

Creation of a custom Grow-FX Foliage



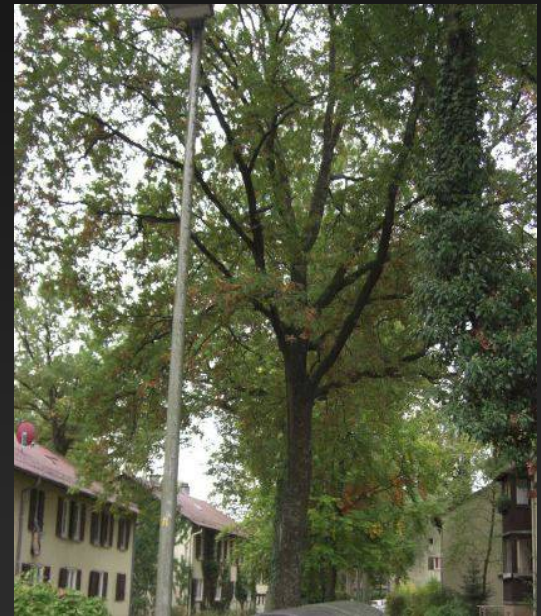
xoio

A. Basics / Research

Let me first point out, how much we really dig that piece of software. Opposed to other tools we used in our studio to grow foliage it may not be as "biological" in its approach, but that makes it more flexible to adjust your growing objects the way YOU want them to behave. Furthermore GrowFX can handle a LOT of geometry all inside of Max, which is indispensable for realistic trees. Not to forget it is actively developed these days so you can expect more to come from the friendly developers which is important for our studio.

Before we get started in Max it is indispensable to do some research first, to have image references of trees you can always come back to. Otherwise your trees will turn out generic and probably not very realistic. In this tutorial we want to cover the building of a tree for a specific project. We had several site-pictures, and the trees needed to be part of the new building scheme.

Additionally we were looking for winter images to get a better impression of the branching of the trees.

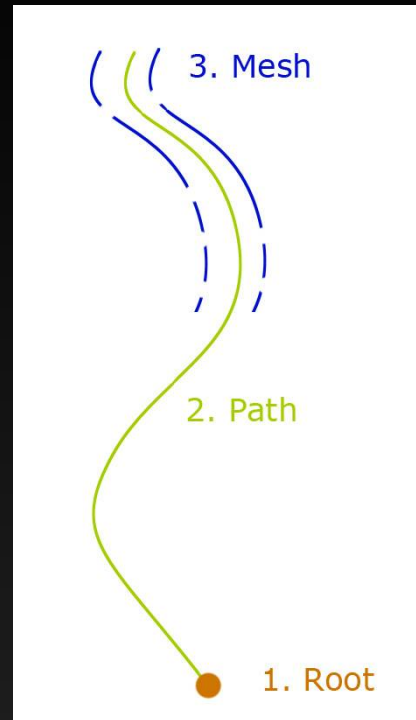


B. Grow FX Basics

First let me quickly explain the basic setup of any growFX Object: Any Growth Object consists of one or more Elements.

Each Element has 3 basic Categories:

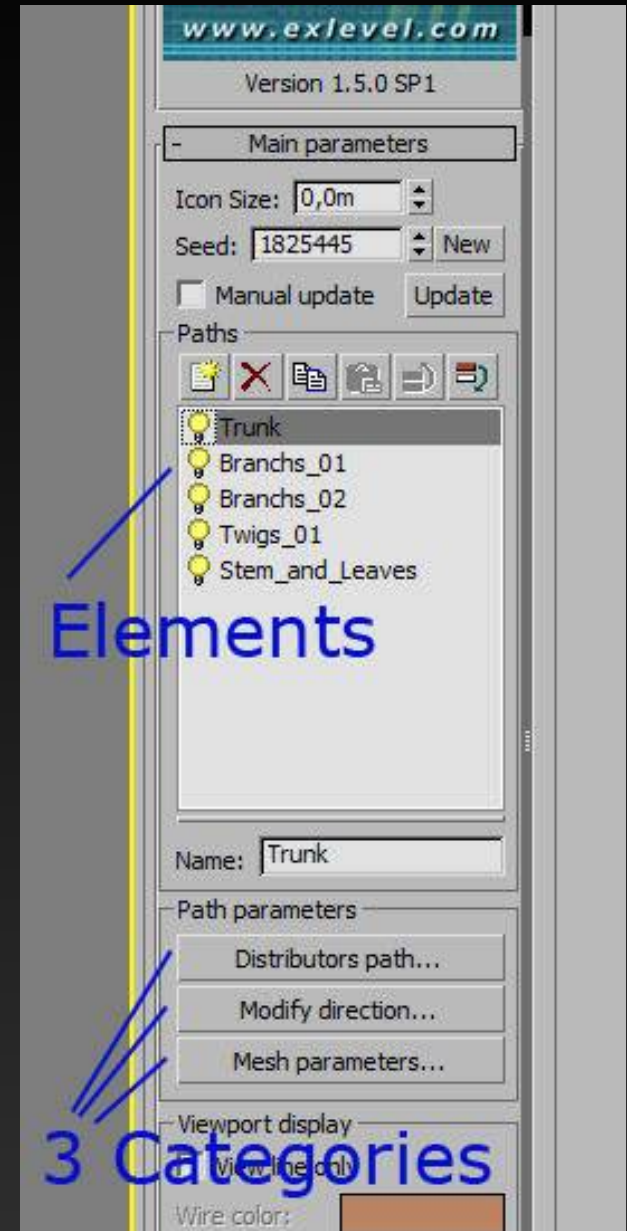
1. The Root (Distributor Window)
2. The Path (Modify Window)
3. The Mesh (Mesh parameter window)



1. The Root tells, where the Element starts and in which direction it takes off.
2. The Path defines, how the Element is growing, how long it is and how its path evolves over its distance.
3. The Mesh finally produces a renderable Geometry. This can be a branch but also various other kinds of custom geometry like leaves, fruits, etc.

That is actually the order in which you should work your way up.

Each one of these categories in Grow FX has a separate window, you can open, which makes it quite easy to treat them separately.



