



Illustration by Simon Danaher
SIMON.DANAHER@NTLWORLD.COM

3D ANIMATION

CINEMA 4D R8 XPRESSO

Try out *C4D*'s exciting new expressions editor with our exclusive demo of *R8*

In issue 77, we took version 8 of *Cinema 4D* for a test-run, resulting in a verdict of four and a half out of five. One thing that came out of our review was the realisation that Xpresso, a superb tool for generating relationships between objects, is probably the new iteration's biggest single feature. Unfortunately, it's also likely to be the one users neglect the most. After all, unlike the new interface features, modelling tools and enhanced animation, it's not immediately apparent what Xpresso can do. An awful lot, as it turns out.

This tutorial will take you through the Xpresso workflow and show you just how simple it is to use. Creating your own Xpresso expressions is no more difficult than mastering drag-and-drop. Don't believe us? Then try it for yourself.

COFFEE, *Cinema*'s previous incarnation of expressions (still available in *R8*) is very good if you didn't mind a bit of maths and scripting, but for most of us it remains an unfathomable curiosity, suitable for high-end animators and programmers. Xpresso, however, offers a graphical approach that us mere mortals will find much easier to get to grips with. The tricky part is understanding which nodes to use and why.

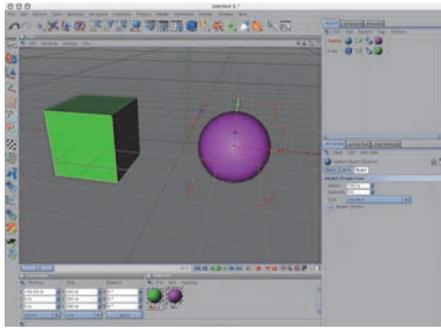
With this in mind, we'll take you through a series of Xpresso scenes, starting from the simplest possible expression to some fairly complex linkages. In this way, you'll learn how the most common features and nodes work, what's possible with Xpresso and how it can save you so much time and effort. With a little ingenuity, you'll be able to create your own expressions in no time.

→

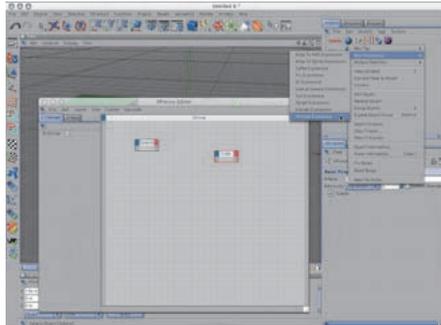


PART 1 GETTING STARTED

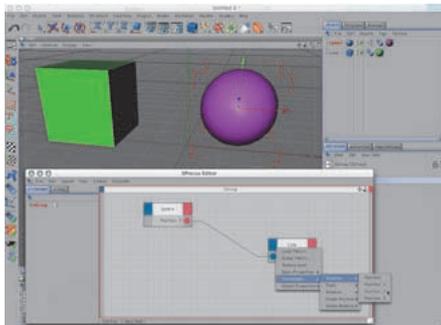
Xpresso isn't as hard as you might think – and here's why ↓



1 Xpresso aims to make expressions simple, and, by and large, it succeeds in doing just that. We'll begin with the simplest of Xpresso expressions, so create two primitives, say a sphere and a cube, and move them apart slightly.



2 Command/Ctrl-click on the sphere in the Object Manager and choose New Expression>Xpresso Expression to bring up the Xpresso Editor. Now drag the sphere and cube into it from the Object Manager to create two object nodes.



3 Each node has an Input (blue, left) and an Output (red, right), but at the moment these ports are undefined, so we need to add a specific type to each. From the sphere's Output menu (the red block), choose Co-ordinates>Position>Position Y. From the cube's Input menu, again choose Co-ordinates>Position>Position Y. Now connect the nodes by dragging from the sphere's Position Y output port to the cube's input port. Now, if you move the sphere in Y, the cube moves too.

