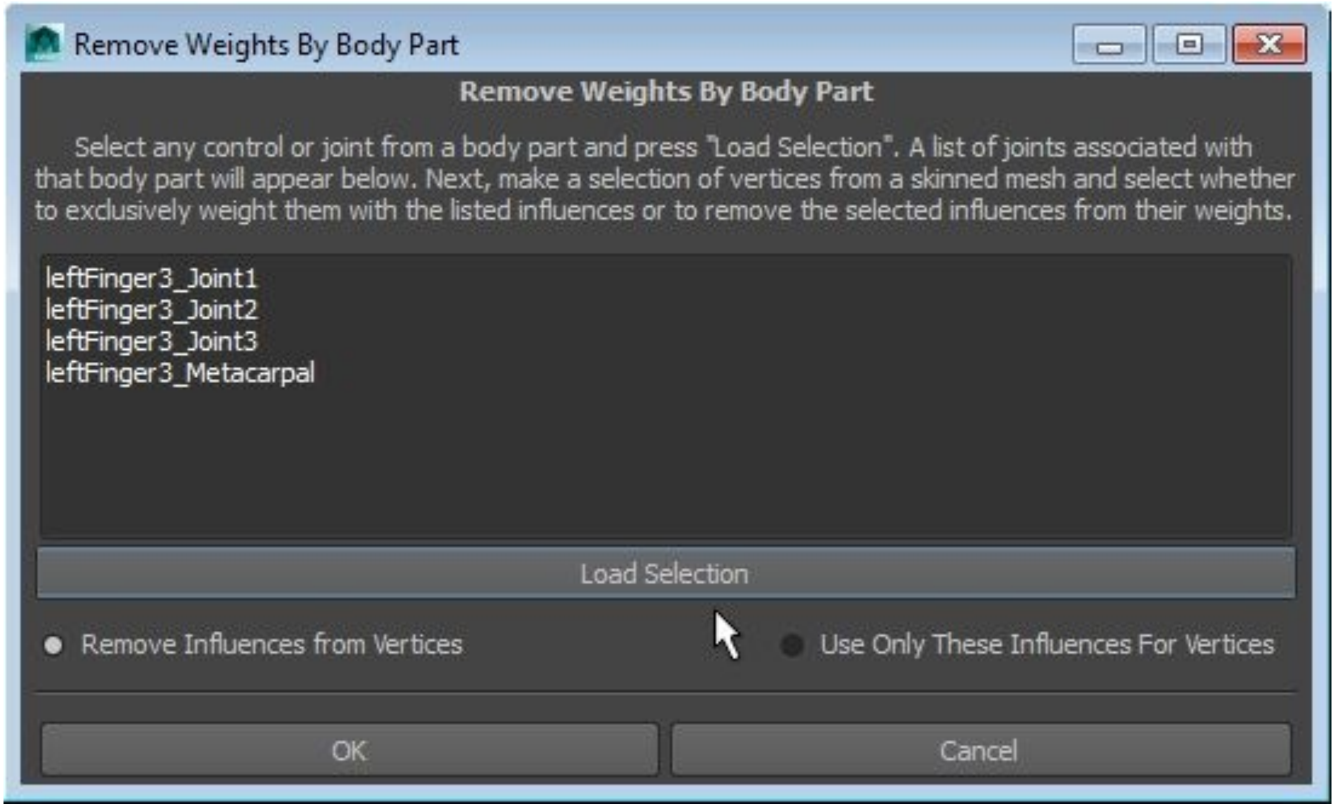

Remove Weights By Body Part

Remove Weights By Body Part

Select any control or joint from a body part and press "Load Selection". A list of joints associated with that body part will appear below. Next, make a selection of vertices from a skinned mesh and select whether to exclusively weight them with the listed influences or to remove the selected influences from their weights.

Load Selection
● Remove Influences from Vertices
○ Use Only These Influences For Vertices
OK
Cancel

Next, select a control from the first finger, and click 'Load Selection.' You'll see that all of the joints belonging to that finger are added to a list:



Remove Weights By Body Part

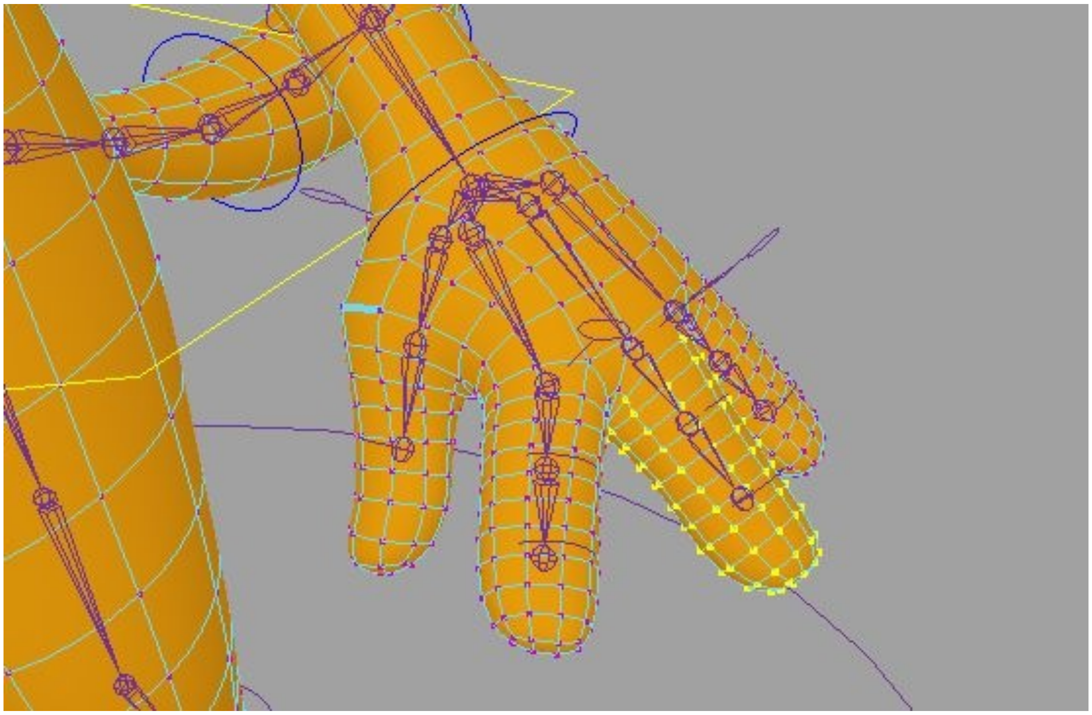
Remove Weights By Body Part

Select any control or joint from a body part and press "Load Selection". A list of joints associated with that body part will appear below. Next, make a selection of vertices from a skinned mesh and select whether to exclusively weight them with the listed influences or to remove the selected influences from their weights.

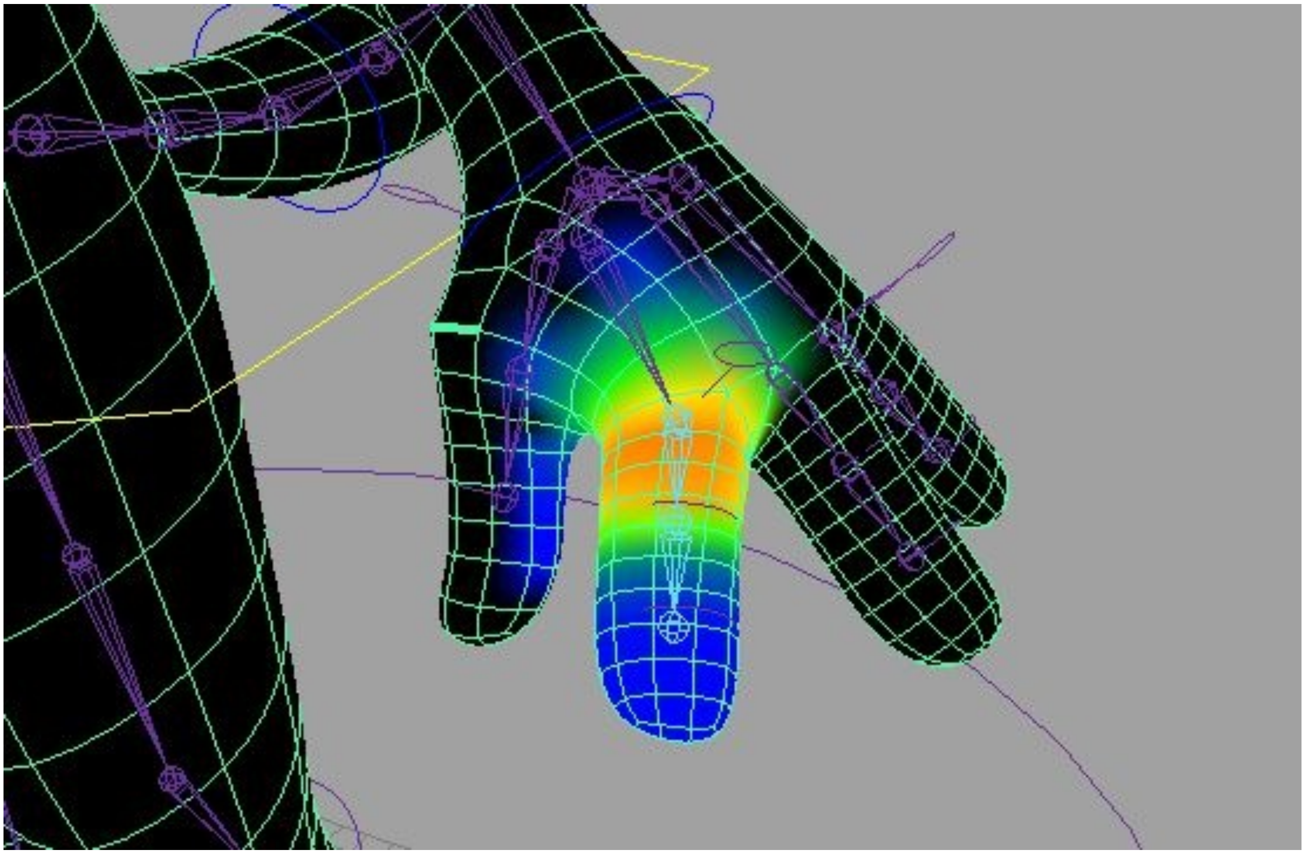
leftFinger3_Joint1
leftFinger3_Joint2
leftFinger3_Joint3
leftFinger3_Metacarpal

● Remove Influences from Vertices
○ Use Only These Influences For Vertices

Next, select the points of the second finger:



Finally, with the radio button still set to 'Remove Influences from Vertices,' click OK in the Remove Weights By Body Part window.