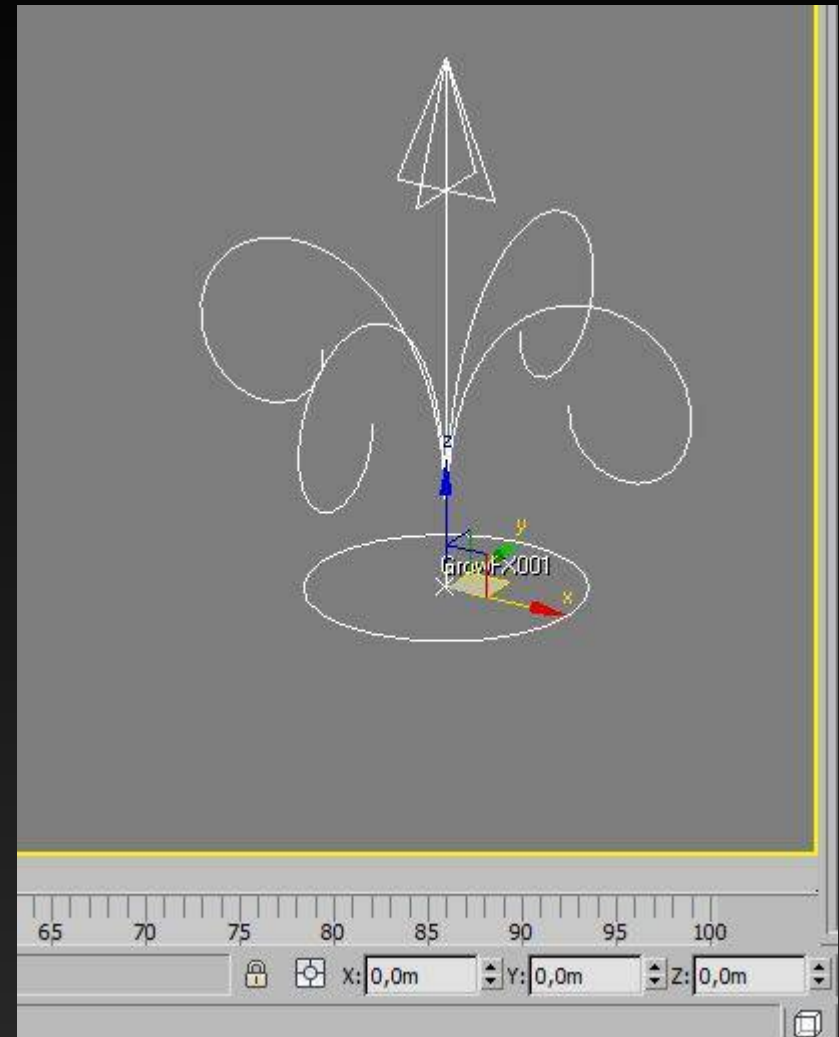
C. Generate your Grow Fx Object

Let's start with the trunk. Before you generate anything it is a good advice to check your units. Otherwise you will spent hours tweaking your plants on a wrong scale and rescaling is always prone to errors.

We generate a growfx-object, resulting in a placeholder icon-object.

I preferably generate these objects in a separate scene, since the calculation load can get quite extreme at a later point.

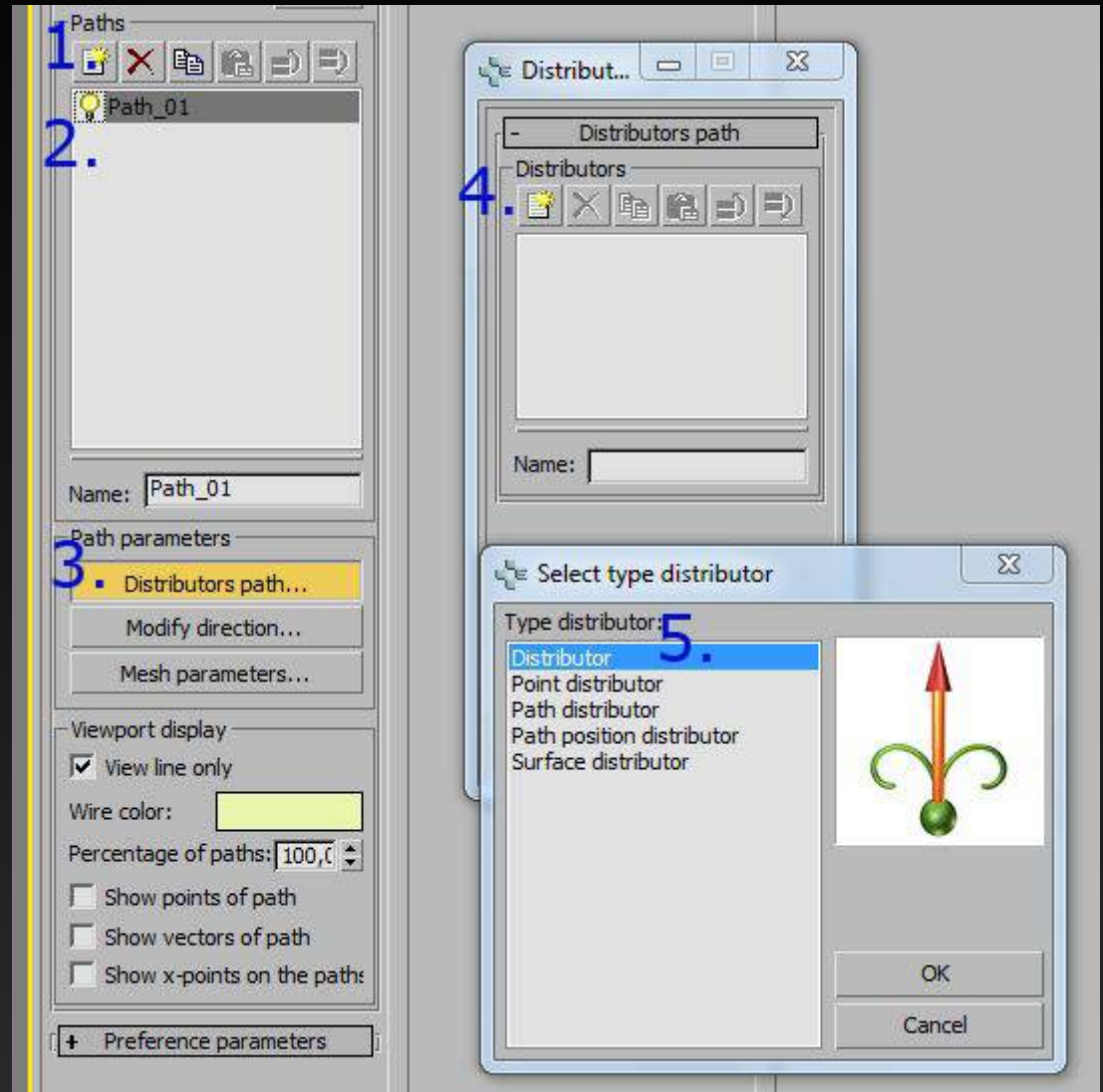
I mostly reset the origin to 0,0,0.



D. Generate the first Element (Path)

Then we create our first path (1.). It is a good idea to rename elements (trunk) (2.) to get organised quickly, since we will jump back and forth a lot. Then we straight open the Root window (called distributor path) (3.). There are no distributors yet defined, so we create one (4.) Here is one great thing about growFx: You have no limitations how many distributors you define, which gives you great flexibility when you finetune your objects. Additionally you can easily turn single distributors on and off for testing purposes.

(5.) Finally we take a simple Distributor which lets our trunk start straight in the Icon centre.