INSIGHT THE BASICS

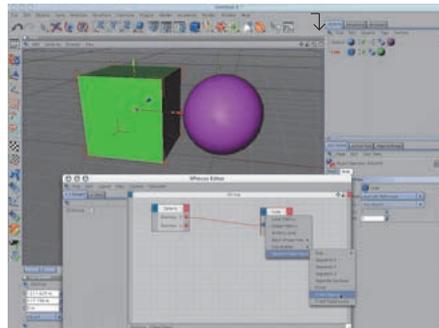
Xpresso is a node-based expressions editor, and for those who don't know, an expression is a relationship between objects within your 3D scene. The relationships usually involve programming, but by using Xpresso, you can manipulate objects using drag and drop – simplifying relational animation.

X, Y AND Z

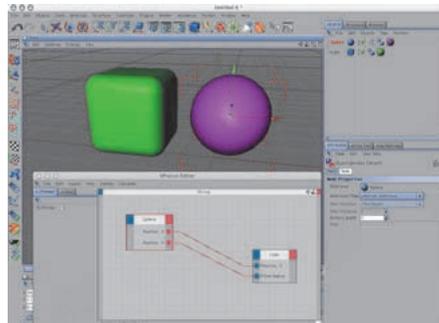
Note that the cube is locked to the Sphere's Y position, so if you try to move it in Y it won't budge. However, none of the other motion channels are affected, so you can still move it in X or Z or scale and rotate it. The effect is like creating a partial parent-child relationship that only affects one motion channel.

PART 2 GETTING MORE INVOLVED

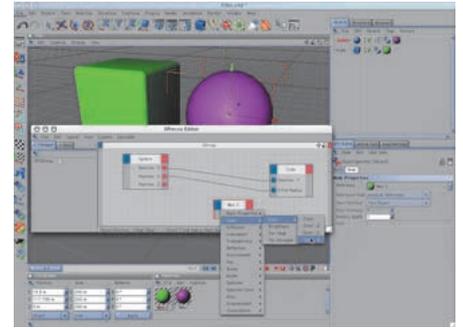
You can use Xpresso to connect unrelated properties ↓



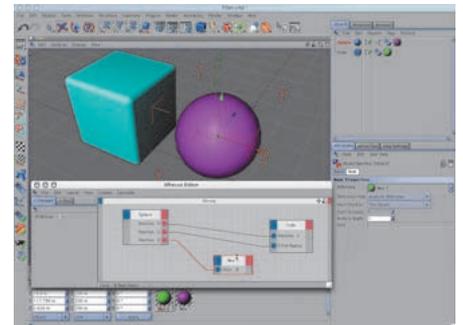
4 The wonderful thing about Xpresso is that you are not tied to linking inputs and outputs of the same type. Let's look at what else you can achieve. In the sphere's Output menu, add a Co-ordinates>Position>Position X port. In the cube's Input port, add an Object Properties>Fillet Radius port, and then connect the ports together.



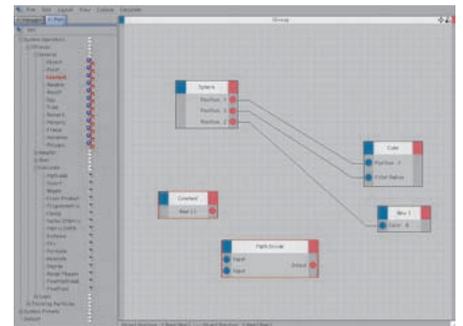
5 Nothing happens yet, because we haven't enabled the Fillet option in the cube's properties. So select the cube in the Object Manager, and, in the Attribute Manager, click the Object menu block. Here you'll find the Fillet option. Click Fillet to enable it. Now as you drag the sphere in X, the cube's Fillet radius changes interactively. Play a little to see how the objects interact. See the file fillet.c4d on the CD.



6 Just to show you once more that it's possible to connect totally unrelated properties, we'll use the sphere's Z position to control the cube material's colour. Add a new material in the Material Manager and apply it to the cube. Drag this material from the Material Manager into the Xpresso group window. The Node will be called New.1.



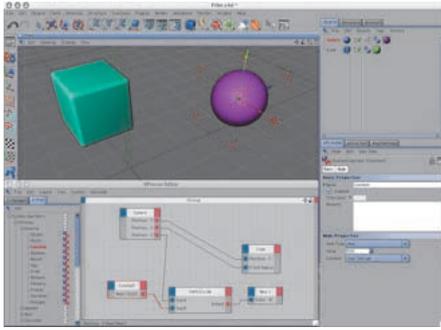
7 Add a Colour>Colour>Colour B port to the New.1 material node's input. Add a Position Z port to the sphere node's Output and connect the two. Now move the sphere in Z. Notice that it takes a minute movement of the sphere to change the colour of the cube, making it difficult to control.