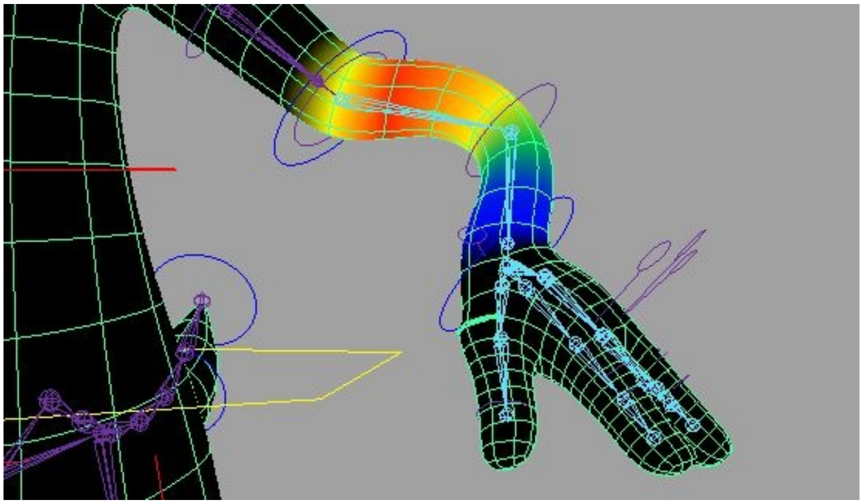
And here you can see that the weights of the first finger have been completely removed from the second.

As an alternate approach, you can select the system that you want to weight exclusively to, then the points that you want to weight, and then select the 'Use Only These Influences for Vertices' option at the bottom of the Remove Weights By Body Part window. That will weight the points exclusively to the selected influences.

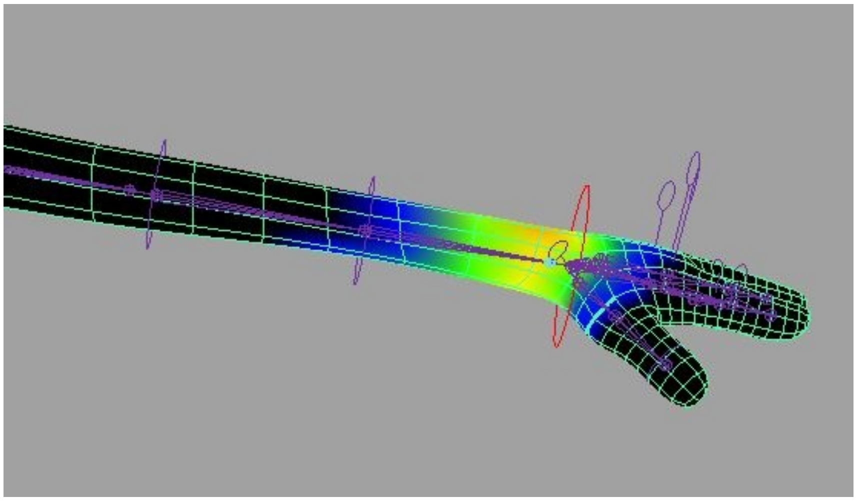
Note: removing weights will fail if it would leave any vertices with zero weights.

WEIGHTING FOR BENDABLE, STRETCHABLE LIMBS

To allow limbs to be bent in between the primary joints, TSM3 uses some additional joints. You'll want to make sure to weight smoothly between them – even if you don't plan to use them to bend, these joints also distribute twist. For best results, you'll also want to use dual quaternion weighting in these areas.

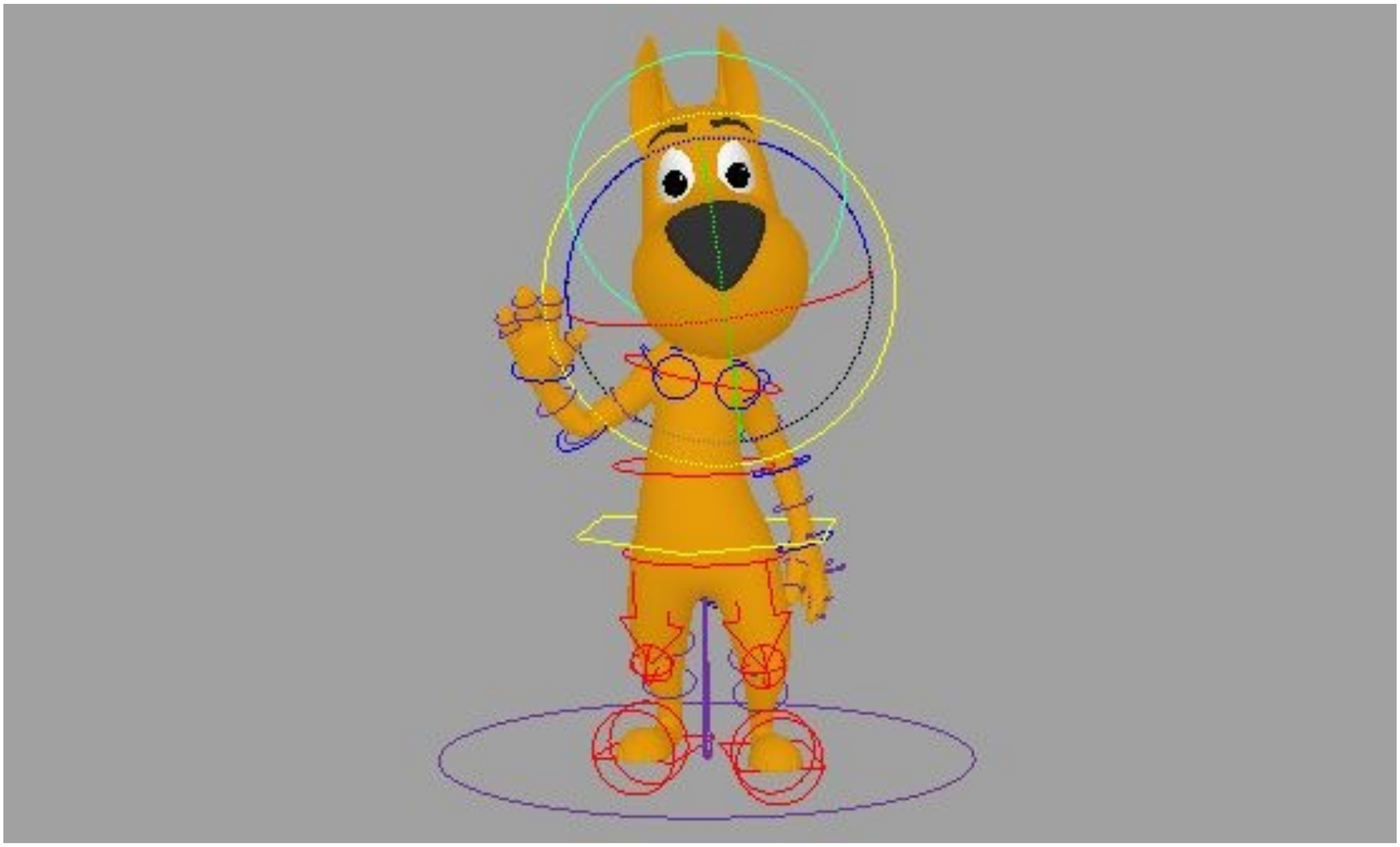


When limbs are stretched, the area near the wrist/ankle can become distorted. To compensate for this, TSM3 has additional joints close to the wrists and ankles. By weighting to these joints as well, you'll be able to stretch your limbs without any undesirable distortion:



USING THE RIG

TSM3 is designed to be intuitive and easy to use. For the most part, you simply grab the controls the brightly colored circles and other shapes – and move them around:



However, there are some extra features that are worth mentioning:

TRUE MIRRORING

For symmetrical poses, you can simply copy keys or values from one side to the other—there is no need for specialized scripts to invert or otherwise change the values being copied. It is common for this to work for rotation values in rigs—for TSM3 rigs, it works for translation values as well.

Note: Prior to 2016, there is a bug in Maya that will lead to manipulators on the mirrored side operating in reverse when using local and object space modes. However, you may find that the convenience of being able to quickly mirror animation is worth the minor inconvenience of dealing with this glitch.

IK POLE VECTORS

For IK systems, both rotational and translation pole vectors are available. The rotational pole vector is visualized as an arrow, with the translation pole vector is a small wireframe sphere, positioned near the point of the arrow.

IK/FK SWITCHING

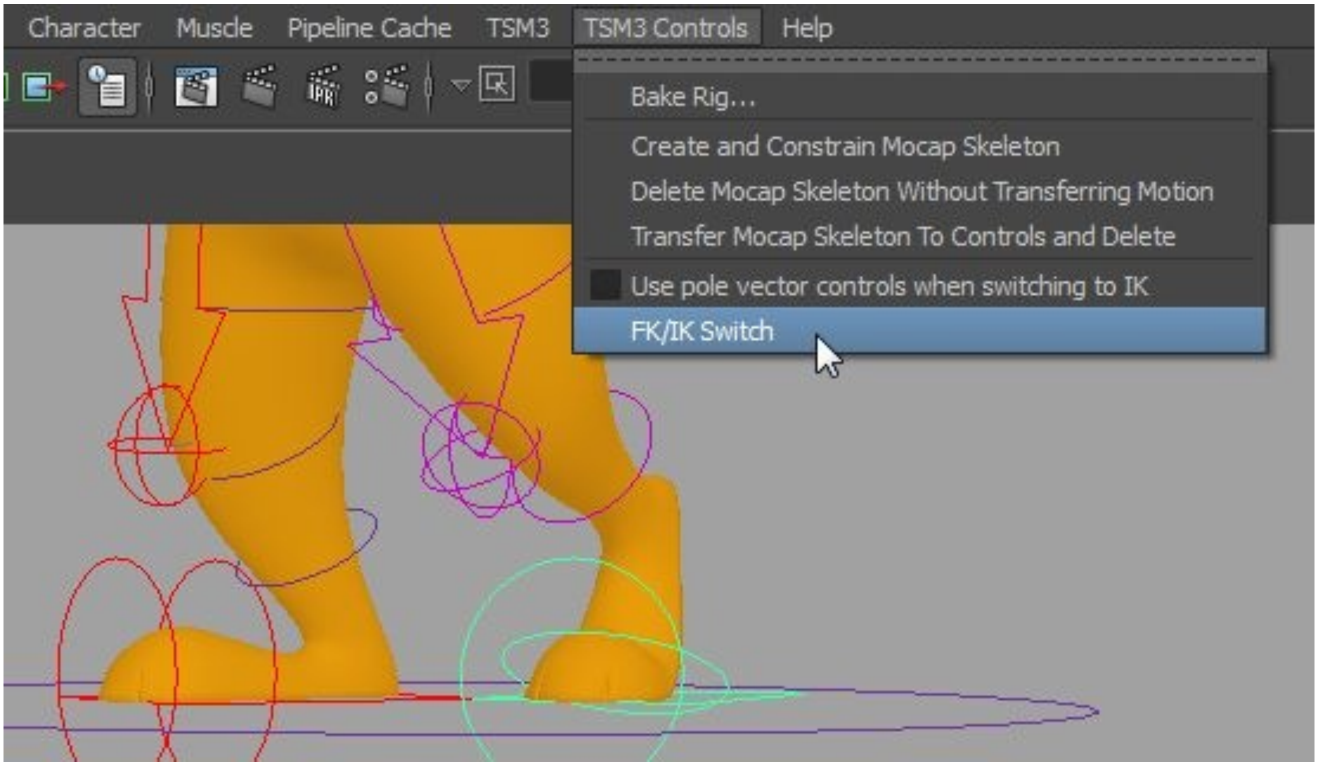
Several of the systems in TSM3 can be switched between IK and FK controls. For those unfamiliar with these concepts, IK, or inverse kinematics, allow you to take the last control in a chain – for example, the hand – and position it freely, using both translation and rotation. The joints in the chain will automatically adjust themselves to allow this positioning.