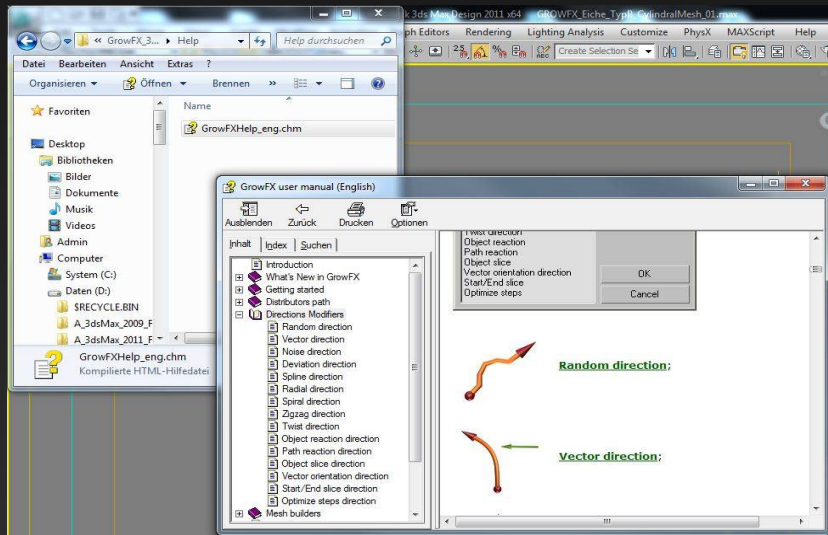
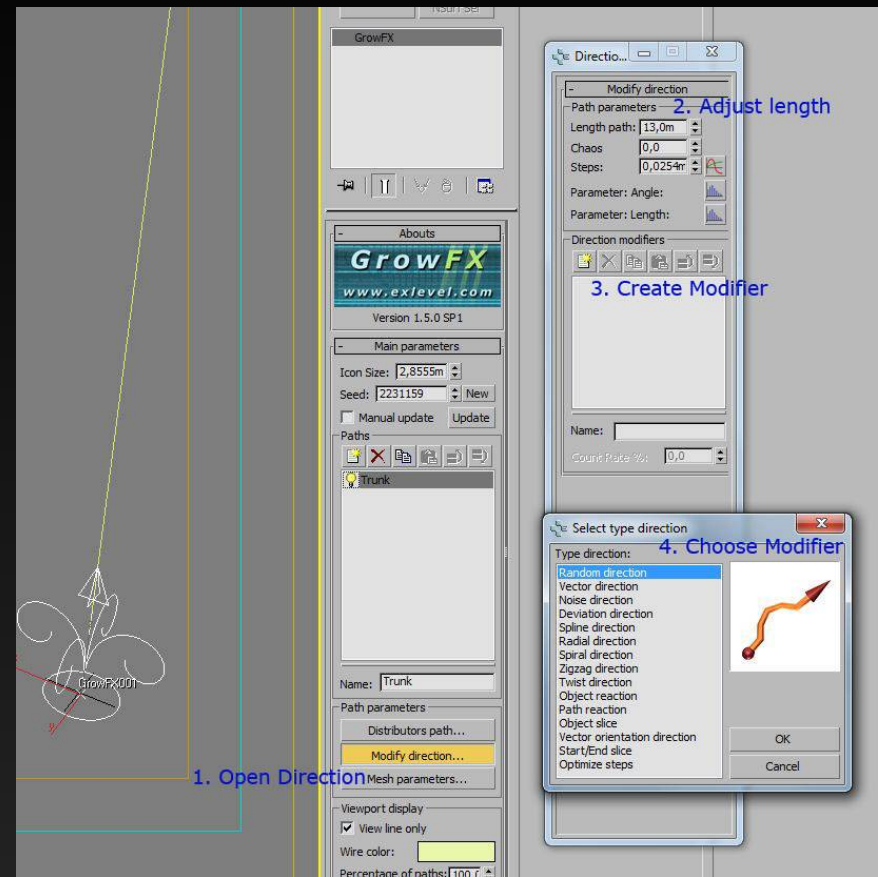


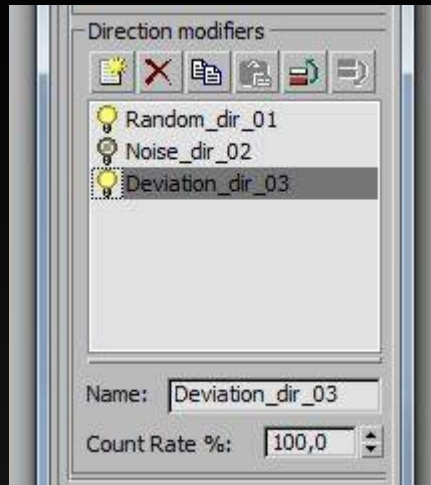
E. Generate modifiers

To influence our paths we open the modify window (1.). In the next step you would adjust the length of the path in total (2.), in this case the trunk). Now we should see a unmodified straight path in the viewport. To influence our path we create our first modifier (3.) and choose a modifier we would like to use. (4.)

Now here we get to a point, where you can make great use of the help documentation, that ships with grow fx. The graphical symbols are pretty self-explanatory and the descriptions give you a good idea what each of the modifiers are doing.

The help-documentation might not be linked into Max (Additional Help), but can be found in the main installation folder quickly.





The choice of modifiers seems overwhelming at first glance, but in fact is well chosen and gives you great control. There is actually no way around testing each one of them by yourself and seeing what they do. One great thing about grow fx is that you can stack modifiers easily and switch them on and off (hit the light bulb), so you can test them separately.

You really need to test the settings yourself and get familiar with each one of them to finetune your growth. The three most important modifiers are random, noise and vector:

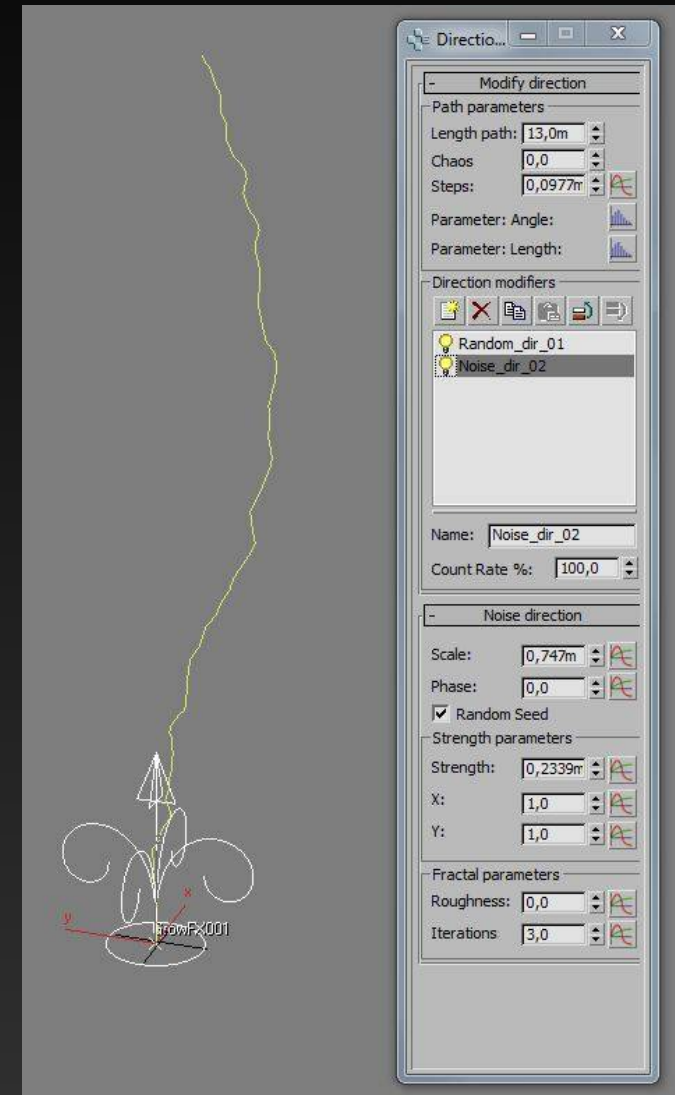
Random: Changes the path direction according to a fractal deviation

Noise: Keeps the general direction but „crumples“ the path underneath

Vector: Bends your vector according to a direction; mostly used for gravitational influences.

