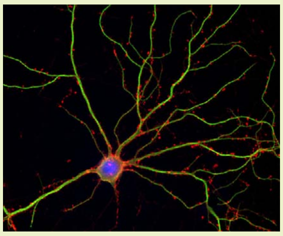
Forkroot

Forkroot is a plugin comprised of a set of objects for the creation of dendritic-like structures.
What are dendritic-like structures? - you may ask. Here are a few examples of dendritic-like structures:


Nerve cell

Lightning bolt

Eye veins

Tree branches

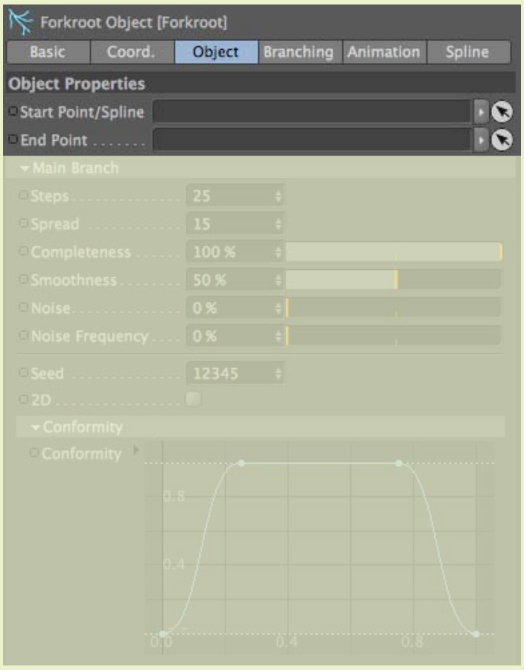
So, basically, Forkroot will allow you to create branch-like geometry. But dendritic-like structures sounds much better, doesn't it?
Forkroot is comprised of three generators:

The Forkroot object that creates a dendritic-like spline.

The Forkroot **Mesh** object that creates simple multi-segment geometry, based on a Forkroot object.

The Forkroot **Quad Mesh** object that creates simple uni-mesh geometry, based on a Forkroot object.

Forkroot object allows you to create highly controllable dendritic-like structure and, to do so, presents you with a generous set of controls. Let us see them all, one by one.



Object Properties
Start Point/Spline
End Point
Main Branch
Steps 25
Spread 15
Completeness 100 %
Smoothness 50 %
Noise 0 %
Noise Frequency 0 %
Seed 12345
2D
Conformity
Conformity

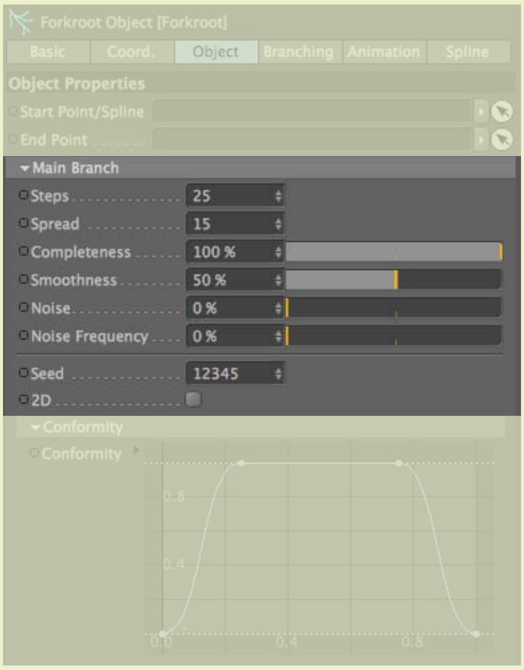
Placing an object in the **Start Point/Spline** field and another object in the **End Point** field creates a branch between those two objects.

Usually, two **Nulls** are the best choice, since they will not render, just providing a location in space.

If a spline is provided in the **Start Point/Spline** field, the branch will be created following the shape of that spline.

Using a spline is the method that provides more control over the shape of the branch.

Primitive or hand-drawn splines can be used.



Object Properties
Start Point/Spline
End Point
Main Branch
Steps 25
Spread 15
Completeness 100 %
Smoothness 50 %
Noise 0 %
Noise Frequency 0 %
Seed 12345
2D □
Conformity
Conformity

The **Main Branch** group contains all the parameters needed to adjust the shape and behaviour of the main branch... or trunk.