FK, or forward kinematics, allow you to take the first control in a chain – for example, the shoulder – and rotate it, which will reposition all of the subsequent controls in the chain. Other controls in the chain can be rotated as well.

To do a basic switch between IK and FK, select the house-shaped upper body control. You will see an FKIKSwitch attribute for each switchable system. Zero represents FK, and one represents IK. It's also possible to have fractional values, if you need to do a transition over time.

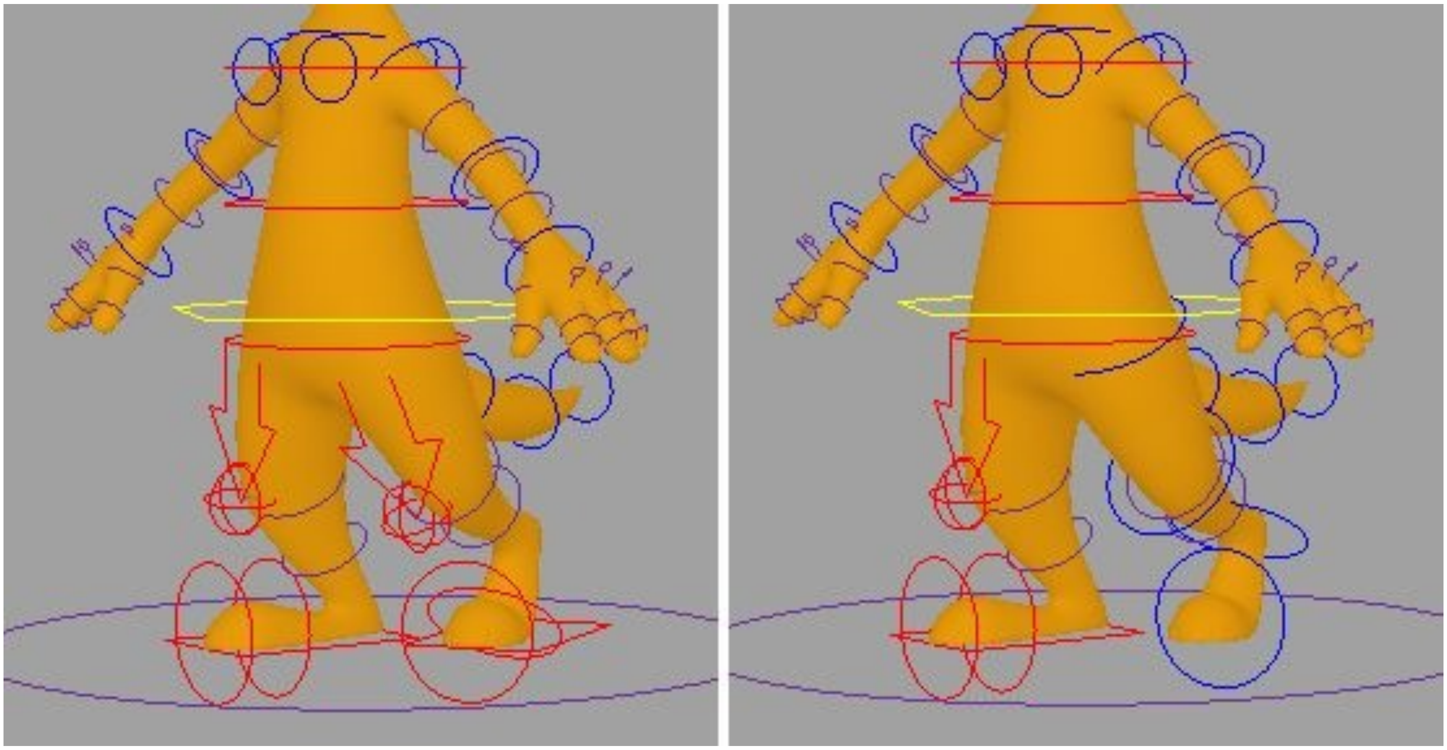
In most cases, though, you'll want to instantly switch between IK and FK, while still maintaining a pose. To do this, simply select any control in the system you want to switch, and select TSM3 Controls > IK/FK Switch.

The TSM3 Controls menu will be present in any scene that has a TSM3 rigged character in it, regardless of whether the TSM3 plugin is installed.


Character
Muscle
Pipeline Cache
TSM3
TSM3 Controls
Help
Bake Rig...
Create and Constrain Mocap Skeleton
Delete Mocap Skeleton Without Transferring Motion
Transfer Mocap Skeleton To Controls and Delete
Use pole vector controls when switching to IK
FK/IK Switch

This will switch the controls to the opposite of their current control type: IK switches to FK, or FK switches to IK.

And the controls will be appropriately placed to recreate the original pose:

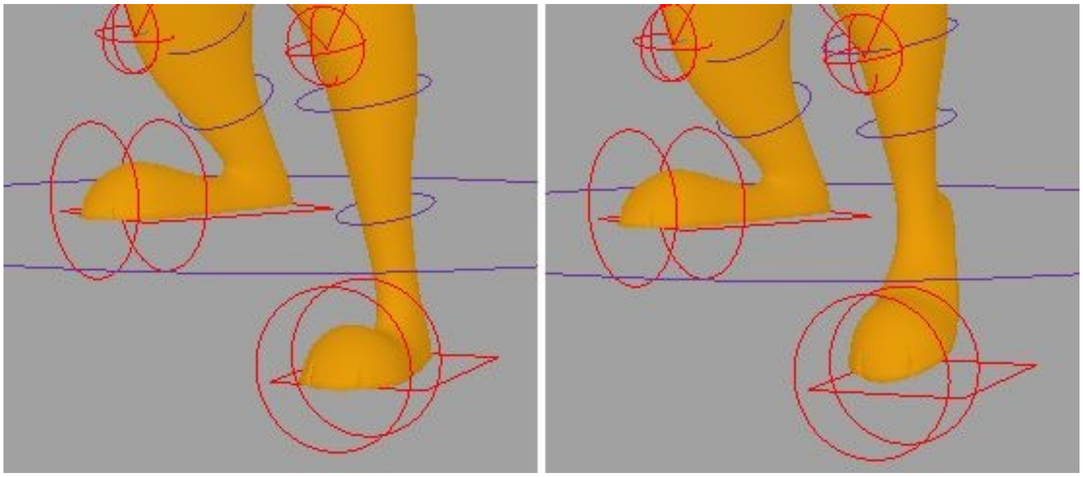


Note: You can choose which of the two pole vector types are used when switching from IK to FK. By default, the rotational pole vector is rotated to match the pose. However, you can check "Use translation pole vector when switching to IK" to have the translation pole vector positioned to match instead.

STRETCH AND AUTO STRETCH

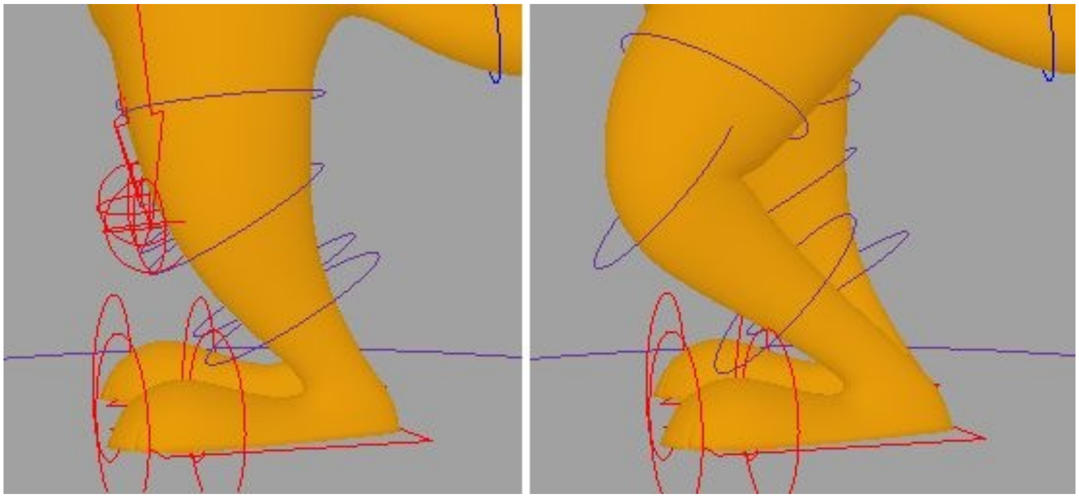
You will find stretch attributes on the FK upper and lower arms and legs, as well as the FK foot. All stretch attributes default to one. Values less than one shrink the section being scaled, while values greater than one stretch it.

You will also find stretch and auto stretch attributes on the IK hand and foot controls. Auto stretch defaults to one, which means that the IK limbs will automatically stretch to reach their controls. Setting it to zero limits the extent of their reach to the natural length of the limb.