For the trunk I used a simple setup consisting of one random modifier and one Noise modifier. You probably need to adjust some scale- and strength values to get some reasonable results.

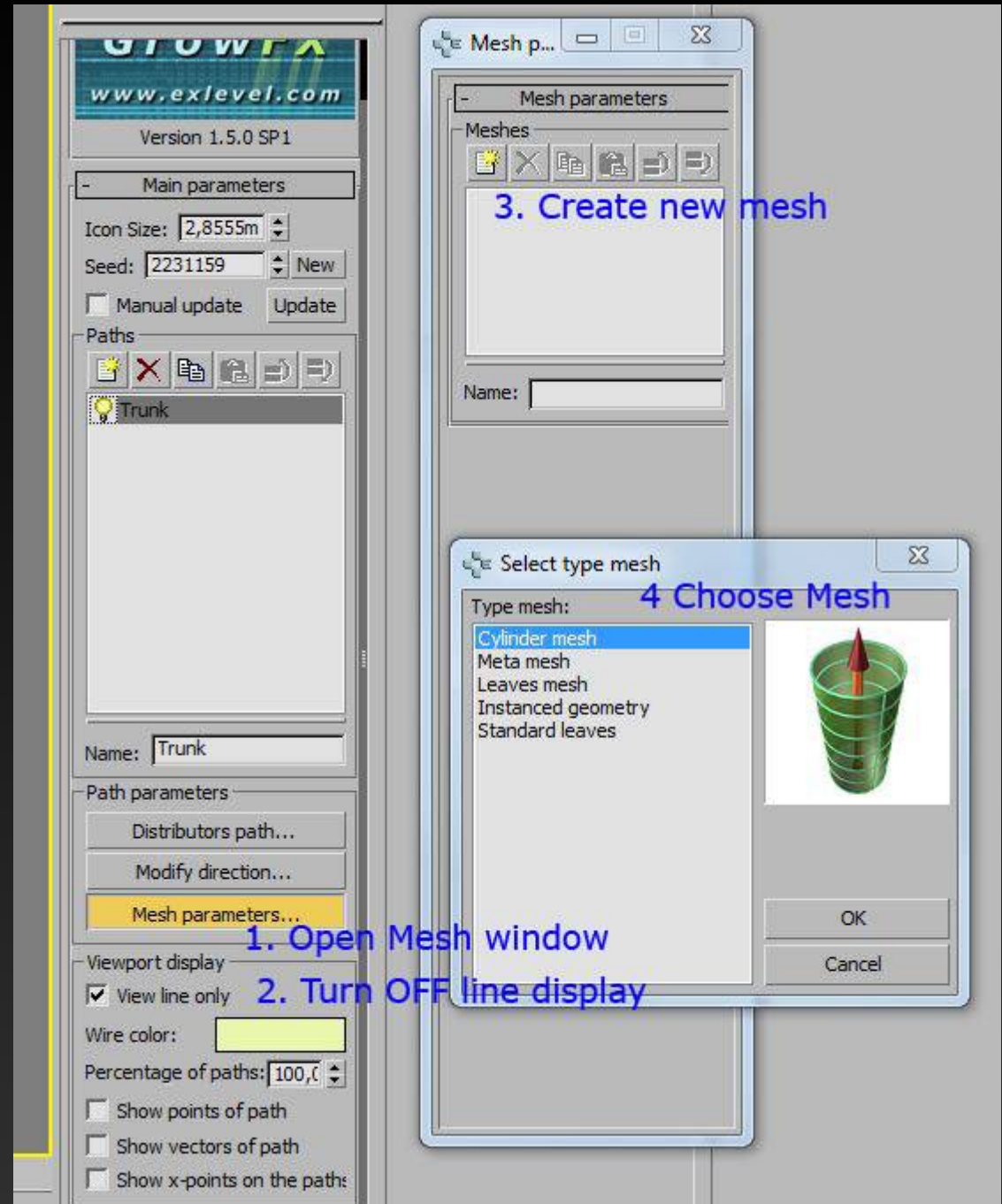
One downside about the user interface at this point is, that you can change values of modifiers that are switched off; you will find yourself fiddling around with some values, seeing no changes and probably messing up a modifier that is switched off. Keep that in mind.



F. Creating first meshes

Now we get to create a renderable representation of our path. (1.) Open the mesh-editor. (2.) Turn line display off, to see what we are doing. (3.) Create a mesh. (4.) Choose a mesh type.

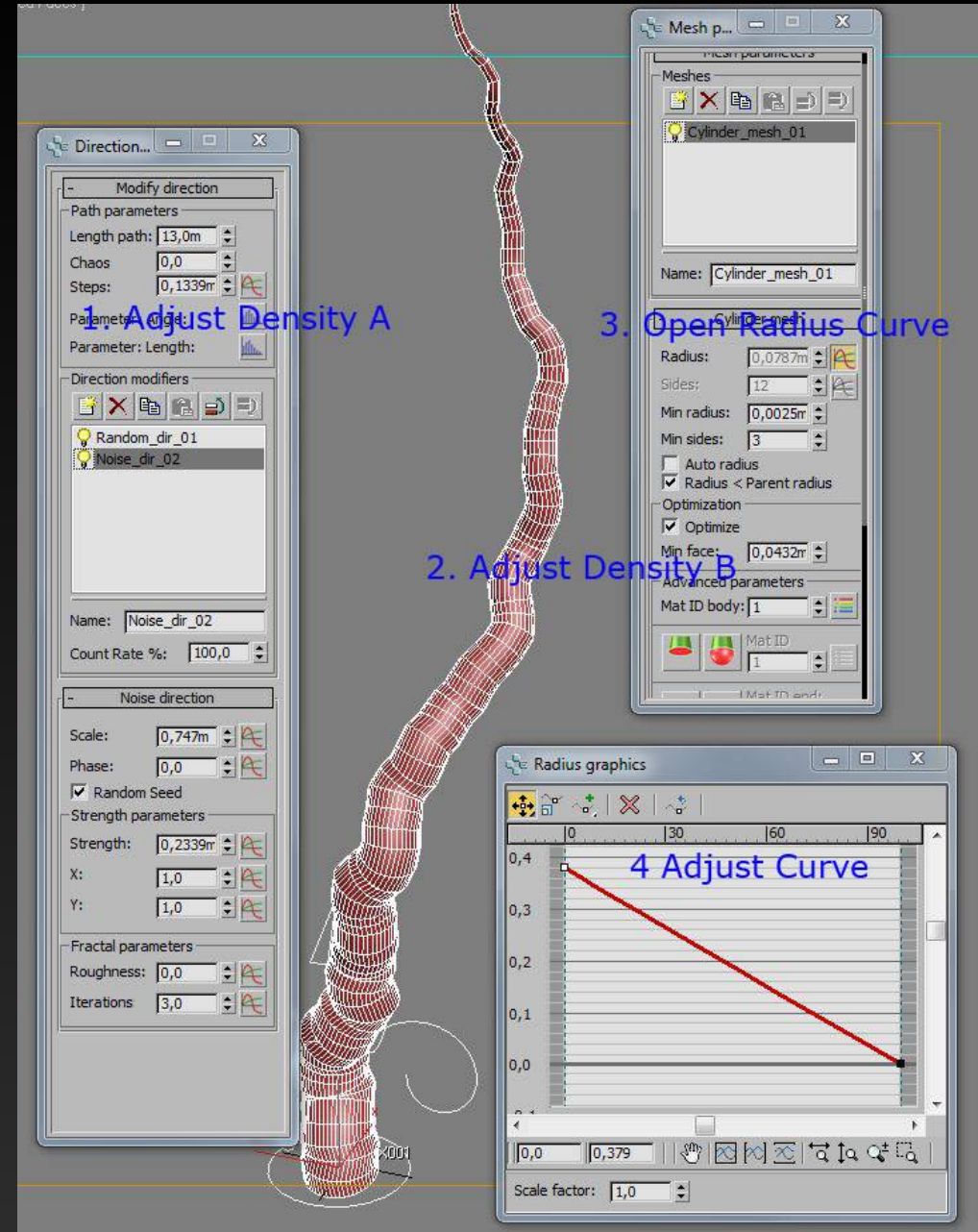
For all kinds of branch-structures it is advisable to start with a simple cylindrical mesh. The meta mesh, does basically the same, but welds branch structures when joining that is quite revolutionary in our opinion, but also takes a lot of time to construct, so we normally keep it until the end. The other meshes are ... other meshes (than branches) ;-)



