

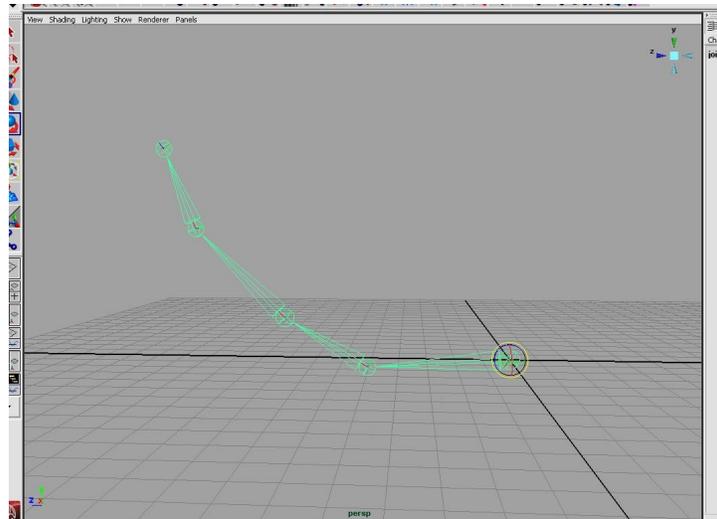
Introduction to Expressions

by Gary Wu

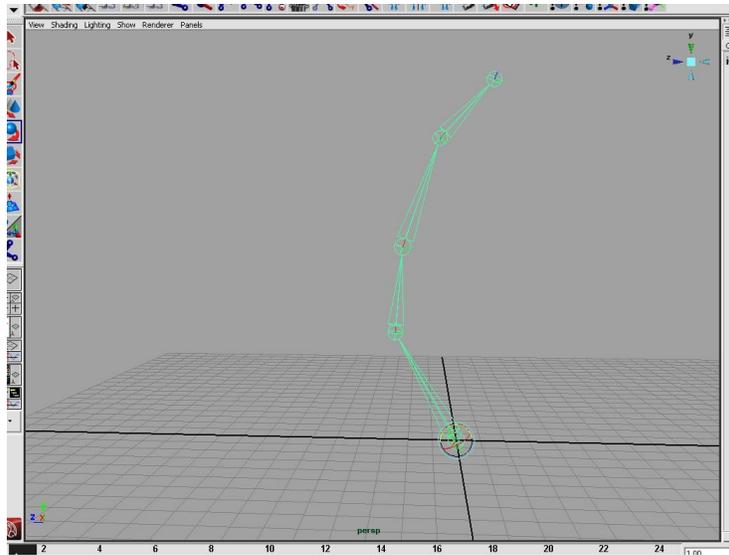
Here's a quick tutorial on using expressions.

What are expressions? Expressions are lines of code that behave like controllers on your rig. Instead of manipulating things by hand, you can let the computer take care of minor nuisances for you. In this example, we'll learn how to use expressions to make a joint hierarchy behave like a real tail.

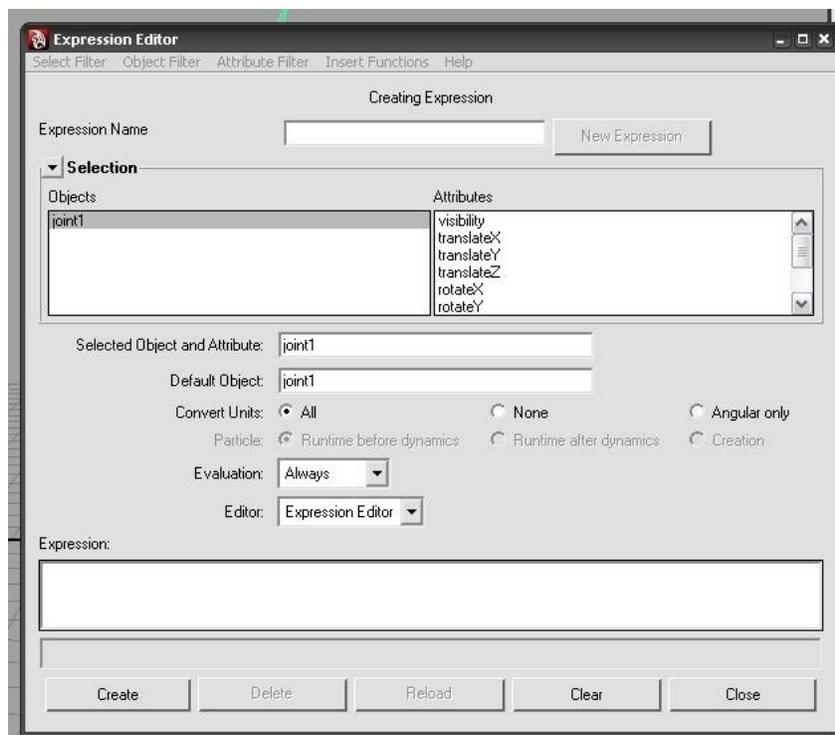
Start by drawing a joint sequence matching the look of a simple tail bone. Try to match the image below.