Steps - The number of segments of the main branch. Increasing this value will create a main branch with more detail.

Spread - This value defines the amount of random deviation of the generated spline from the guide spline or from the straight line between the start and end points, if provided.
Increasing this value will generated a more convoluted and twisted spline for the main branch.

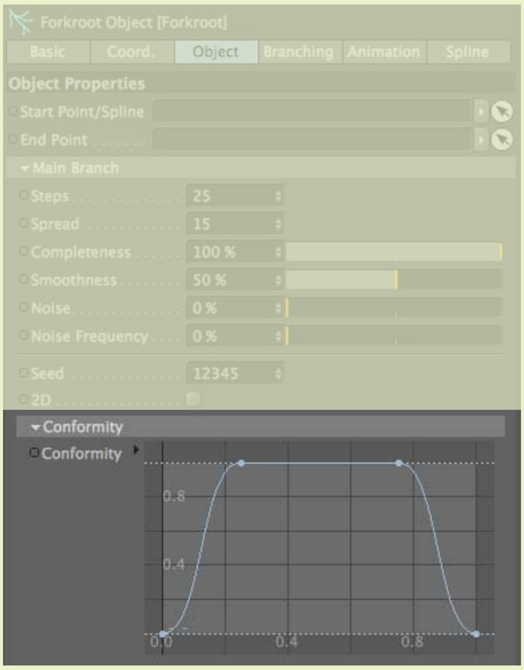
Completeness - This slider controls the growth of the branch. At 0% there is no branch and at 100% produces a full grown branch. More about this slider when discussing the mesh generating objects.

Smoothness - Increasing this value will smooth the resulting spline, making it less curved and/or spiky.

Noise and **Noise Frequency** - The Noise value adds a second layer of noise on top of the main random deviation. Increasing this value produces a spline that is more random and twisted.
The Noise Frequency defines how spiky and fine is this second layer of noise. Higher values produce finer noise.

Seed - Changing this number will produce a new random branch. This way it is possible to have several Forkroot objects with different shapes.

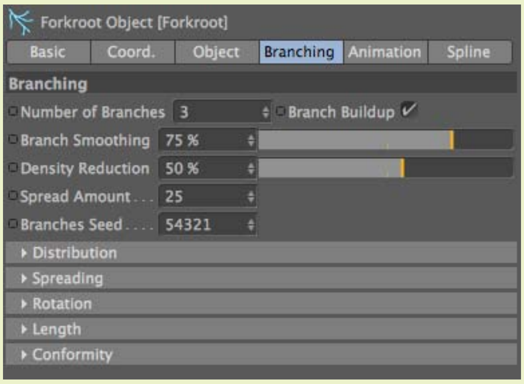
2D - This option turns on the production of a flat set of branches. It is mainly used for a very special and specific purpose. It will be explained in Appendix A.



Object Properties
Start Point/Spline
End Point
Main Branch
Steps 25
Spread 15
Completeness 100 %
Smoothness 50 %
Noise 0 %
Noise Frequency 0 %
Seed 12345
2D □
Conformity
Conformity

Conformity - This spline gizmo defines how much the generated spline deviates from the guide spline or from the straight line connecting the start and end points.
Knots placed at the bottom (Y=0) will make the generated spline coincident with the guide spline or the straight line connecting the start and end points.
Knots placed at the top (Y=1) will make the generated spline deviate the most from the guide spline or the straight line connecting the start and end points.
The maximum deviation is set by the **Spread** value but can be any random deviation.
The leftmost position (X=0) relates to the start of the branch. The rightmost position (X=1) relates to the end of the branch.
The default curve will create a branch that starts coincident with the guide spline or the straight line connecting the start and end points, spreads out in the middle and ends up coincident with the guide spline or the straight line connecting the start and end points.

It is possible to add as many knots to this gizmo as required to achieve the intended conformity.



Number of Branches 3 Branch Buildup ✓
Branch Smoothing 75 %
Density Reduction 50 %
Spread Amount . . . 25
Branches Seed . . . 54321

The **Branching** tab is where most of the control resides. Hence, it is the most complex tab in the Forkroot object.

Five adjusting groups (**Distribution**, **Spreading**, **Rotation**, **Length** and **Conformity**) are provided to control each secondary branch.