G. Adjusting your mesh

Now you should see a simple cylindrical shell, which is probably highly tessellated. To adjust the density to a reasonable level you can adjust the shape substeps in the direction window (1.) and the circular density in the minimum faces spinner in the mesh window (2.). To influence the radius in a more natural way we open a curve window by clicking the little curve icon besides the radius spinner (3.). GrowFX uses a lot the internal Curve editor, always adjusting Value over „distance“, in this case „Radius“ over „Growthlength of Trunk“ (4.)

A simple straight line is already not to bad, but adjust to your liking.

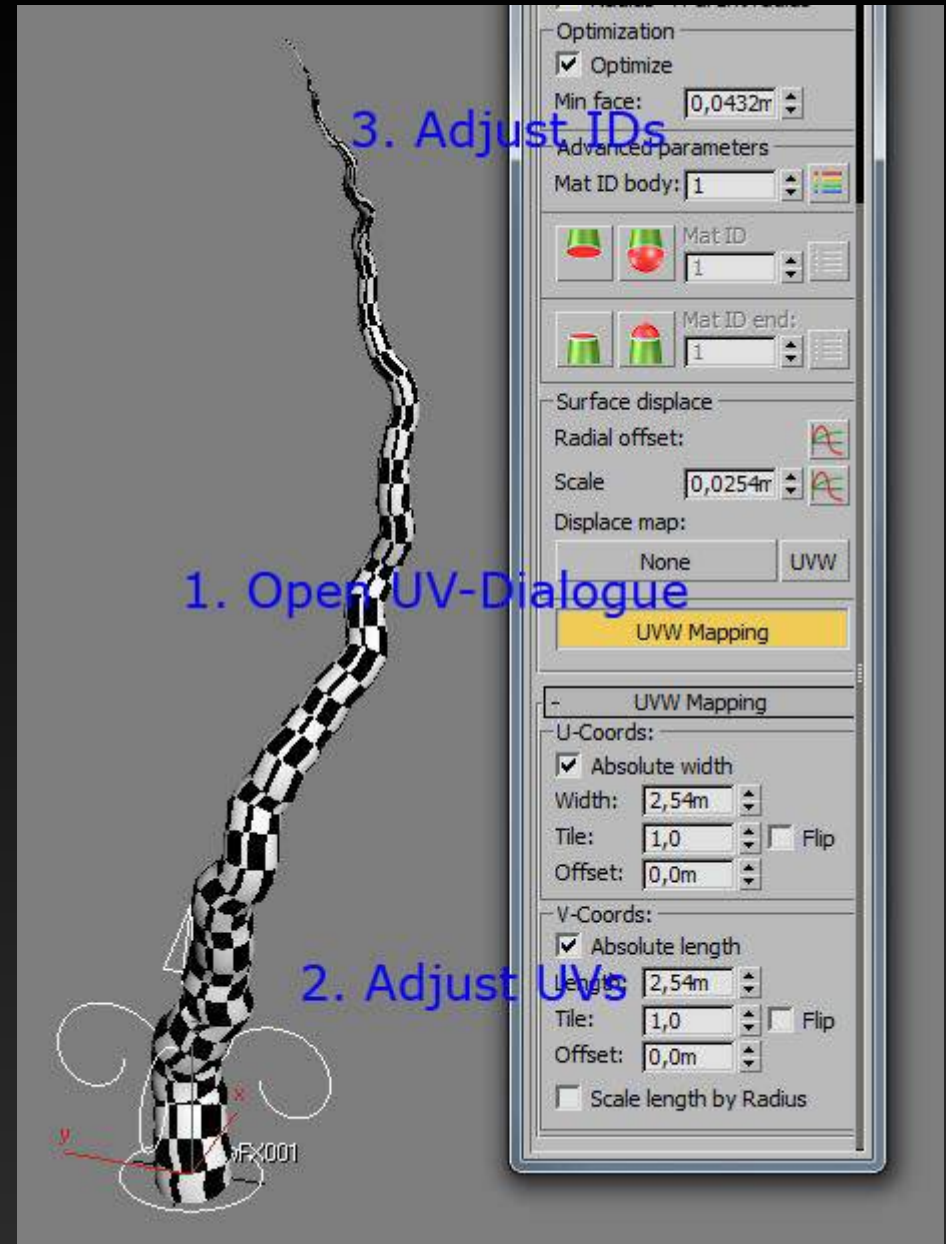


H. Adjusting Uvs

GrowFX has a very functional but efficient way of texture mapping. Scroll down on the mesh editor and hit the broad UVW Mapping button (1.). This should open up a new submenu. I created a quick checkermap-material, assigned it and tested some absolute values (2.). Later you should replace the map with some bark of your choice.