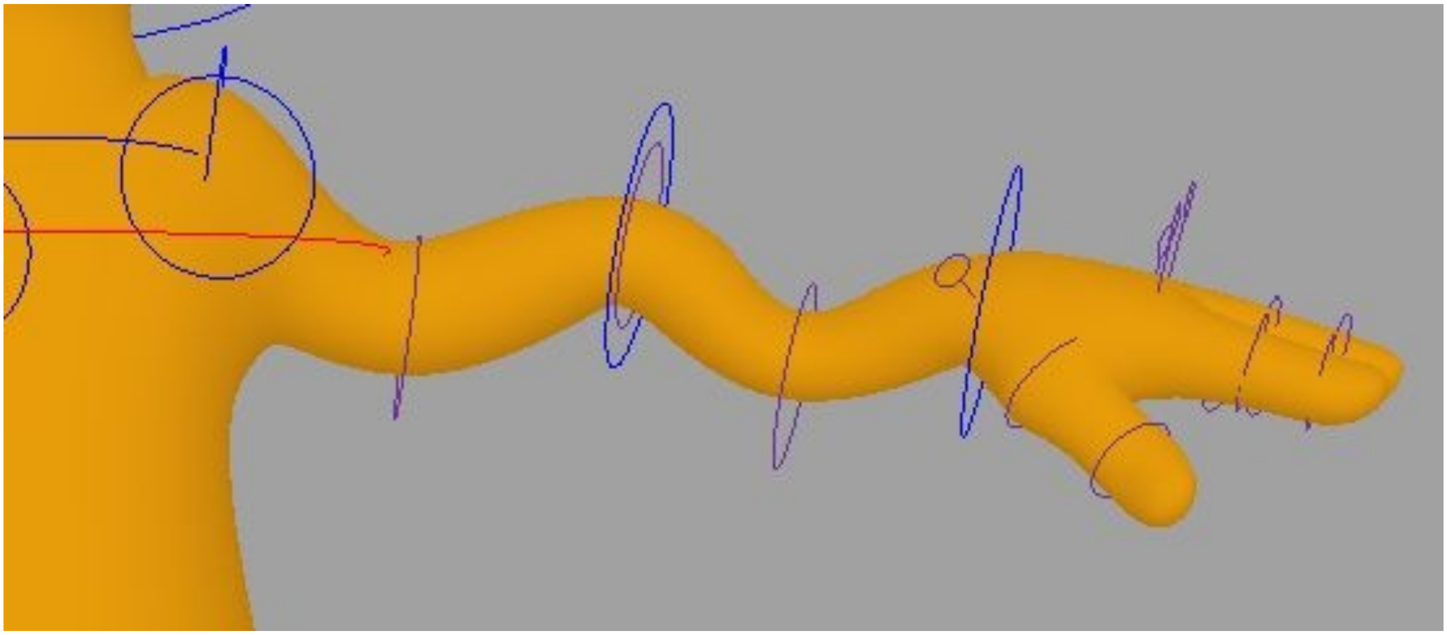
Left, auto stretch = 1. right, auto stretch = 0.

The stretch attributes for the upper and lower limbs, on the other hand, shrinks or stretches that portion of the limb without moving the relevant hand or foot control. You'll find these attributes as individual “stretch” attributes on the individual FK controls, or as “upper leg/arm” and “lower leg/arm” on the IK foot and hand, respectively.



Left, lower leg stretch = 1. Right, lower leg stretch = 1.5.

BENDY LIMBS



When you rig arms and legs in HiRes mode, in addition to the standard controls at the joints, you'll get additional, smaller controls that can be used to bend the limbs in various ways. This functionality is like having a “broken rig” setup on top of the standard rig, and provides great flexibility.

ANTI IK POP

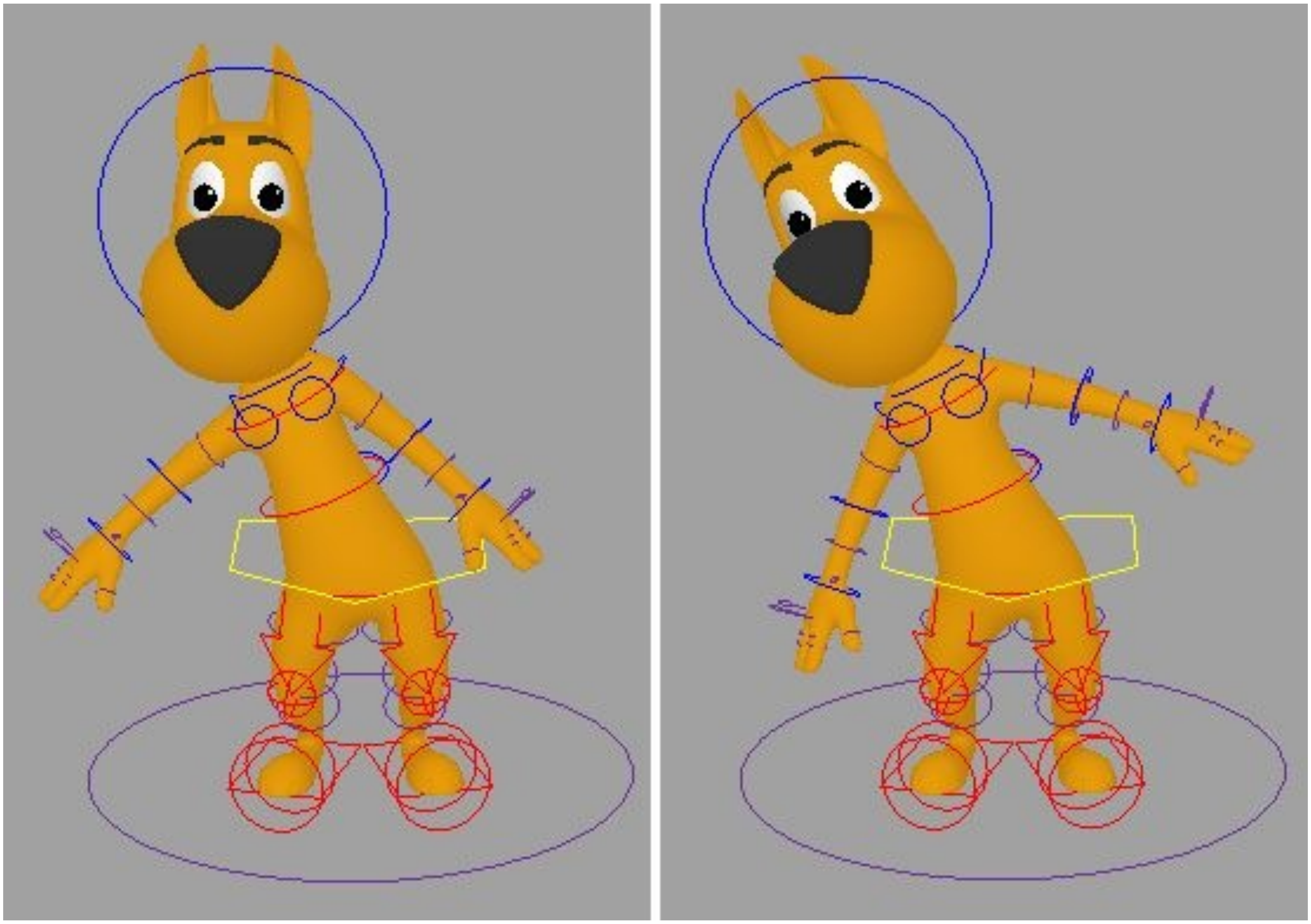
There is an Anti IK Pop attribute on the IK hand and foot controls. It defaults to zero. When set to 1, this eases in a slight stretch at the point when the IK limb is almost straight, smoothing out the traditional 'pop' at the transition from bent to straight.

Note: Turning on Anti IK Pop may change the length of the limb slightly in its default pose, which can be countered by using stretch values to return it to it's original length.

ROTATION ISOLATION

Rotation isolation is a concept that applies to FK systems, including the FK controls for the arms, legs, and tails, as well as the head. Each of these systems has its own Rotation Isolation attribute on the upper body control.

This attribute defaults to one, making the systems isolated. Switching them to zero makes the systems non-isolated:



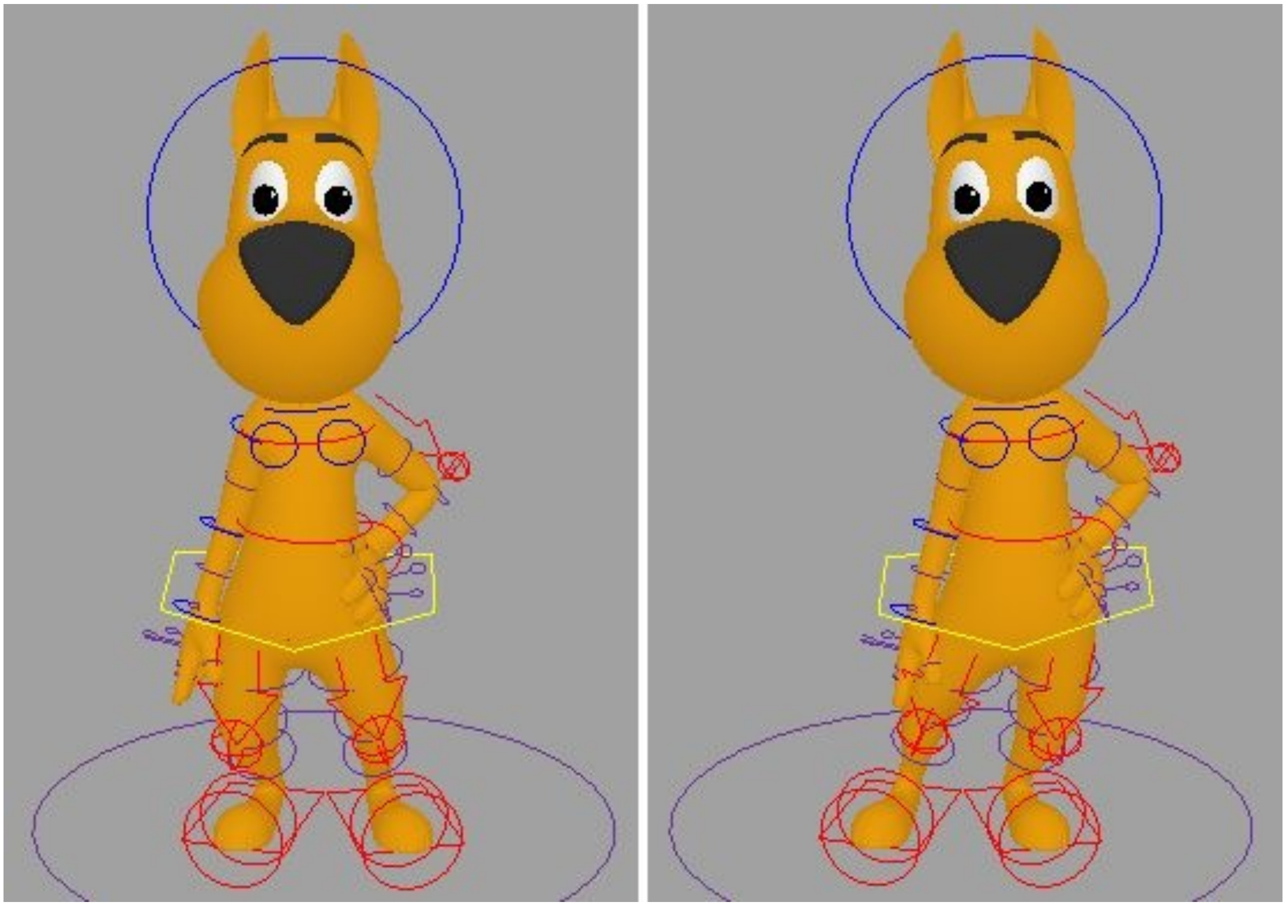
Left: isolated arms and head. Right: non-isolated arms and head.