8 To alter this, Click the X-Pool tab to the right of the Xpresso group view. Here you'll find all the different nodes. Go to System Operators>Xpresso>General and add a Constant node by dragging it into the Group view. Do the same with a System Operators>Xpresso>Calculate>Math:Add node.

INSIGHT PATTERNS CONFUSED?

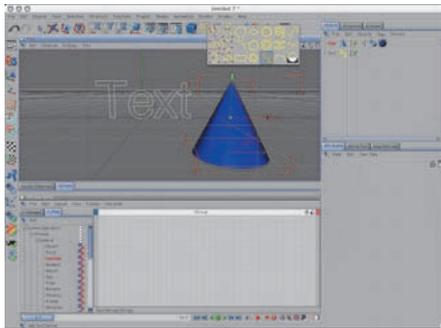
What we've done is taken the output of the sphere's Position Z port and divided it using the Math node, but with a constant value. The result? You now have to shift the sphere much further to change the cube's colour.



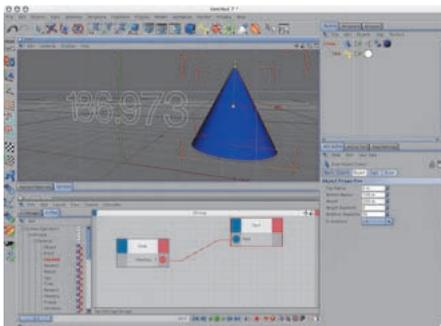
9 Connect the Position Z output of the sphere to the first (upper) input port of the Math node, and the Constant to the second input port. Connect the Output to the cube's Colour B input. Select the Math node, and in the Attribute Manager set it to Divide mode. Select the Constant node, then try out some values in the Attribute Manager. We ended up using 500. See Math.c4d on your CD.

PART 3 CREATING READ-OUTS

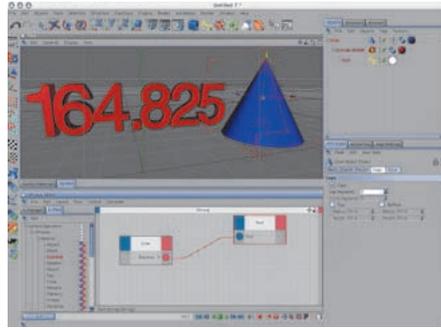
There may be a time when you need to see values of an object quickly ↴



10 Using Xpresso, you can build interface elements that provide live, on-screen feedback of object property values. Very useful. Add a cone to the scene, then a text object. Now add an Xpresso expression to the cone.



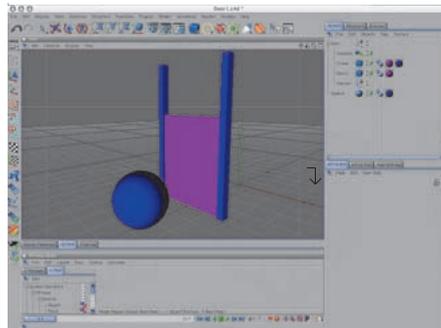
11 Parent the text to the cone in the Object Manager. Now drag the cone and text object into the Xpresso panel. Add a Position Y port to the cone's output and an Object Properties>Text port to the text object's input, then connect them up. Now the text object displays the values of the Cone's Y position in the editor view as you move it.



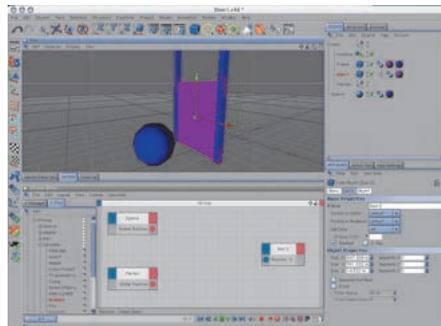
12 The text behaves just as it did before, so you can create a NURBs Extrude object, drop the text into it and get extruded 3D text. This is simply to show that adding the Xpresso expression doesn't change the nature of the objects or impose limits; they behave exactly the same as they did before.