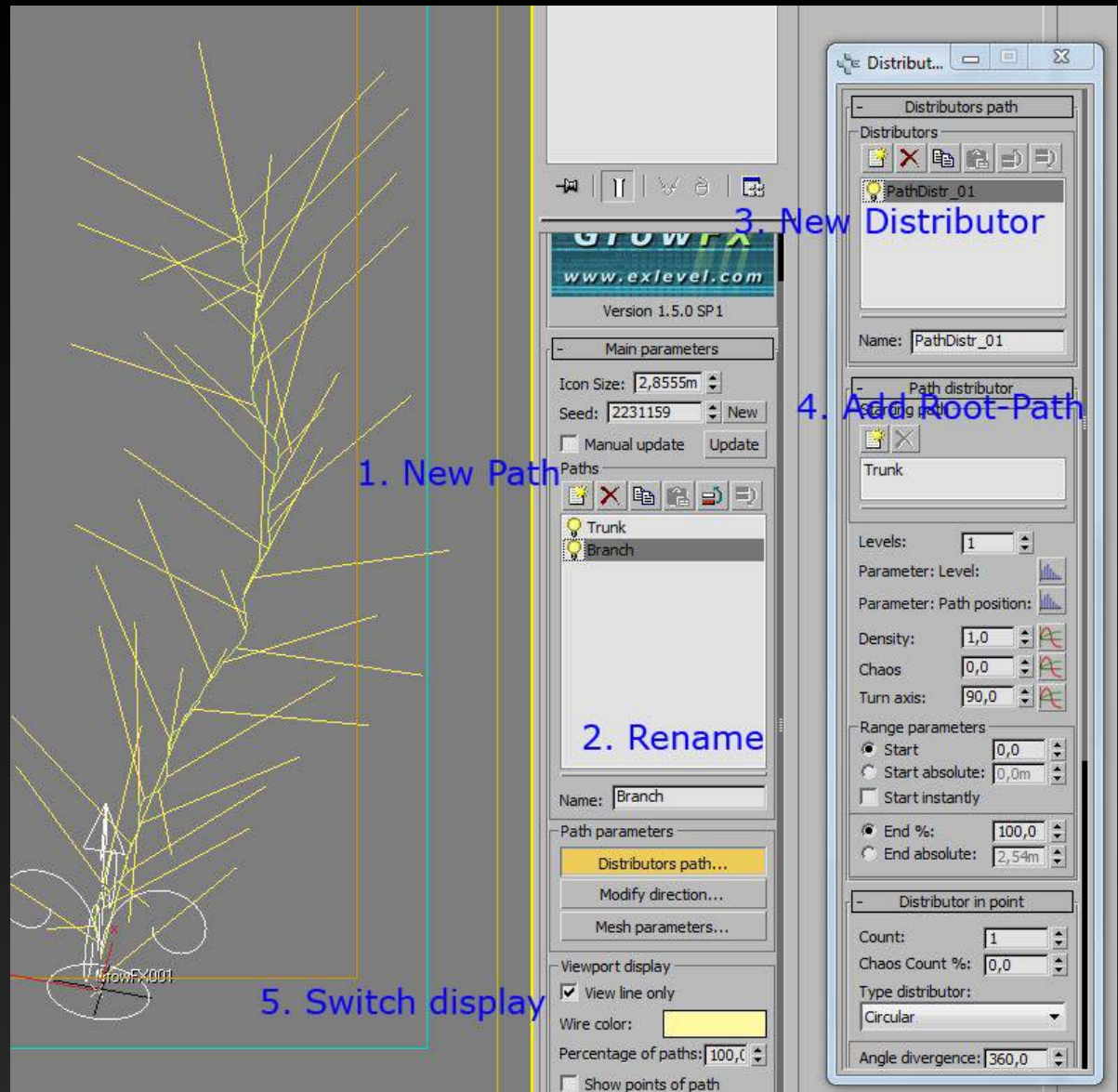
It is generally a good idea to have a slightly larger structure on the main trunk gradually getting smaller moving upward.

For later use with a multimap you can quickly assign the material ID above (3.)



I. Adding elements

Based on our trunk we now start adding additional elements. We create a new path (1.), and rename it to „branches“(2.) . We create a new distributor (3.) and choose a path distributor. This modifier lets us root our branches on other paths. We add a root path and choose our trunk (4.). If you switch to line display (5.), you should now see the newly generated paths.



J. Using Affectors

One important feature in GrowFX are Affectors. These basically represent one value affecting another value in the same hierarchy or deeper. In this case we want, the relational position of the branches to influence the length of itself. In other words, the further up the branches are, the shorter they get.

Make sure your branch-element is selected. Open the Affector control (1.) based on the branch position. Add a new Affector (2.), and pick the Branch - Length, you want to control. Add a curve representation (3.) and edit the curve your liking (4.). See the first results in the viewport representation here.

The image displays the GrowFX software interface with four numbered annotations:

- 1. Open Affector:** Points to the 'Affects path' panel where 'Branch->Length' is selected.
- 2. Add Affector:** Points to the 'Affect factor' section, showing the formula $X = P * \text{Factor} * \text{Chaos}$ and the 'Factor' set to 1,0.
- 3. Add Curve:** Points to the 'Affect factor graphics' window, which shows a red parabolic curve on a graph with X-axis from 0 to 90 and Y-axis from 0 to 1.
- 4. Edit Curve:** Points to the 'Affect factor graphics' window, showing the curve being edited.

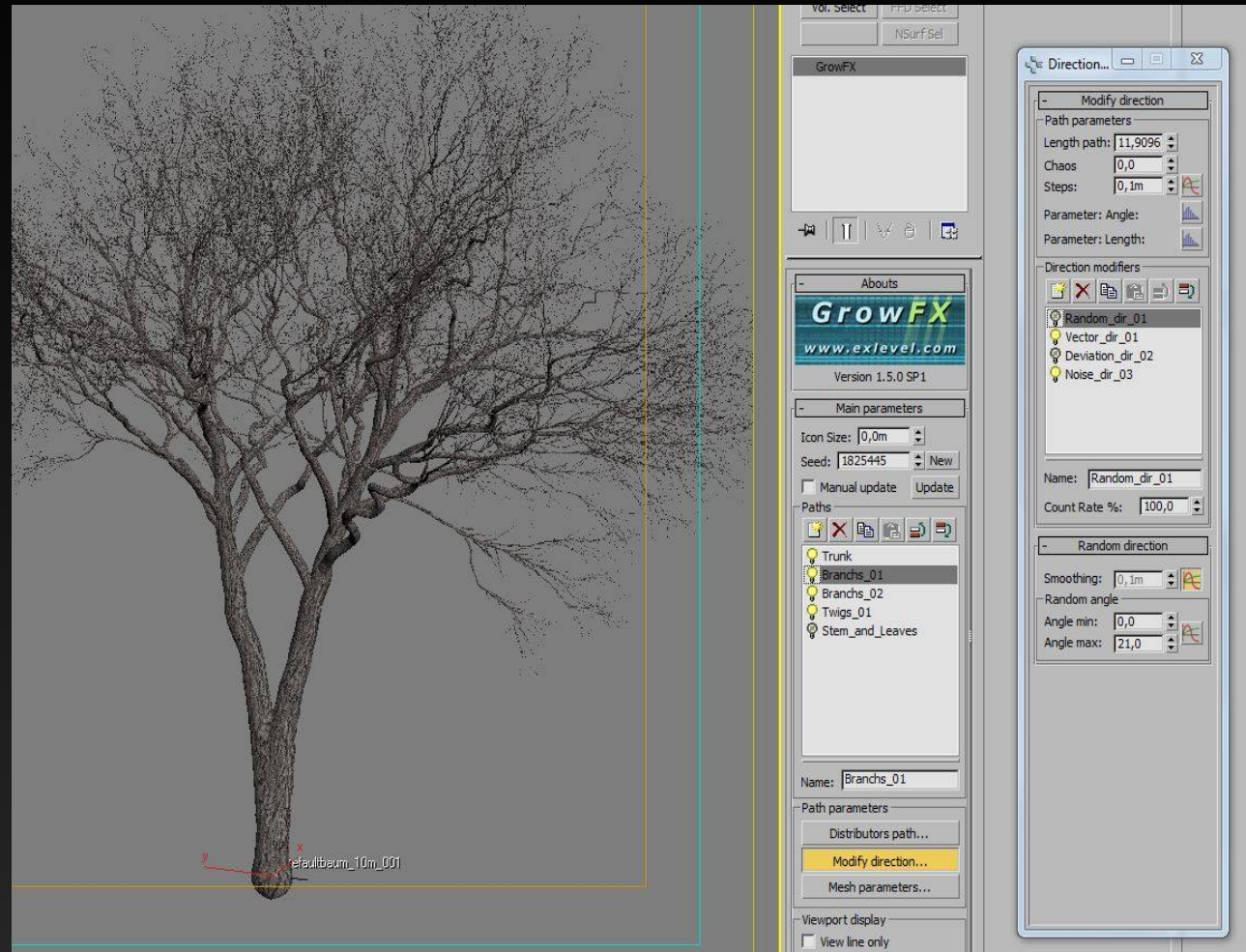
The viewport at the bottom shows a 3D tree model with a yellow branch highlighted, illustrating the effect of the Affector.