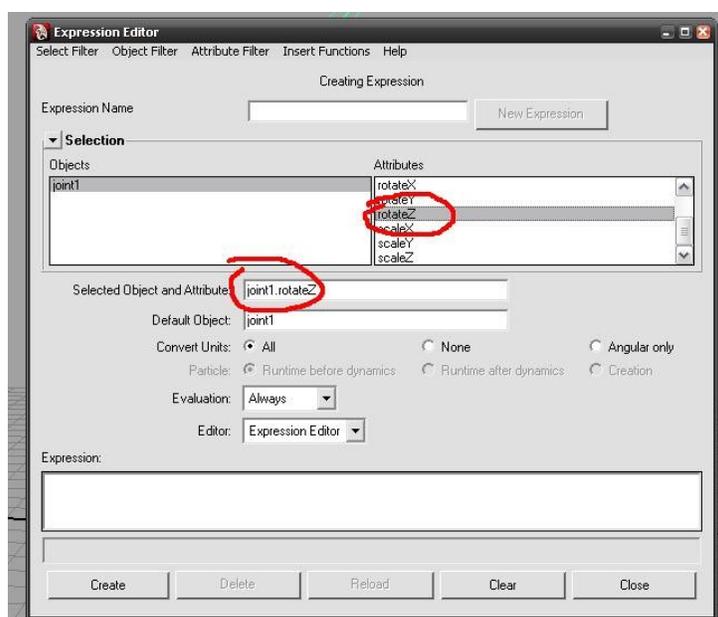
Notice if you rotate the start of the tail joint, all the other joints follow it very stiffly. This isn't how a tail behaves though. Think of a dog or cat tail. When the base rotates, the rotation is propagated to each subsequent tail joint. This means that if my base rotated by 30 degrees, each joint should rotate by some incremental amount. A picture of a rotation from the base of our stiff tail joints below:



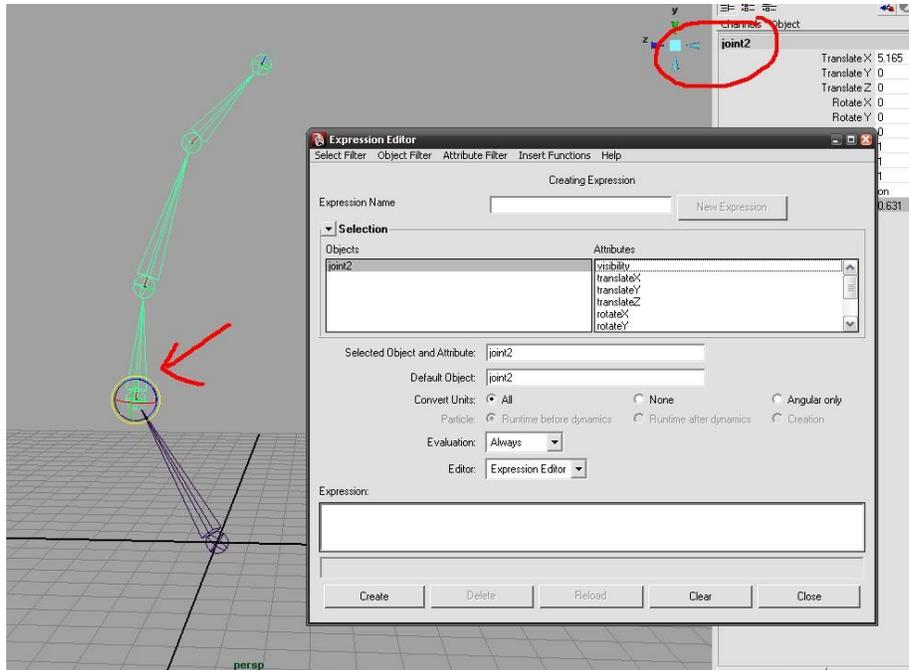
Now if we wanted to fix this, each rotate we would do on the base, we would have to individually select each subsequent joint and rotate it by some amount. We would repeat this down the chain until we reach the end. When animating, this becomes very, very, repetitive. To fix this, we use expressions. First, notice the naming of your joints. Your base joint (the one at the origin in the pictures) should be called joint1, then joint2, then joint3 .. and ends with joint5). Now select joint1. From the top menu, go to Window>Animation Editors>Expression Editor.