PART 4 ANIMATION

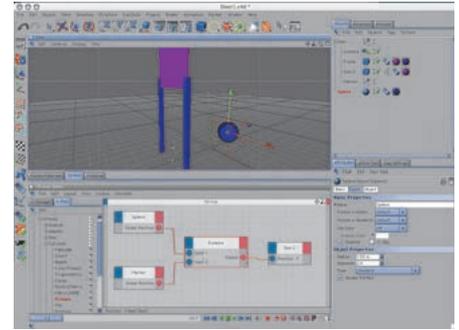
Xpresso comes in hand when creating animations too, of course ↴



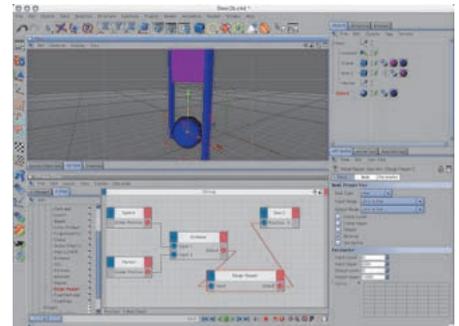
13 Let's say you have to animate an object travelling through a series of doors that need to open as the object passes through. Imagine 20 to 30 doors, and the animation you'd need to create. Now imagine having to change the animation of the object and then having to update all 20 doors by hand! Open the file Door1.c4d.



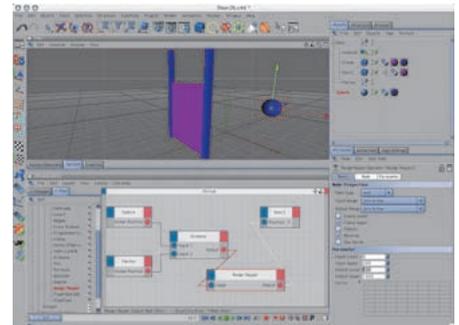
14 Here's a much simpler way to achieve the same results... Open the door hierarchy and add an Xpresso expression to the Door2 object. Open the Xpresso panel and drag the objects Door2, Marker and Sphere into it. Add a Co-ordinates>Global Position>Global Position Output port to the sphere and marker nodes, and a Co-ordinates>Position>Position Y input port to the Door2 node.



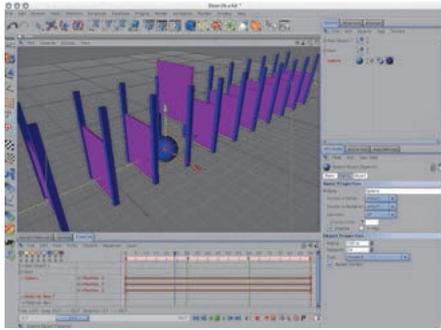
15 Add a Distance node to the Xpresso group (X-Pool, Xpresso>Calculate>Distance) and connect the marker and sphere to it. Connect the output of the distance to the Door2's Position input. Unfortunately, if you now move the sphere towards the door, it moves in the wrong direction, and as the sphere moves further away, the door keeps going!



16 To fix this, add a Range Mapper between the Distance and Door2 nodes (X-Pool, Xpresso>Calculate>Distance). This enables you to control the relationship between input and output. Select the Range Mapper and, in the Attribute Manager, click the Reverse option. In the Parameter section, set Input Upper to 200, Output Upper to 1000. Move the sphere to test the result.



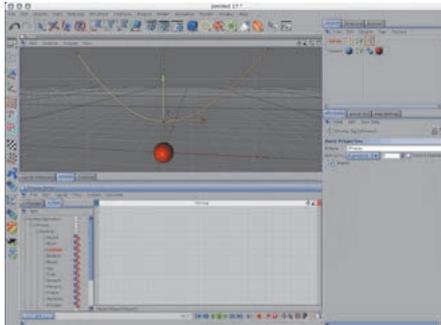
17 This sets the motion of the door. To stop it moving too much, though, click the Clamp Upper option, then set Output Lower to about 100 (this sets the closed position of the door). It won't go past this point because of the Clamp setting. See the file Door2b.c4d.