K. Move on by yourself!

From here you know all the parameters, you need to move on.

It would take 50 extra pages to illustrate all the finetuning details here, but I will not go any deeper into the process. Furthermore you really understand the settings only if you explore them yourself.

Be sure to doublecheck your reference-images a lot, to make sure you stay on track. Sometimes it is better to switch back to your root elements and turn all other elements off again. The nice thing about GrowFX is, you can easily test settings and compare them by switching controllers off and on.



Here you see the full branching, consisting of 4 elements (leaves turned off. To the very right you can see, that we tested various configurations and kept some turned off.

You have lots of control how your branching works. We kept the start value in general relatively high, to get a tree that is light in the inside and has fluffy balls of leaves towards the edges.

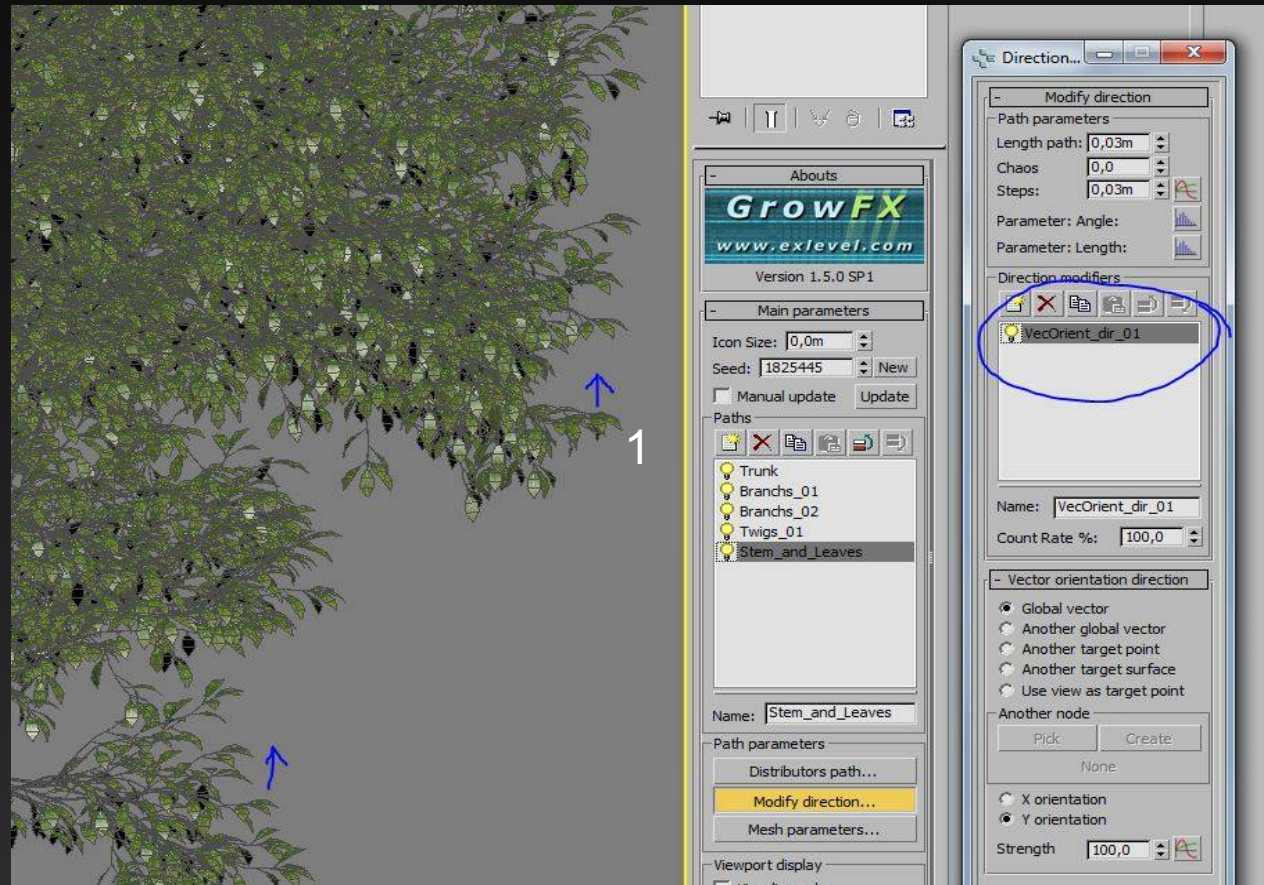
L. Custom Leaves



One important modifier for the use with leaves is the vector-Orient-Modifier which guarantees that your leaves all face upwards. Maybe you need to turn your Base-instance-Mesh on a subobject level, if they all look sideways instead.

GrowFX offers various Leaf-Meshes which are ok. But since we want loads of leaves they will definitely make the biggest part of the Mesh-Load generated. So we rather generated our own optimized mesh, which uses exactly as many triangles as we thought are required.