As you can see above, isolated systems maintain their own rotation, independent of the systems they're parented to. Non-isolated systems rotate with their parents.

SPACE SWITCHING

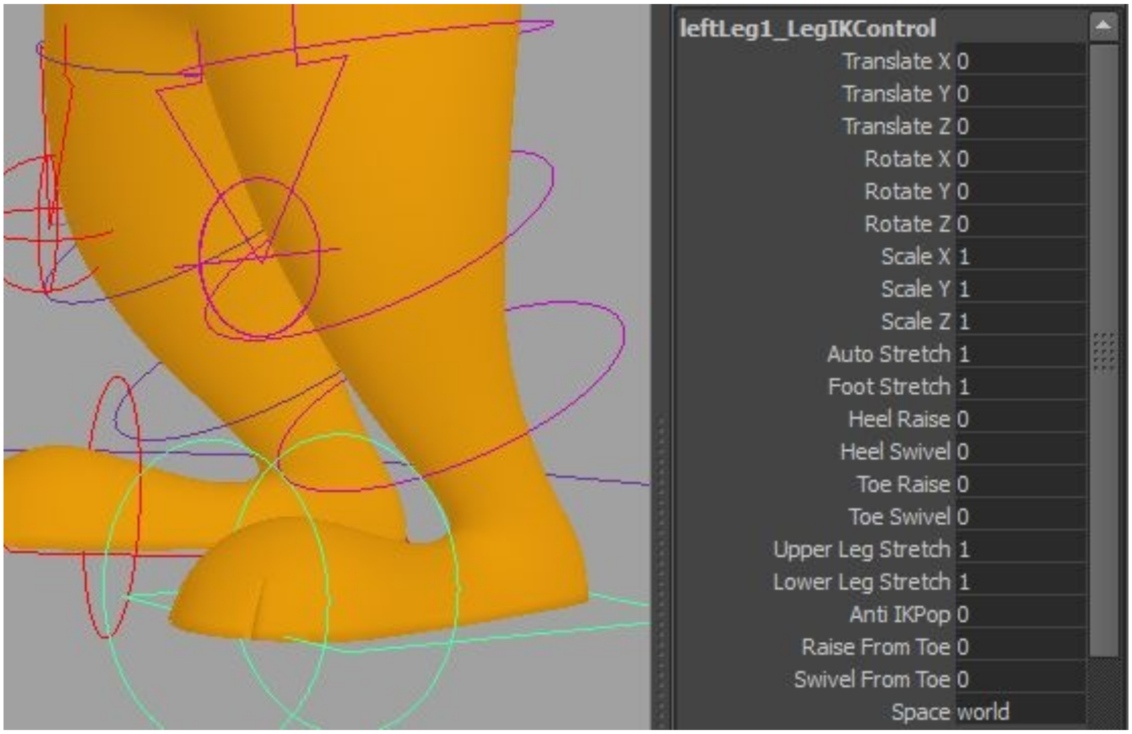
For IK hands and feet, you can choose what space the control operates in, using the 'Space' toggle on each control. By default, space is set to world, which results in a standard IK setup, where moving the body leaves the hands and feet in place.

However, in some circumstances, you may want to use IK controls, but have them move with the body. For example, if our character is resting his hand on his hip, you can switch the space of that hand to "upperBody." Now you can move the upper body, and the hand with stick to the hip:



FOOT CONTROLS

If you select one of the square foot controls and look at the channel box, you'll see a number of extra attributes:



leftLeg1_LegIKControl

Translate X	0
Translate Y	0
Translate Z	0
Rotate X	0
Rotate Y	0
Rotate Z	0
Scale X	1
Scale Y	1
Scale Z	1
Auto Stretch	1
Foot Stretch	1
Heel Raise	0
Heel Swivel	0
Toe Raise	0
Toe Swivel	0
Upper Leg Stretch	1
Lower Leg Stretch	1
Anti IKPop	0
Raise From Toe	0
Swivel From Toe	0
Space	world

These function as follows:

Stretch Features: See the Stretch and Auto Stretch section.

Heel Raise: Defaults to zero. Can be positive or negative. Raises or lowers the heel, respectively.

Heel Swivel: Defaults to zero. Can be positive or negative. Swivels the heel around the ball of the foot, inward or outward, respectively.

Toe Raise: Defaults to zero. Can be positive or negative. Raises or lowers the toe, respectively.

Toe Swivel: Defaults to zero. Can be positive or negative. Swivels the toe around the ball of the foot, outward or inward, respectively.

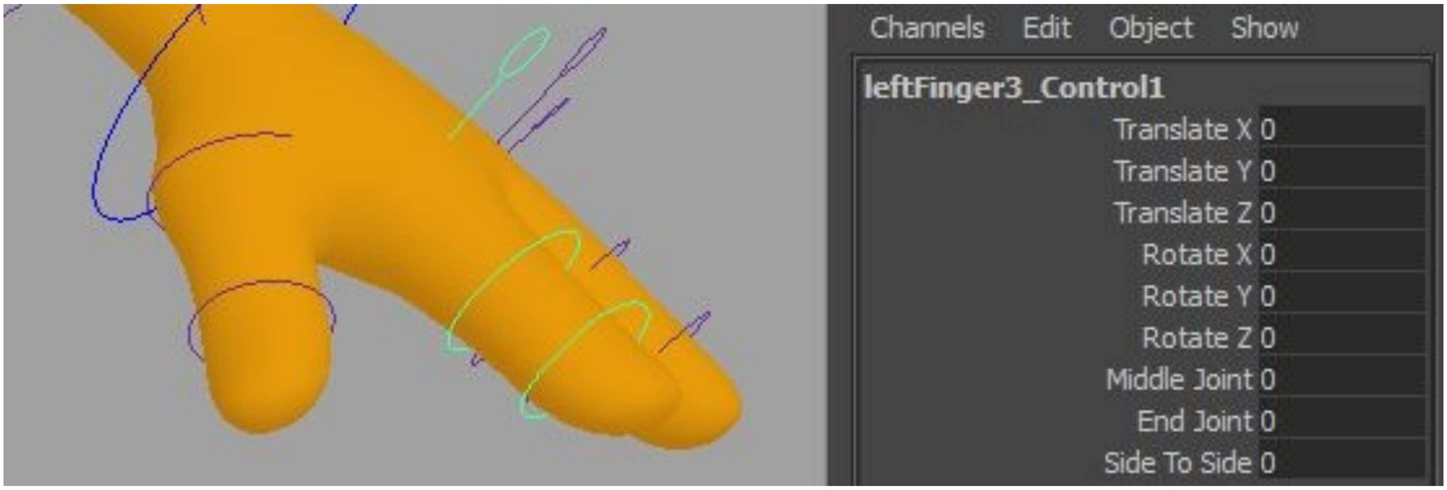
Anti IKPop: See Anti IKPop .

Raise From Toe: Defaults to zero. Can be positive or negative. Raises or lowers the foot using the toe as a pivot point.

Swivel From Toe: Defaults to zero. Can be positive or negative. Swivels the foot using the toe as a pivot point.

Space: See Space Switching. (link to Space Switching section, above.)

FINGER CONTROLS



leftFinger3_Control1

Translate X	0
Translate Y	0
Translate Z	0
Rotate X	0
Rotate Y	0
Rotate Z	0
Middle Joint	0
End Joint	0
Side To Side	0

Each finger control also has a number of additional attributes:

Middle Joint, End Joint: Controls the middle and end joints of the finger, respectively.

Side to Side: Rotates the middle joint of the finger from side to side.

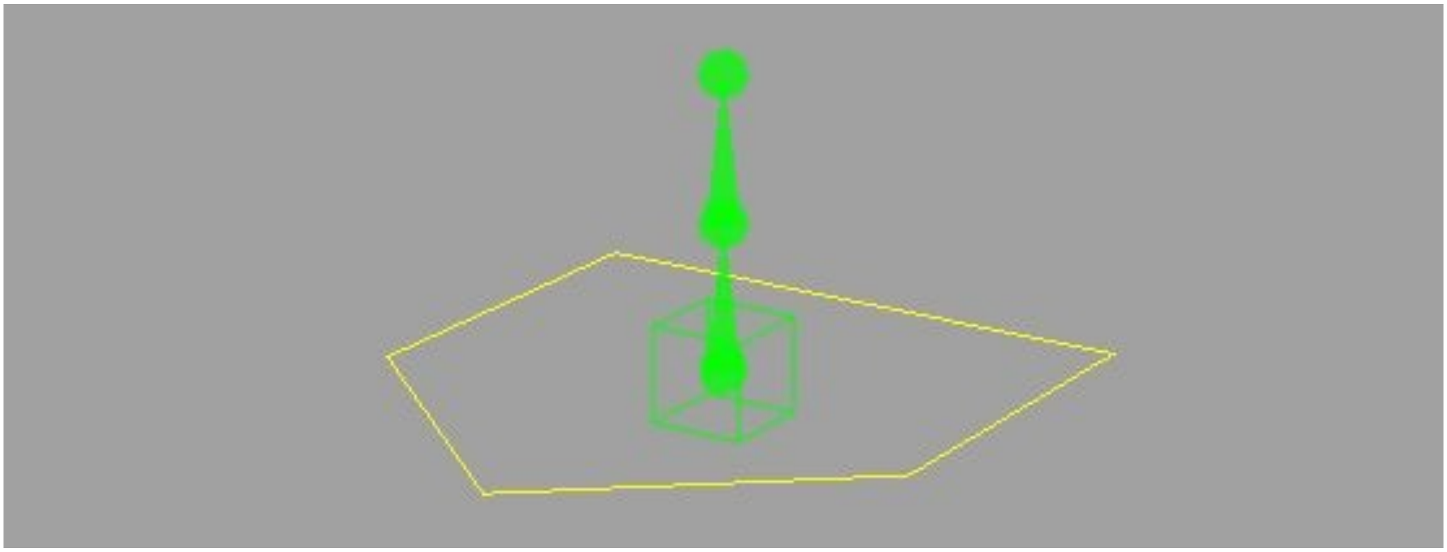
BUILD YOUR OWN

Although we provide a pre-assembled biped and quadruped, with numerous options for each, you may find that you need to add additional parts, or even build your own custom configuration.

To do this, go to TSM3 > Build Parts and select whichever part you want to add.