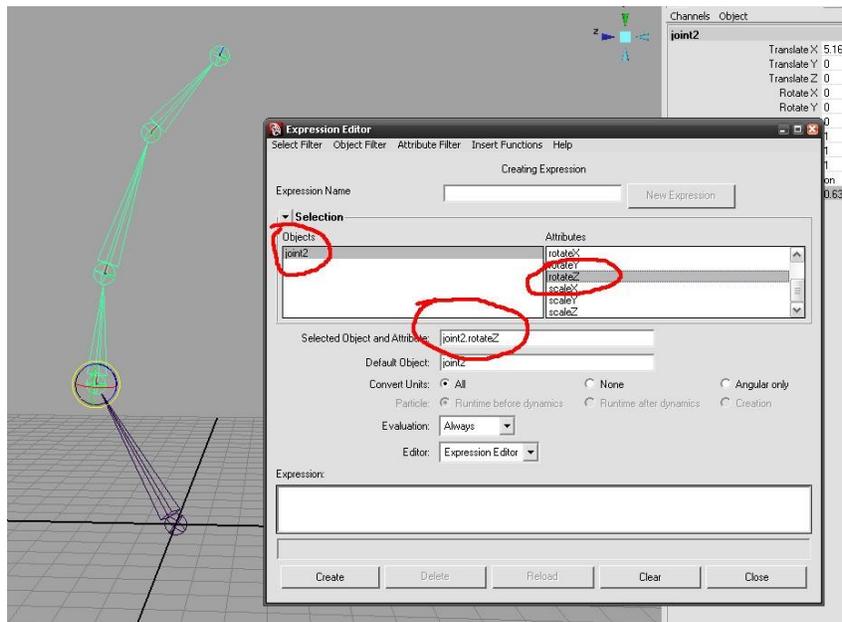
This is the expression editor. In the “Objects” window, this is the item you are currently selecting in Maya. In the “Attributes” window, these are the attributes that you can currently manipulate. If you click on these attributes, notice that the “Selected Object and Attribute” field changes accordingly. This field is important in writing your code that will control your joints. The most important field is the “Expression” field. This is where you will be writing your code. Now going back to Maya, I notice that I want my tail to rotate in the Z direction (this can be any direction you want...but for me it’s in the Z). Consequently, I will be looking at my rotateZ attribute for all of my joints. Now that I have identified the attribute I want to manipulate I go back to the expression editor and in the “Attributes” window, I choose “rotateZ.”



Notice the line of code in the “Selected Object and Attribute” field and remember or copy it down somewhere because this is going to be the code we are going to use in joint2. Now going back to your Maya, select joint2 (this should be the joint following joint1).