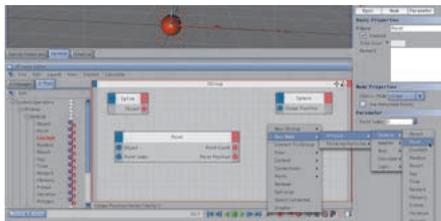
18 Select the Door hierarchy and choose Functions>Duplicate. Uncheck the Make Instances option and set a value for Z at about 2000 and the number of copies to 10. Now you're free to animate the sphere any way you like, and as it comes close to each door, they open automatically. See Door3.c4d on the CD.

PART 5 EXTRA CONTROL

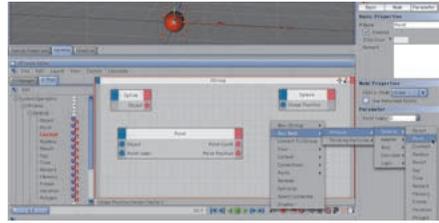
Controlling object components and geometry ↴



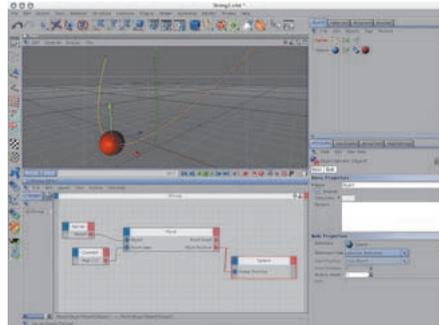
19 Controlling the object properties is one thing, but what if you could actually control object geometry, too? Well, you'll be pleased to hear you can – and quite easily. Here's a simple example that should open your eyes to a whole world of possibilities. Open the scene string.c4d from the CD.



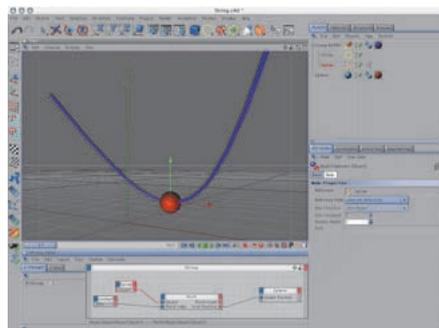
20 The scene consists of a simple three-point spline (with an empty Xpresso expression) and a sphere. Double-click the Xpresso expression tag in the Object Manager to open the Xpresso panel. Drag the sphere and spline into it. Now add a Point node; rather than getting this from the X-Pool, right-click (or Command-click) to view a contextual menu. Find it in New Node>Xpresso>General>Point.



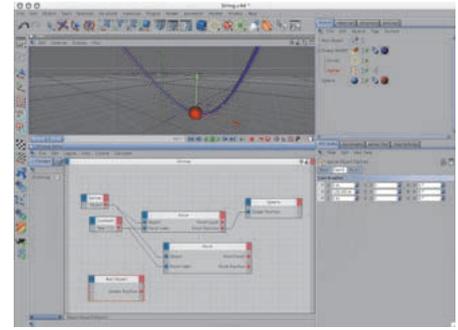
21 We want to link the sphere's position to the central point in the spline, so first add an Object Output port to the Spline and link it to the Object Input of the Point node. We have to tell the Point node which point we want to use, so add a Constant node and connect it to the Point Index (each point has an index number starting from 0).



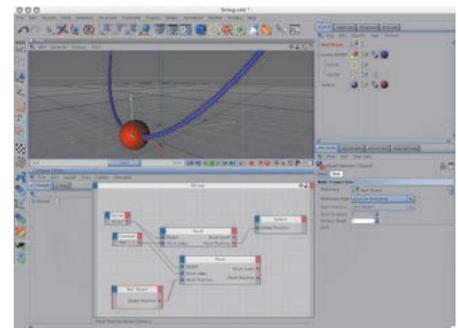
22 By selecting the Constant node and changing the value in the Attribute Manager, you can make the sphere jump to each of the points on the spline (0, 1 or 2). Set it to 1, then select the spline in the Object Manager and hit Return to enter Point mode. Select the middle point and move it; the sphere follows along, locked to it by the expression. Note that the spline interpolation type is set to Cubic so that the spline actually passes through the points.