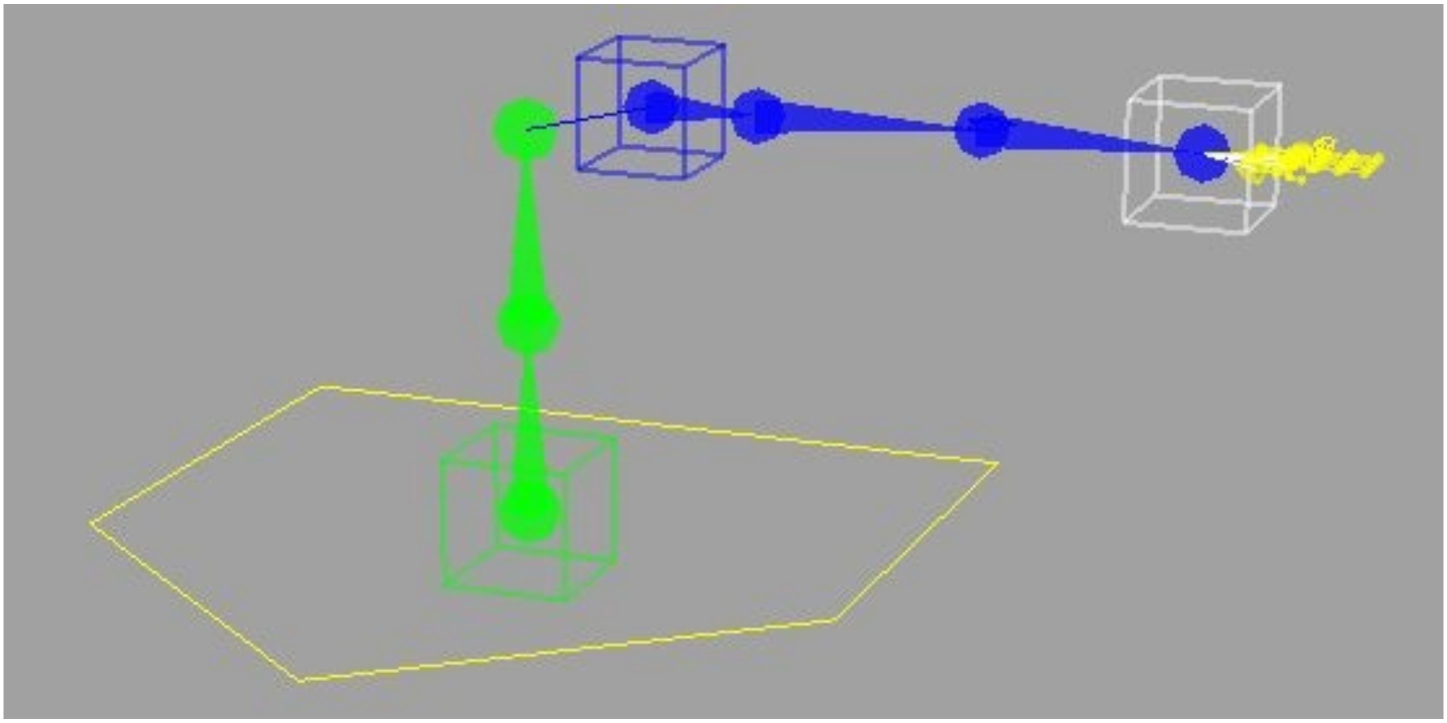
Note: Building the first part will also create the house-shaped upper body control. This control is a necessary part of the TSM3 rig – please don't delete it!

Here we've created a spine:



Once you have multiple parts in the scene you can attach them to each other. To do this, select the cube-shaped widget control of the part you want to attach, then the cube-shaped widget control of the part you want to attach it to. Select TSM3 > Attach Parts and the parts will be attached.

You'll see a thin line between them, representing the attachment:



Some parts have option boxes, which allow you to choose such options as mirroring and mirror axis (for limbs) and number of segments (for tails). Mirrored parts automatically mirror attachment – attach one side and the other side will be attached as well.

Note: This feature can be disabled using the Attach Parts option box.

To detach parts, select the 'child' widget control and choose TSM3 > Break Attachments.

In addition to the spine, limbs, fingers and head that you'll find in the biped, there are several other parts that you can create:

TAILS

Tails can be used to rig actual tails, but the flexibility they provide can also be used for tentacles, long necks, or any other long flexible part. When you build a tail, you're given the option for how many segments you want – from two to ten – which will determine how many controls your tail has.

Like limbs, tails have both IK and FK controls, which you can switch either from the Upper Body control, or to maintain the current pose, using the FK/IK Switch in the TSM3 Controls menu.

EYES

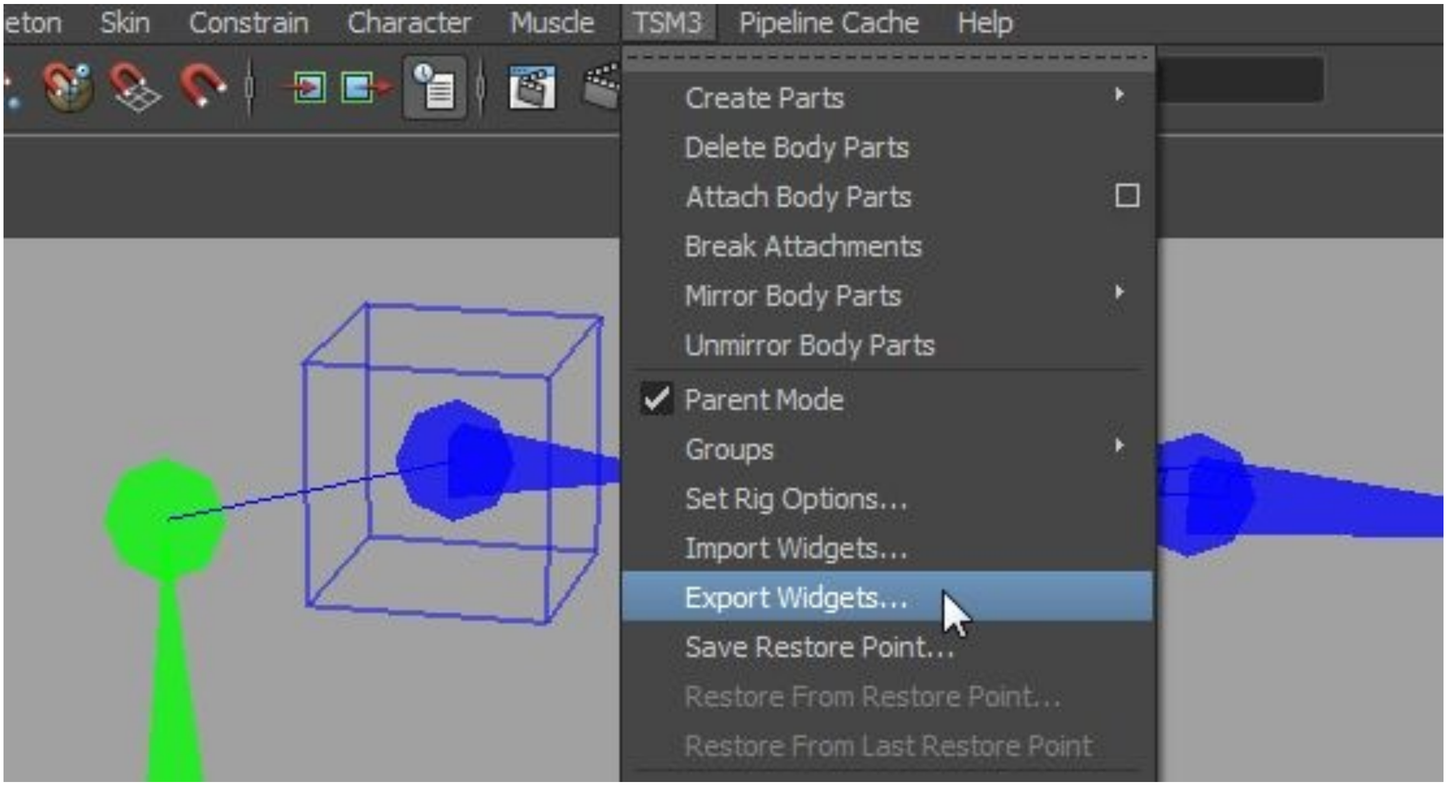
Eyes can be created singly or in mirrored pairs. When rigged, an eye is a single joint with both a direct control and an aim control. When created in a pair, there will also be a master control that moves the aim controls for both eyes.

SINGLE CONTROLS

For anything else that you might need a control for, you can add a single control. When rigged, a single control is just a single joint with a control.

EXPORTING WIDGETS

Once you've set up your widgets for a specific character, you may find that you want to reuse them on a different, but similarly proportioned character. To do this, you can select TSM3 > Export Widgets.

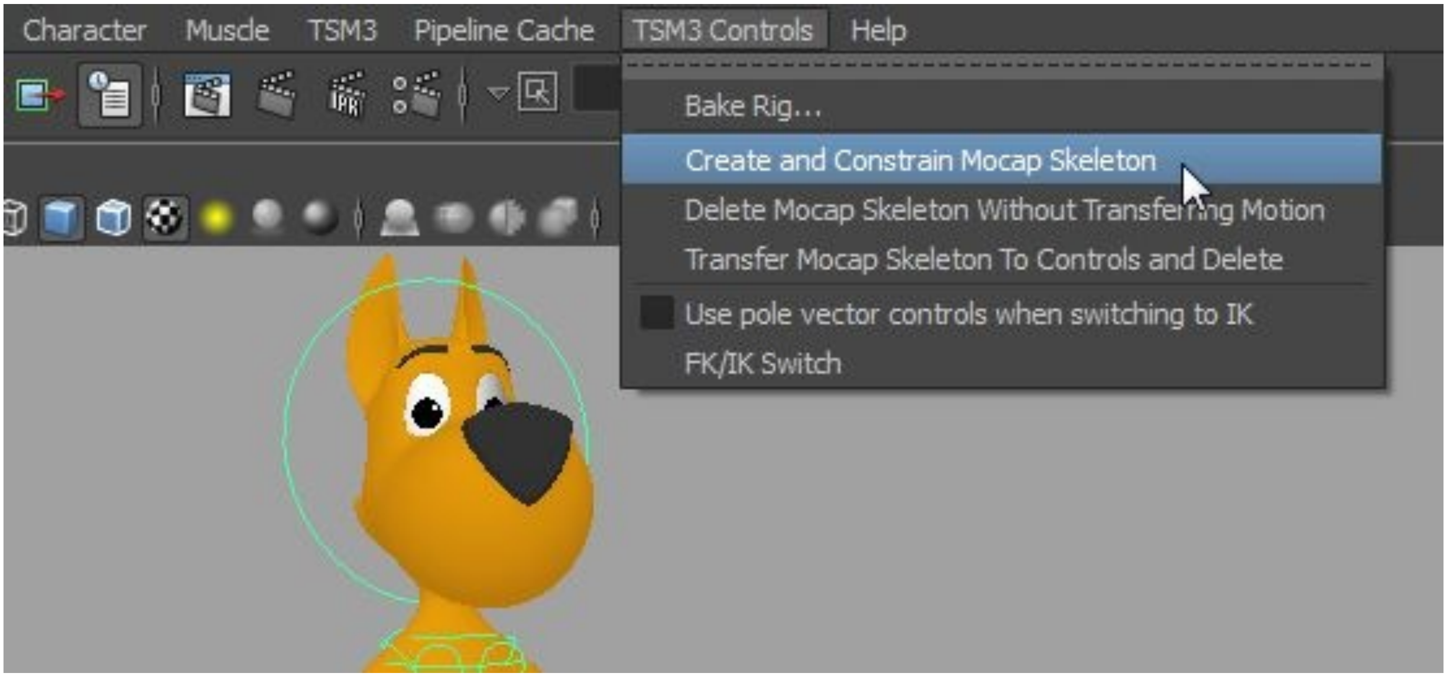


This will allow you to save a .json file with the widget information in it.