Their default state is closed as, when opened, they take up a lot of space in the Attribute Manager. Because of that, it is advisable to keep open just the groups that are absolutely necessary.

The main **Branching** parameters are:

Number of Branches - This value defines how may branches sprout from the main branch. It is advisable to set this number before adjusting the branches in the five adjusting groups because whenever the value of the Number of Branches changes, the five adjusting groups are reset to reflect the new set of branches.

So, BE WARNED!!! If you adjust the branches in the adjusting groups and, later, change the number of branches, your adjustments will be lost.

Branch Buildup - When this parameter is on, adjusting the Completeness value will produce branches that grow along with the growth of the main branch. If this parameter is off, the secondary branches sprout suddenly, as the main branch grows.

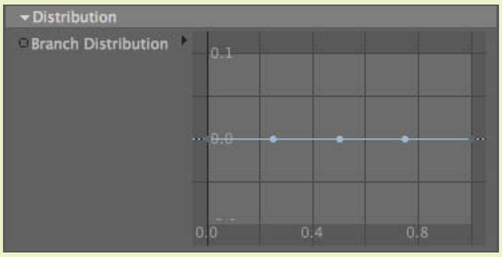
Branch Smoothing - It has the same effect as the Smoothness value of the main branch but this one applies to the secondary branches.

Spread Amount - It has the same effect as the Spread value of the main branch but this one applies to the secondary branches.

Branches Seed - It has the same effect as the Seed value of the main branch but this one applies to the secondary branches. This way, you can generate different sets of random branches without changing the randomness of the main branch.

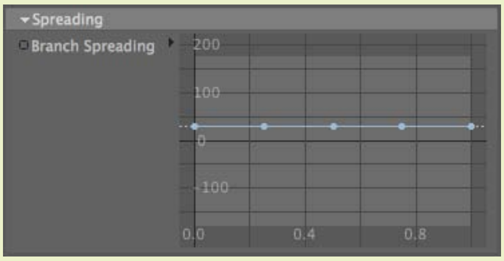
The adjustment groups will present a set of knots to control each branch. Each adjustment group has as many editable knots as there are branches. Sometimes additional non-editable knots are provided at the start and end. They are there for aesthetical reasons and because it usually makes more sense to understand their relation with the branches in the spline.
It is not possible to add or delete knots from these groups. The number of knots is controlled by the Number of Branches values.

SECOND WARNING!!! If you adjust the branches in the adjusting groups and, later, change the number of branches, your adjustments will be lost.



Distribution adjustment Group

This adjustment group allows the control of the placement of each branch along the main branch.
Moving the knots to the left will move the corresponding branches nearer the start of the main branch. Moving the knots to the right will move the corresponding branches nearer the end of the main branch.
The leftmost and rightmost non-editable knots stand for the start and end of the main branch and the editable knots can only be moved horizontally.
If a finer control is required, it is advisable to open the spline gizmo in an external window.



Branch Spreading adjustment Group

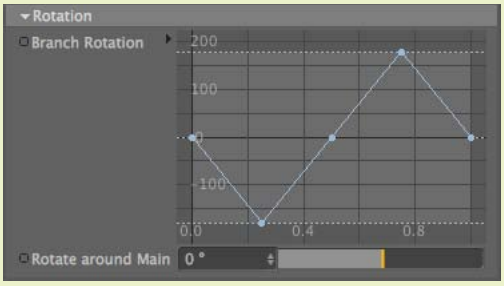
This adjustment group allows the control of how much each branch spreads out from the main branch.

Moving the knots up and down will rotate the corresponding branches in relation to the main branch.

It is possible to rotate each branch between -180° and +180˚, allowing for a full 360˚ rotation around the point of connection with the main branch.

The leftmost and rightmost non-editable knots stand for the start and end of the main branch and the editable knots can only be moved vertically.

If a finer control is required, it is advisable to open the spline gizmo in an external window.



Rotate around Main 0 °

Branch Rotation adjustment Group

This adjustment group allows the control of how much each branch is rotated around the tangent axis along the main branch.

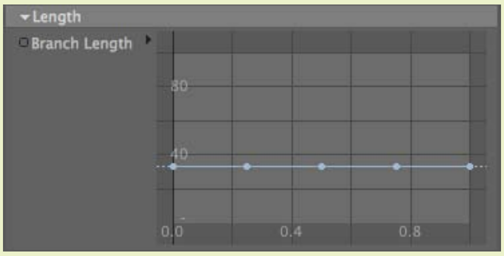
Moving the knots up and down will rotate the corresponding branches in relation to a tangent to the main branch.