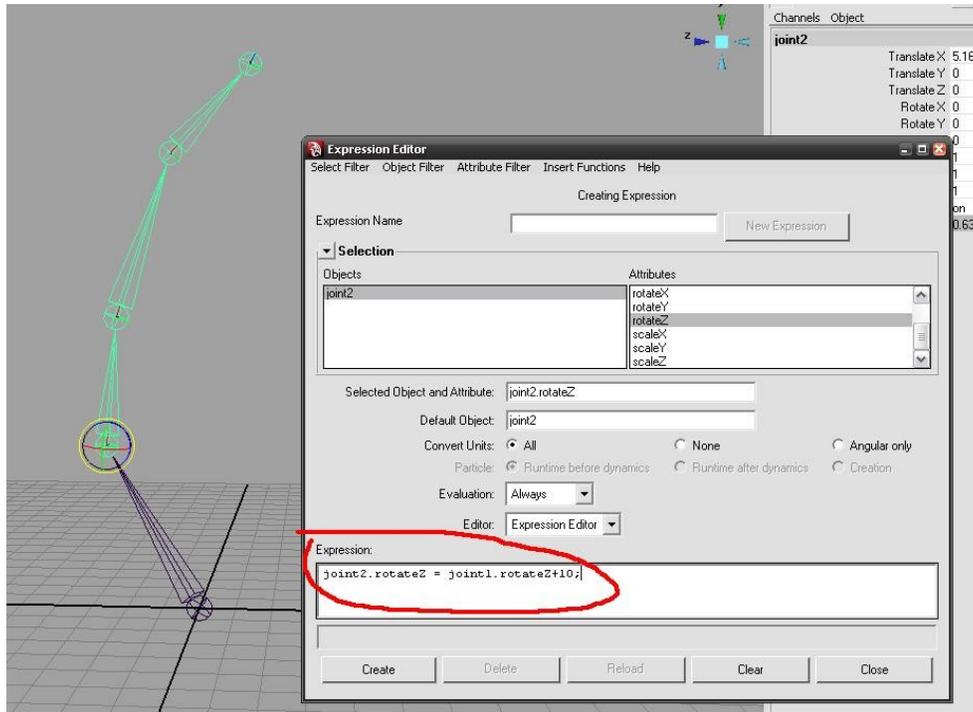


Now in your expression editor, notice that in the “Objects” field, the joint shown has been changed to joint2. Now select the “rotateZ” attribute in the “Attributes” window for joint2.



Now click in the “Expression” field and let’s write our first line of code. So now we ask ourselves how we want joint2 to behave in accordance with joint1. When we think of a tail moving, if the base joint rotates by some degree, the subsequent joints should be a bit larger than that degree of rotation to have a realistic tail movement. If you can’t picture this, follow the rest of the tutorial and you’ll see what I mean. So in the “Expression” field we’ll write:

joint2.rotateZ = joint1.rotateZ + 10;



Notice the semicolon at the end of the code. This is important. This ends the line of code and will not work unless you have this semicolon (think of it as a period in writing ... without it, we wouldn’t know where the end of a sentence would be ... a semicolon is a period to the compiler that reads and interprets this code). What this says is literally, whenever my joint1 rotates by some amount in the Z direction, I want that amount + 10 to be the amount I rotate in the Z director for joint2. When you finish the expression, hit the “Create” button. Now back in the Maya window, click on your join1 (freeze its transformations first!) and rotate it in the Z direction. You should notice that your joint2 behaves more realistically. And all this is done without having to touch individual joints. Now repeat the process for joint3. However, the only difference is that instead of having joint1 drive the rotation of joint3, it should be joint2. So the line of code you would put in the “Expression” field for joint3 would be:

joint3.rotateZ = joint2.rotateZ + 10;

Subsequently, the line of code you should write for joint4 should be:

joint4.rotateZ = joint3.rotateZ + 10;

This should make sense to you logically.

Hit create when you are down and repeat until the end. When you are down, you can select the base joint (joint1) and rotate it in the Z direction. Notice that the tail behaves more realistically now and the great thing is that all of this is now automated. We only touched on the rotate attribute for this tutorial but this is just a window to the power and control you can have with expressions. You can literally use any attribute in the expression to write to control something. This could be the translate of a controller, or the scale of a polygon object.