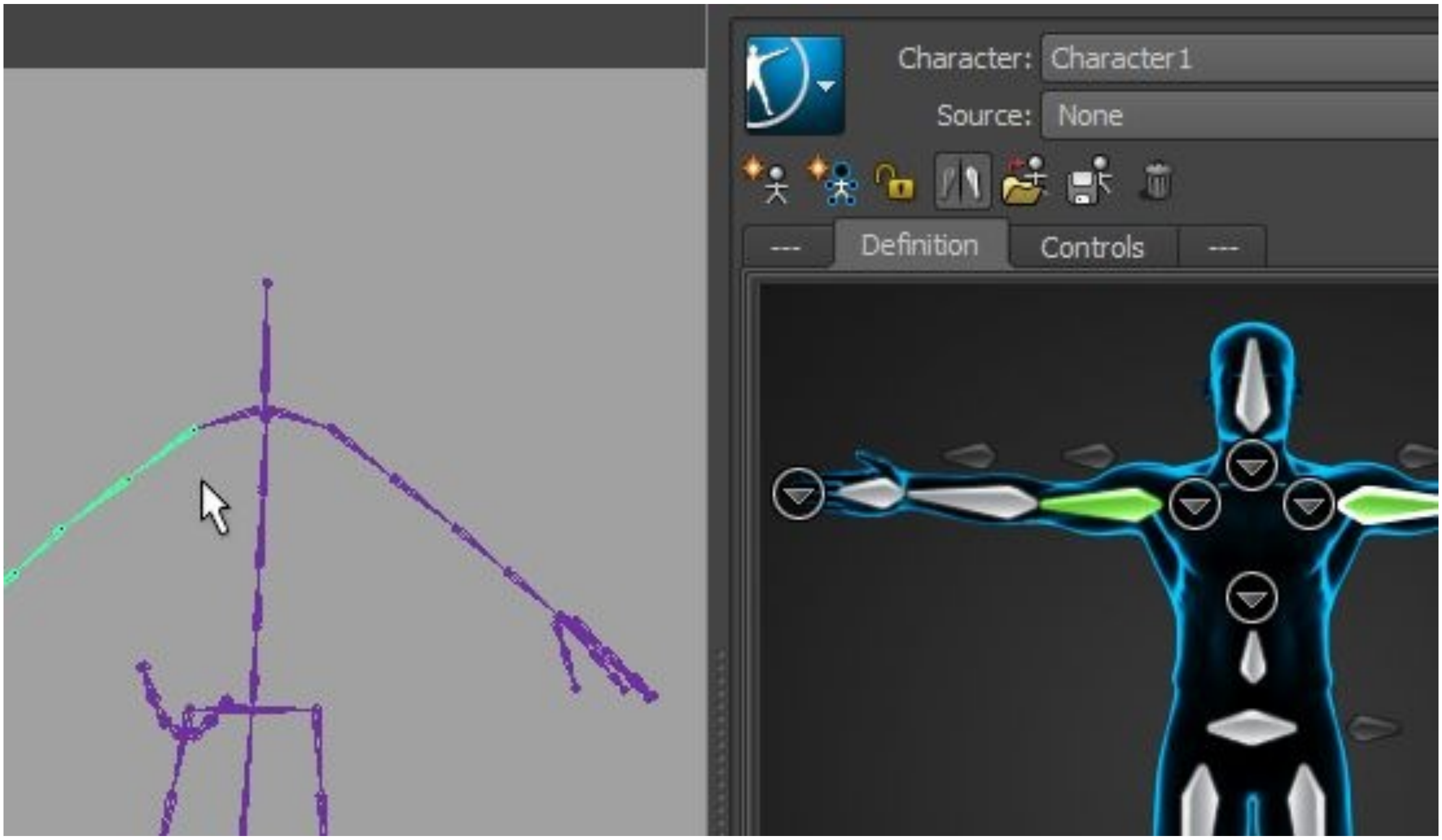
Once you've exported your widgets, open the file you want to import them into and select TSM3 > Import Widgets. Select the .json file you previously saved and your widgets will be loaded into the scene.

WORKING WITH MOCAP

To work with mocap data we recommend using Maya's HumanIK system to transfer it to your TSM3 rig, allowing you to use TSM3's more powerful controls to tweak the animation. To assist with this process, you can create a skeleton that drives the TSM3 rig. Select a control from the rig you want to drive, then go to the TSM3 Controls menu, and choose Create and Constrain Mocap Skeleton.


Character
Muscle
TSM3
Pipeline Cache
TSM3 Controls
Help
Bake Rig...
Create and Constrain Mocap Skeleton
Delete Mocap Skeleton Without Transferring Motion
Transfer Mocap Skeleton To Controls and Delete
Use pole vector controls when switching to IK
FK/IK Switch

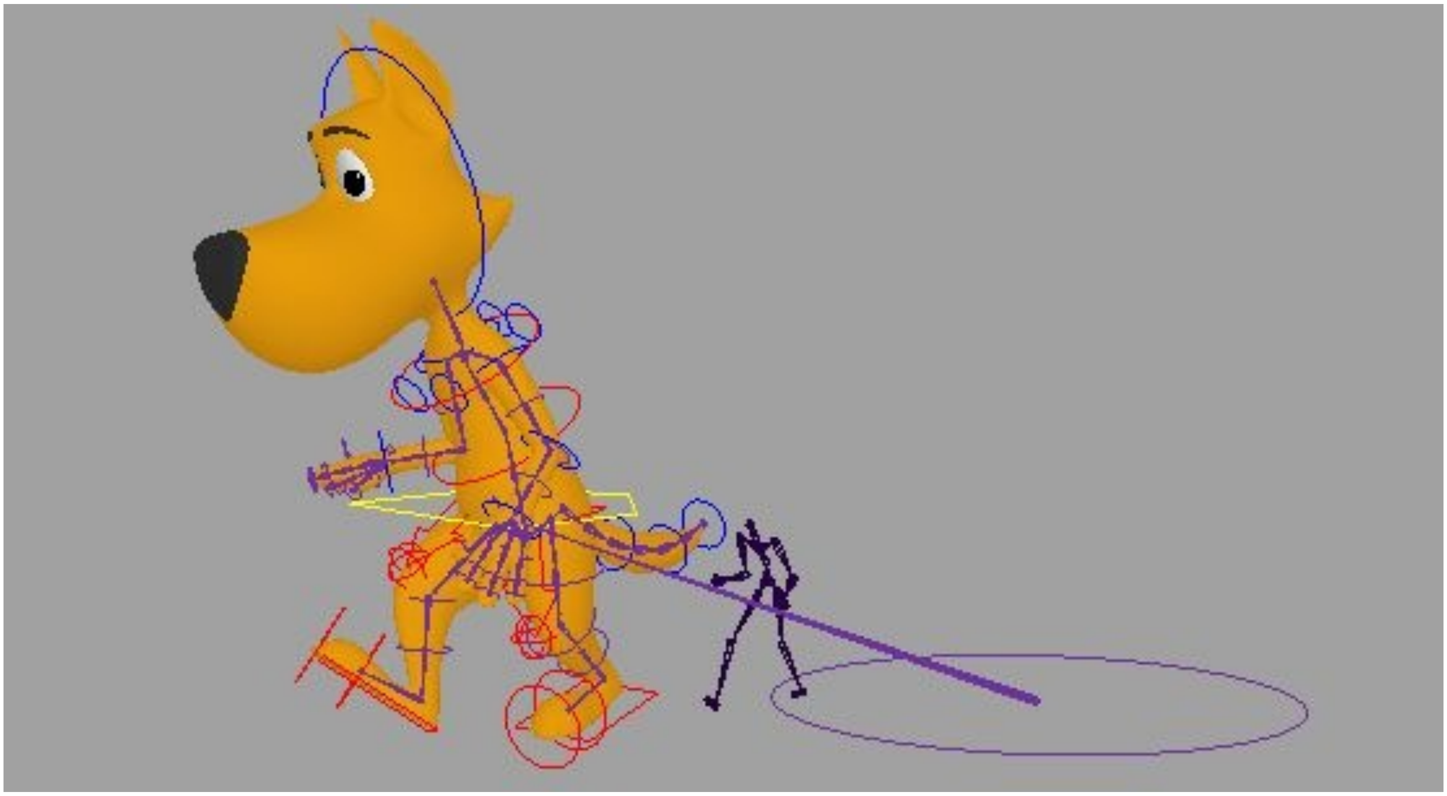
Next, you'll need to *define* the mocap skeleton using the HumanIK system. Hide everything but the skeleton and choose Skeleton > HumanIK. Then choose Define > Skeleton and start defining your skeleton, double clicking a bone in the HumanIK interface, and then selecting the appropriate joint.



Note: While most aspects of the definition should be intuitive, the HumanIK "Hips" should actually be defined as TSM3's Spine1 joint. You'll also need to adjust the arms to a horizontal position, if they aren't already, by rotating the joints of the skeleton.

When you're done, lock the skeleton definition using the lock/unlock button on the character definition toolbar, and then follow the same steps to define the skeleton of your mocap data.

Switch back to your TSM3 character (Character1, by default), and then set it's *source* to your mocap character (Character2, by default.) Your character should now be using the mocap skeleton's animation.