It is possible to rotate each branch between -180° and +180˚, allowing for a full 360˚ rotation around the tangent with the main branch.

The leftmost and rightmost non-editable knots stand for the start and end of the main branch and the editable knots can only be moved vertically.

If a finer control is required, it is advisable to open the spline gizmo in an external window.

If all the branches need to be rotated at once, this can be achieved with the **Rotate around Main** slider.



Branch Length adjustment Group

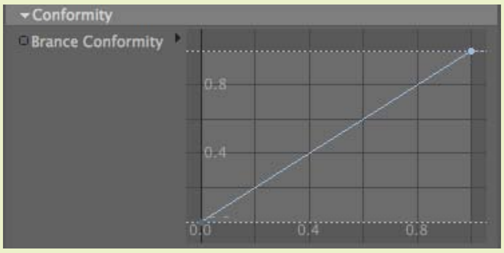
This adjustment group allows the control of how long each branch is in relation to the size of the main branch.

Moving the knots up and down will scale the corresponding branches in relation to the size of the main branch.

It is possible to set a size between 0% and 100% of the size of the main branch.

The leftmost and rightmost non-editable knots stand for the start and end of the main branch and the editable knots can only be moved vertically.

If a finer control is required, it is advisable to open the spline gizmo in an external window.

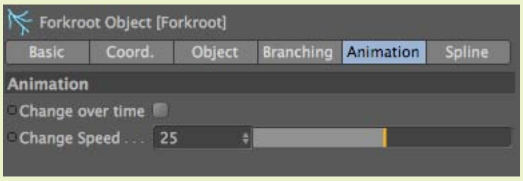


Branch Conformity adjustment Group

This adjustment group has the same effect as the **Conformity** group of the main branch but this one applies to the secondary branches.

The first knot is not editable because the beginning of the branch ALWAYS rests in the main branch.

It is possible to add as many knots to this gizmo as required to achieve the intended conformity.



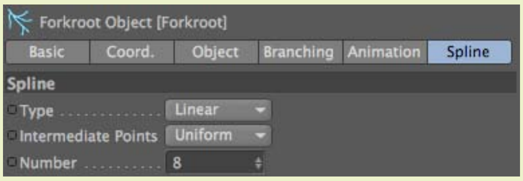
Forkroot Object [Forkroot]

Animation

Change over time □

Change Speed . . . 25

By default, the main branch and the secondary branches are not automatically animated.
But, some dendritic-like structures have random, ondulating movement.
If that is the intended behaviour, the **Change over time** parameter can be turned on.
The speed at which the main branch and the secondary branches move and twitch can be controlled by the **Change Speed** parameter.



Forkroot Object [Forkroot]

Spline

Type . . . . Linear

Intermediate Points Uniform

Number 8

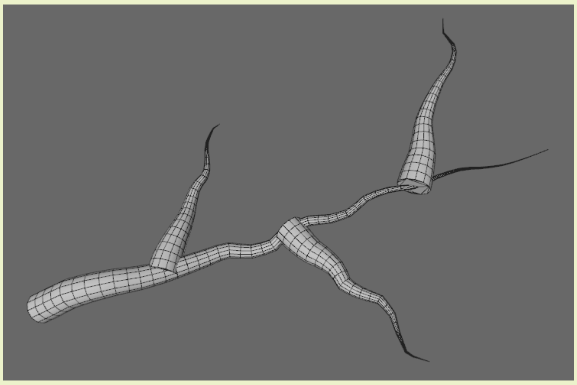
The generated spline can be of type **Linear**, **Cubic**, **Akima** or **B-Spline**.
The **Bezier** type is not allowed because the generated tangents could lead to a weird looking curve.

The type of **Intermediate Points** can be **None**, **Natural** or **Uniform**. The **Adaptive** and **Subdivided** types are not allowed because they produce interpolations that have a very variable density.
Finally, it is possible to define how many intermediate points are generated between control points.

The Forkroot object creates dendritic-like splines but splines don't render. So, to have something coming out of a render, polygonal geometry is required.