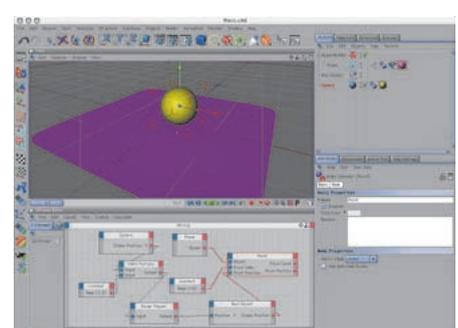
23 Add a Circle primitive and an Extrude NURBS and drag the spline into it, then the circle, to make a pipe. Reduce the circle's radius to about 4 to create a thin rope-like object. You can still move the spline point, even animate it using a PLA sequence in the timeline.



24 What we can also do is link that point to another object and animate that. Add a Null object (Object menu) and drag it into the Xpresso Group view. Select the Point node and copy and paste it to make a duplicate. Connect the Spline and Constant nodes, just like the other Point node.



25 If you click the red Input Port menu on the new Point node, you can add a Point Position port. Add a Global Position Output port to the Null node and link the two. Now move the Null – not forgetting to switch back to Model mode – and the point and sphere will follow along. See String2.c4d on the CD.

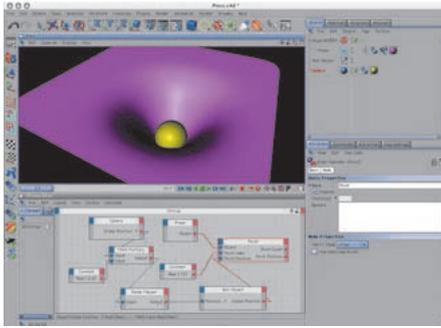


26 Let's take this further. Because we have access to objects' points and their positions, we can connect objects so that they deform other objects. Open the file Press.c4d from the CD and have a look at the Xpresso expression.

INSIGHT

SOPHISTICATED EFFECTS

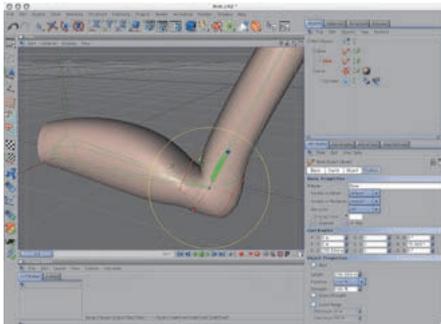
Imagine a character grabbing a rope... Link the points to nulls and you can place the latter in the hierarchy to follow any animation already created. For a character dragging a heavy load by rope, for instance, you could create an expression to make the rope stretch slightly with each tug.



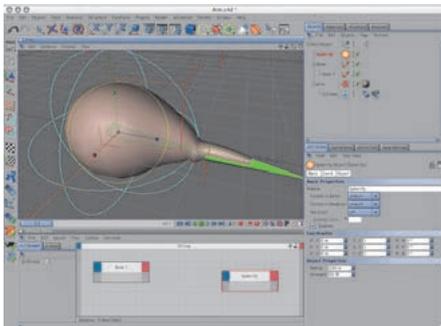
27 As you move the sphere in Y, it causes the plane to deform, yet only in one direction. To do this, we've linked one of the plane's points to a Null, then driven the Null with the sphere's Y position. Using the Clamp option or a Range Mapper enables us to limit the motion in one direction, while the Math-Multiply node makes the motion accelerate as the sphere passes, compensating for the fall-off caused by the HyperNURBS smoothing effect.

PART 6 CHARACTER ANIMATION

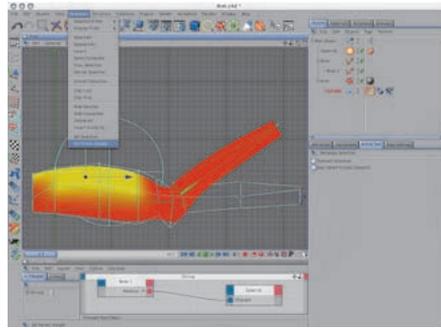
Time to put some muscle into it ↓



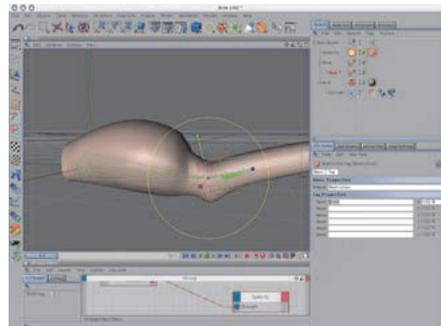
28 For character animation, Xpresso comes in very handy. By default, R8 doesn't come with any means to provide secondary skin animation, so we'll create an expression that will bulge as an arm bends. Open the CD file Arm.c4d.