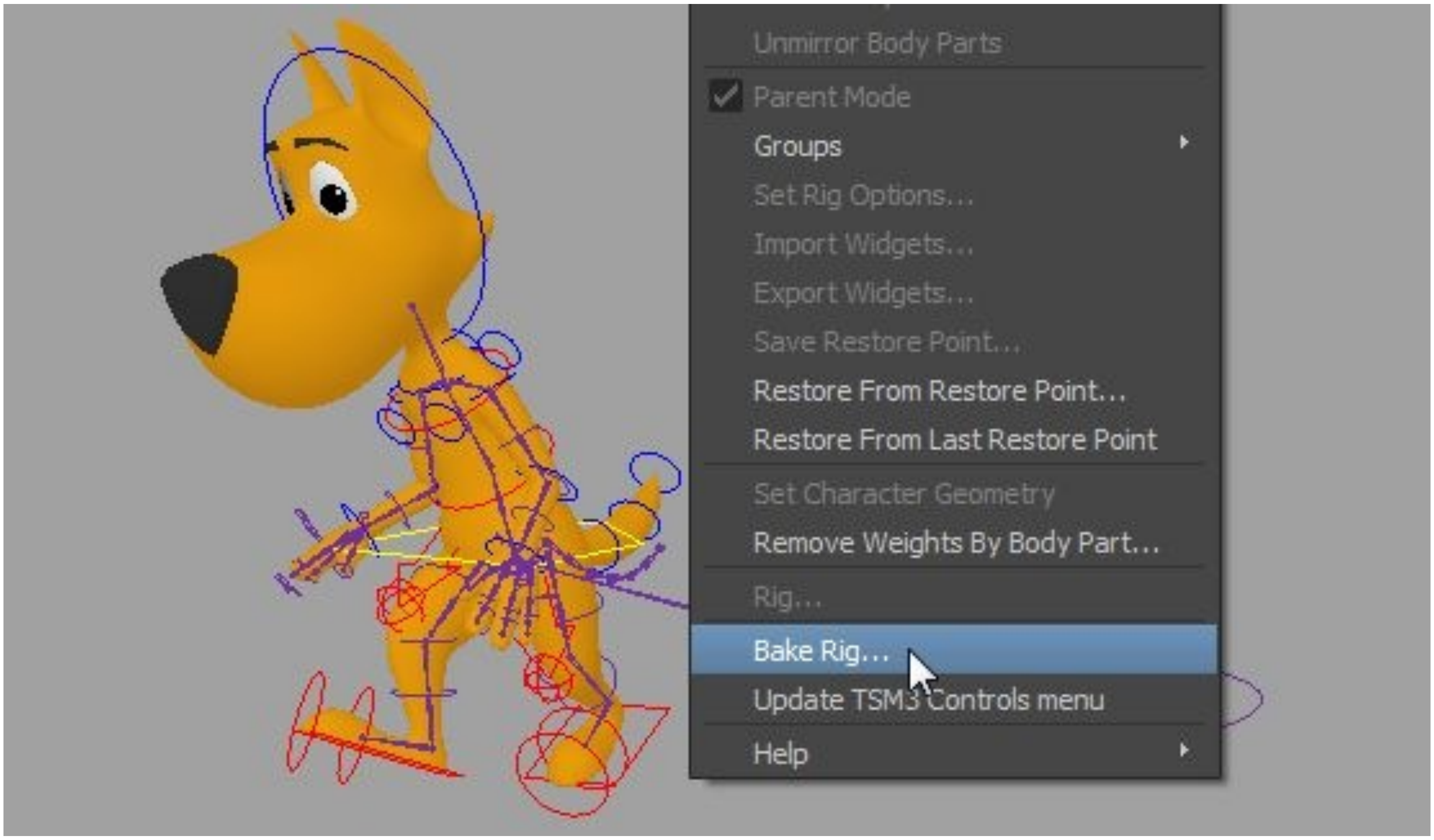
To bake this animation down to the control rig, go to the TSM3 Controls menu, then select Transfer Mocap Skeleton To Controls and Delete. This will automatically bake the animation to the controls, and delete their constraints to the skeleton.

Now that you've got the mocap data transferred onto your TSM3 rig, you can use animation layers to tweak the animation quickly and easily, using TSM3's more robust control scheme. Unlike HumanIK, TSM3 is stretchy and deformable, with fully independent torso and pelvis motion.

The script bakes the mocap data to both FK and IK controls, so you are free to switch between them or blend between them while editing the mocap data.

EXPORT TO ENGINE

Because Maya supports a number of features that most game engines don't, it's impossible to export the entire TSM3 rig into a game engine. Instead, we've provided automated baking functionality, which will bake all animation onto the character's underlying skeleton, and then remove the control rig. Since baking isn't undoable, you'll want to save your file first – and be sure not to save over it after! When you're ready to export, choose TSM3 > Bake Rig.



Once the file is baked, you can use Maya's export functionality to export to FBX, or some other format supported by your game engine.

Note: The engine itself must support joint translation as well as rotation. Please check the documentation of the engine you're considering using.

Again, the workflow here is as follows:

. 1) Save your animation file with the control rig intact

. 2) Bake, destroying the control rig

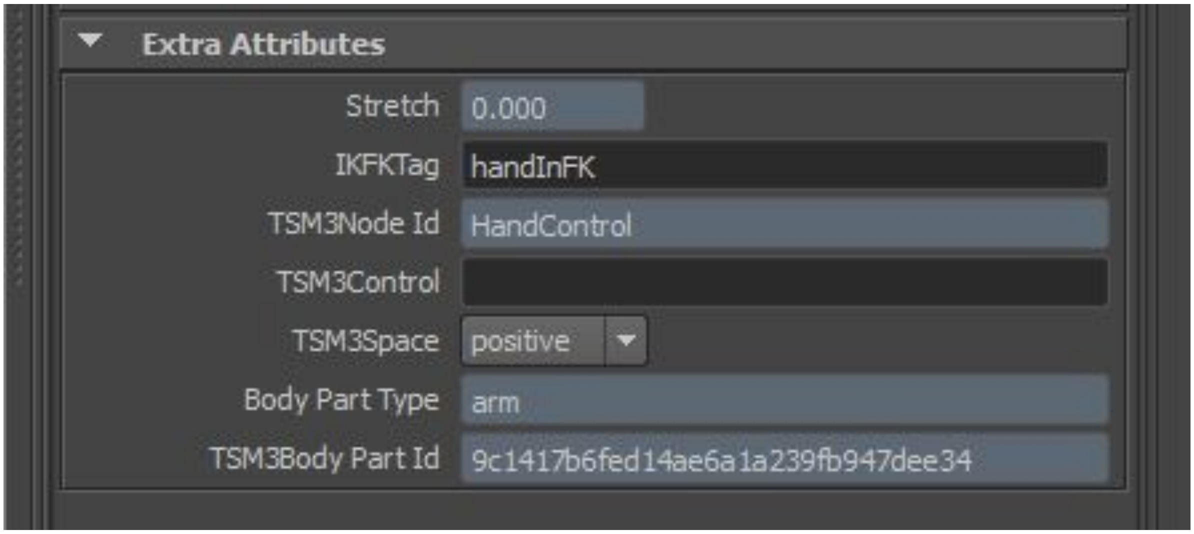
. 3) Export to FBX.

After Step 2, do NOT save over your animation file, or you'll lose the ability to edit the animation!

SCRIPTING

USING TSM3'S CUSTOM ATTRIBUTES

TSM3 controls have a number of custom attributes. If you need to interact with a TSM3 rig via scripting, you can use these attributes to do so. To see them, select a TSM3 control, open the attribute editor, and then open the 'Extra Attributes' section at the bottom:



Extra Attributes

Stretch	0.000
IKFKTag	handInFK
TSM3Node Id	HandControl
TSM3Control	
TSM3Space	positive
Body Part Type	arm
TSM3Body Part Id	9c1417b6fed14ae6a1a239fb947dee34

Stretch: Controls the amount of stretch.
IKFKTag: Used by the IK/FK system.
TSM3Node Id: A read-only attribute that indicates what kind of node this is.
TSM3Control: A flag to indicate that the node is a TSM3 control.
TSM3Space: Indicates whether the node is in a positive or negative space.
Body Part Type: Indicates what kind of body part the node belongs to.
TSM3Body Part Id: A unique identifier for the specific body part that the node belongs to.

You can interact with these attributes using MEL , the Python commands module, or PyMEL. If, for example, you wanted to get a list of all TSM3 controls in the scene, you could use the following script:

```
import pymel.core as pa

controlList = [control for control in pa.ls(type='transform') if
control.hasAttr('TSM3Control')]
```

ADDING YOUR OWN RIGGING

In many cases you may want to add your own rigging to a TSM3 rig—for instance, you may wish to implement a facial rig or clothing rig that sits on top of TSM3's rig. If you plan to use TSM3's Bake tool, you'll just need to keep a few simple rules in mind.

RIGGING TO BAKE

First, any joints you add to the TSM3 joint hierarchy that begins with TSM3WorldJoint will automatically be baked when the TSM3 'bake' function is used. For instance, eye joints would be parented to the head joint in that hierarchy. It is important to only add joints that you intend to bind to the mesh to this hierarchy, as you don't want extraneous nodes to be exported to your engine. The one exception to this is constraints, which are automatically removed after TSM3 bakes the hierarchy.

Second, anything else you add under the TSM3_Rig hierarchy will automatically be deleted when the TSM3 bake function is used. This is where you would add additional control curves or other rigging that you don't want to be exported with the character.

With this in mind, you can do additional rigging that will automatically be handled correctly by the TSM3 bake function. In most cases, that's all you'll need to know. However, if your additional rigging can't be constructed within those restrictions, you can set it up however you need to, and bake it via scripting instead.

First, you'll need to import PyMEL:

```
import pymel.core as pa
```

You can then import and call prepCharactersForExport with an additional argument that lists any additional nodes you want to bake, as a list of PyNodes. For example, if you have an extra joint called “joint1”:

```
from tsm3.scripts.prepForExport import prepCharactersForExport
prepCharactersForExport(additionalNodes=[pa.PyNode(‘joint1')])
```

This will do everything that would normally happen when you use the TSM3 menu to bake, as well as baking those additional node(s). However, if you just want to bake the TSM3 joints and any additional node(s) of your own, without deleting any of the rigging afterwards, you can import and call bakeKeyframes:

```
from tsm3.scripts.prepForExport import bakeKeyframes
bakeKeyframes(additionalNodes=[pa.PyNode('joint1')])
```

Note: Restoring the rig to widget mode will remove any custom rigging.

TROUBLESHOOTING & FAQ

Q: Can I set up my character's face with TSM3 for Maya?
A: No, TSM3 for Maya does not do facial rigs. We do have a product called The Face Machine specifically for faces.

Q: Where do I send a bug report?
A: anzovin.com/support. Be sure to include your system specs, version of Maya, and how you generated the problem. We may ask you to send your file for analysis. You may also email support@anzovin.com.

Q: How do I get a press copy for review?
A: Please contact sales@anzovin.com

SUPPORT

For any sales-related questions, like "Where's my serial number?" e-mail sales@anzovin.com.

For technical support, please visit www.anzovin.com/support and fill out a support request or e-mail our support team: support@anzovin.com.

When requesting support via email, please include relevant information: platform, OS version, and Maya version. Please be as specific as possible when describing the problem, and attach any files displaying the issue.

CONTACT US

For sales related questions, like “Where's my serial number?” please contact sales@anzovin.com . For technical support, please visit anzovin.com/support.

Please include as much information as possible about the problem, including your OS and Maya version, and whether you're running 32 or 64 bit Maya. You may also want to include a screenshot or a file that demonstrates the problem.