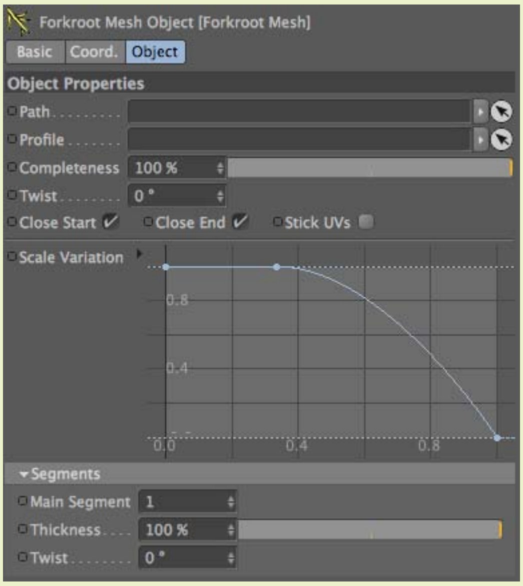
The Forkroot object can be placed inside a **SweepNURBS** object and suitable geometry can be obtained this way. But there are a couple of problems associated with **SweepNURBS** being used to generate geometry.

When tapered branches are created (by using the **Scale** spline gizmo in the **Details** group of the **SweepNURBS** object), all branches start out with the same thickness, including the main branch. So, if the main branch is thick at the start and thin at the end, all the secondary branches will do the same. But they will start with the same width that the main branch has at its start. Like this:



Another problem is with the animation of the growth of the branch.
If the branch is animated with the **End Growth** of the **SweepNURS** object, the main branch will grow first and then each branch in succession.
This may be the intended behaviour but, usually, the **Completeness** value of the Forkroot object created far better results.

The Forkroot toolset provides two ways of creating geometry for dendritic-like structures.
The parameters of each geometry generator are basically the same so, an explanation will be provided for just one and, after that, the differences between both will be presented.



Forkroot Mesh Object [Forkroot Mesh]

Basic | Coord. | Object

Object Properties

Path

Profile

Completeness 100 %

Twist 0 °

Close Start ✓ Close End ✓ Stick UVs □

Scale Variation

Segments

Main Segment 1

Thickness 100 %

Twist 0 °

The Forkroot Mesh object is, basically, a **SweepNURBS** object on steroids.
In the **Path** field, a forkroot object is required.
In the **Profile** field, any spline can be used. This spline will be sweeped along the splines generated by the Forkroot object.
So, it is advised to use a **Profile** that is a closed spline and with a shape similar to a circle.

The way the axis is generated is also very important.
The spline must be generated in the XY plane, with the Z axis perpendicular to that plane.

Actually, you have the same worries that you have when dealing with a standard **SweepNURBS**.
It is also advised to keep the intermediate points of both the **Path** and the **Profile** splines as low as possible to create speedier and cleaner geometry.

The **Completeness** value is very similar to the **Completeness** value of the Forkroot object. But, when animating the growth of a dendritic-like structure, it is preferable to use the parameter of the Forkroot Mesh object. This will produce a smoother animation as the spline is not being recalculated with different lengths for each new frame.

It is also possible to twist the profile as it is travelling along the forkroot spline, using the **Twist** value. Positive or negative values are allowed.

The start and end of the main branch (and ends of the secondary branches can be opened or closed, depending on the value of the **Close Start** and **Close End** parameters, respectively.

If the **Stick UVs** value is **Off** (it is **Off**, by default), it is as if the material attached to the branch is already stretched and the branches grow along it. If it is **On**, the material grows along with the branches as they grow.

The **Scale Variation** spline gizmo provides an excellent control of how thin the branches (main and secondary) become along their length.

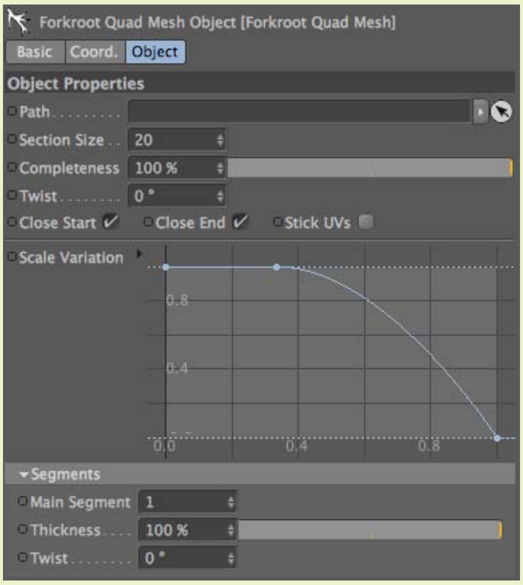
In the **Segments** group, it is possible to control which of the segments is considered the main segment, using the **Main Segment** value. The main segment is the one that is used for the main branch. Usually, the main segment is the segment number 1.