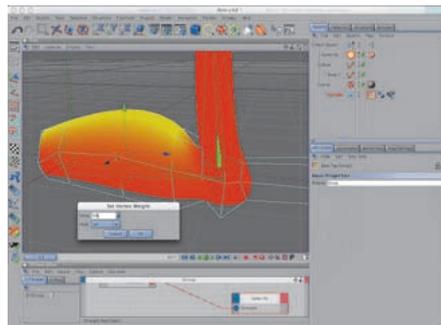
29 Add a Spherify deformer and drag it into the Null Object hierarchy. Now to control its behaviour... Add an Xpresso expression to the Null object, drag Bone.1 and Spherify into the Xpresso Group panel. It doesn't matter which object owns the Xpresso tag, unless you're using the Hierarchical reference properties of Object nodes.



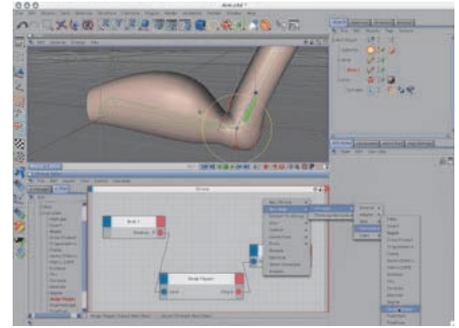
30 Select the Cylinder polygon object and enter Point mode by hitting Return. Choose the Rectangle selection tool and disable Only Select Visible Elements from the Active Tool Manager. In a side view, drag around the points of the upper arm that define the bicep area, then choose Selection>Set Vertex Weight, and in the panel that appears, click OK for 100 per cent.



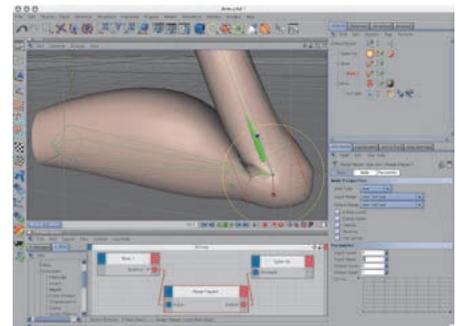
31 Add a Restriction tag to the Spherify object (from the Object Manager>File>New Tag>Restriction Tag) and enter Bicep in its first slot (in the Attribute Manager). Double-click the Weightmap tag on the Cylinder object and change its name to Bicep – the arm should now be properly weighted.



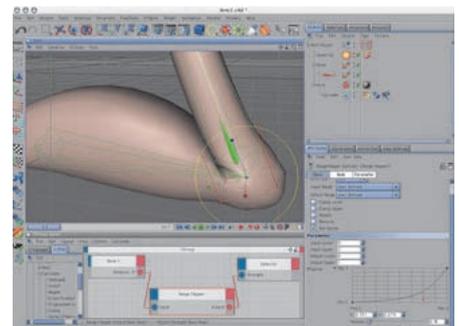
32 The effect of the Spherify deformer is too great, though. So, using the Restriction tag again, reduce the Value field from 100 per cent to 20 per cent. But it still looks wrong, especially from the top: the sides are expanding as much as the top. This is easy to fix by editing the weights. Select the side point and set their weights to 30 per cent using the Selection>Set Vertex Weights command.



33 Rotating the Bone.1 object, you can see the effect is shaping up, but the relationship between arm rotation and deformation isn't quite right. Back to Xpresso. Add a Range Mapper to the Xpresso group from the Calculate sub-set, then connect it between the two Object nodes.



34 Range Mapper enables you to modify the range of values through which an output controls an input, and vice versa. At the moment, the Spherify strength hits maximum too soon. To fix this, we need to broaden the range through which the input is working. Increase the Input Upper value from 1 to 2 or 3.



35 Finally, we can re-map the values using a spline curve, enabling us to create a non-linear relationship between values. Click in the Range Mapper's graph, adding a point at the start and in the middle. The end point is dragged up to the Max Y value, and the start and middle points to Min Y. Move the middle point to the right to create a steep curve – this makes the bulge accelerate as the arm bends closer. Take a look at Arm2.c4d on the CD. □