The **Thickness** value defines how much thinner the secondary branches are, in relation to the main branch. This value is provided as a percentage of the size of the Profile.

Finally, the secondary branches can have their own twist, independent of the twist value of the main branch. This twist along the length of the secondary branches is defined by the **Twist** value.

If you convert the Forkroot Mesh into an editable object, you will verify that it is all made with **SweepNURBS** objects. But they are being cleverly controlled by the code inside Forkroot Mesh.

Usually, it is advised to create simpler, lighter geometry and then, smooth and increase the density of the geometry with an **HyperNURBS** object. That can easily be done with the Forkroot Mesh object. But the connections between the main branch and the secondary branches will not be perfect or "organic".
That is because the **HyperNURBS** is being provided with a set of independent objects and it is smoothing each one, individually.
For more organic results, the Forkroot Quad Mesh object is provided.



Forkroot Quad Mesh Object [Forkroot Quad Mesh]

Object Properties

- Path
- Section Size 20
- Completeness 100 %
- Twist 0 °
- Close Start ✓ Close End ✓ Stick UVs □
- Scale Variation

Segments

- Main Segment 1
- Thickness 100 %
- Twist 0 °

The Forkroot Quad Mesh object has exactly the same parameter as the Forkroot Mesh object except that it replaces the **Profile field** with a **Section Size** field.
That is because the section of the branches generated by the object are ALWAYS square.
The **Section Size** field controls the size of the section of the main branch.

The Forkroot Quad Mesh object will make its best to smoothly connect the main branch with the secondary branches but, if some kinkiness is detected (by you, the branch-maker), minor adjustments in the distribution and/or spread and/or rotation of the branches in the Forkroot object will solve that.

The full effect of the Forkroot Quad Mesh object will only be noticeable when it is placed inside an **HyperNURBS** object.

The Forkroot Mesh and the Forkroot Quad Mesh objects are created without a Phong tag, by default. One may be added for smoother display and rendering.

Appendix A - Attaching a Forkroot object to a shape

It is possible to attach a Forkroot object to another object. To do so, the object where the Forkroot object will be attached needs to have very clean and contiguous UVs. That is usually the case when dealing with primitive object. But it is specially important if we are dealing with polygonal sculpted/edited objects.

For this process to work better the Forkroot object needs to be perfectly flat. That is what the **2D** parameter is good for. So, start by drawing a spline in the top view.

It is **VERY IMPORTANT** that the original spline is drawn in the top view as it needs to be perfectly flat and parallel to the XZ plane.

Now create a Forkroot object, assign the spline to it, turn on the **2D** parameter and edit your branches any way you like. It is advisable to be viewing the result in the top view.

When you are happy with the result of the Forkroot object, create a **Surface deformer**. Make it a child of the Forkroot object .

Drag the object to which you want to attach the Forkroot object into the **Surface** field of the **Surface deformer** object and adjust its parameters until you have the object in the location you want.

NOTICE THAT THE TYPE OF MAPPING IN THE SURFACE DEFORMER NEEDS TO BE MAPPING (U,V) OR MAPPING (V,U) AND NOT PROJECTION.

You can now use a Forkroot Mesh object or a Forkroot Quad Mesh object to generate geometry.

Addendum

The Forkroot object now has some additional buttons. On top of each adjustment spline there is a **Reset** button. Pressing that button will reset the spline to its default state but using the number of branches defined in the **Number of Branches** field.

These buttons are specially useful if, by any change, some knots were deleted or added to any of the adjustment splines. The Forkroot object will make its best to prevent you from deleting or adding knots but if, somehow, the number of knots of any of the adjustment splines does not match the number of branches set in the **Number of Branches** field, Forkroot will **NOT** create a Forkroot spline.

In this case, reset the offending adjustment spline.