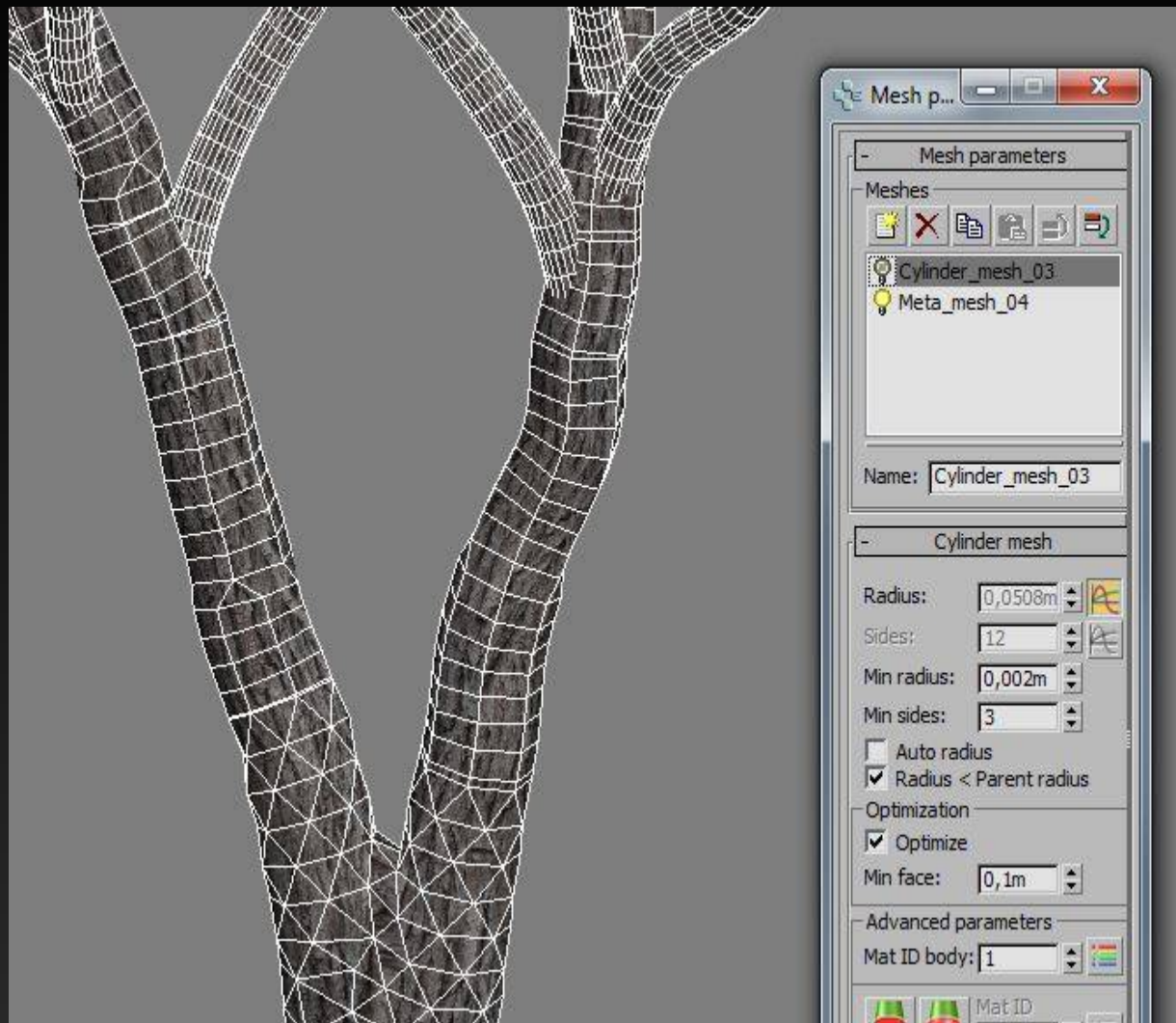
Since Xoio-Studio mostly relies on Vray and occasionally on MentalRay Alphas are an issue. We always try to keep rendertimes as small as possible and Alpha-Maps can be a killer in both engines. Therefore we build the depicted customleave which includes the stem and forms a fairly nice round leaf-shape with 7 tris / 3.5 quads. It is slightly bent to give nice reflections.



M. Hint 1: Metameshes and Cylindrical Meshes

In this image you can see the metamesh (trunk) versus the standard cylinder (bough). While the metamesh nicely connects the elements, the cylindrical elements stick in each other, but calculates way faster..

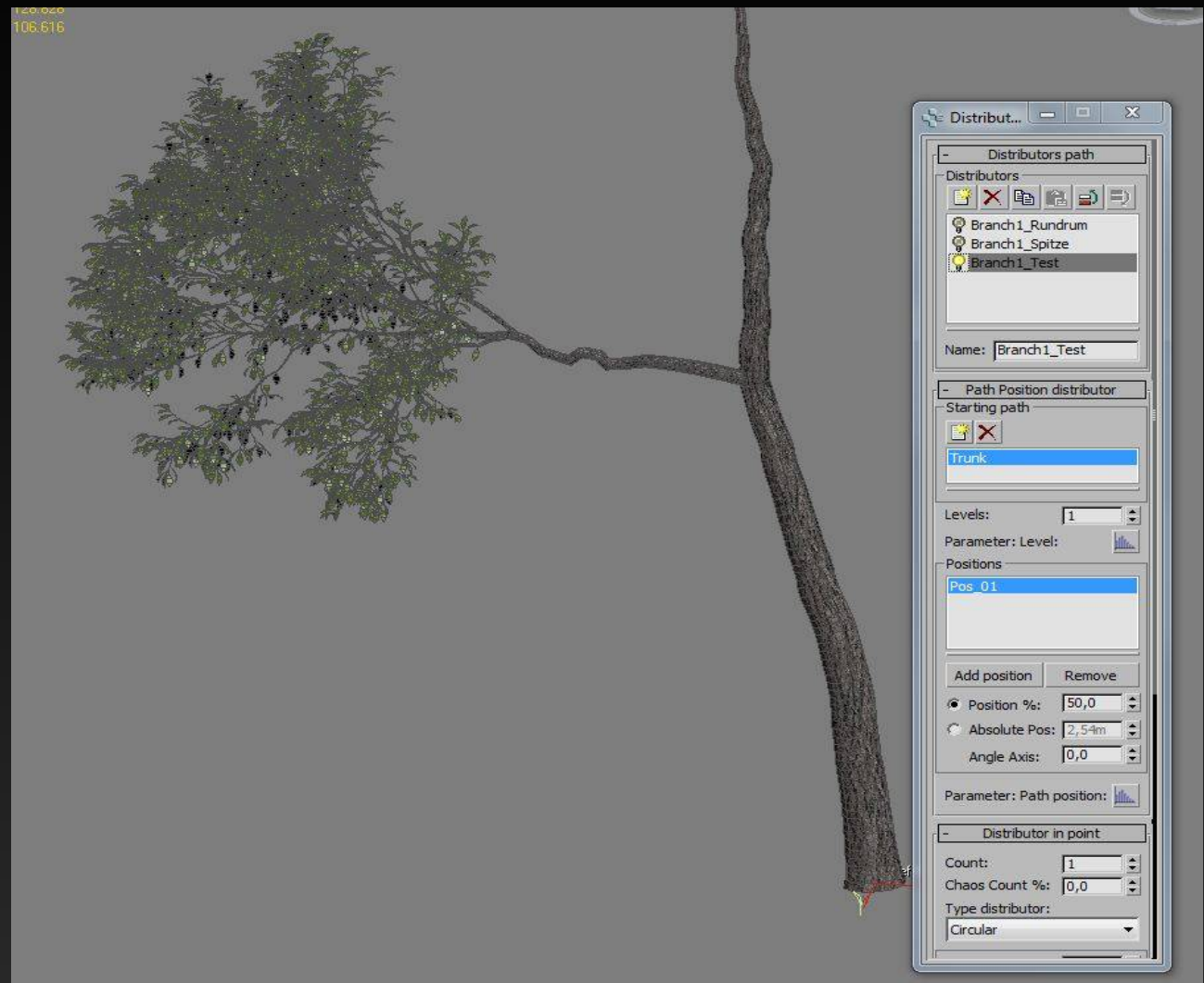
Since Meta-Mesh-Construction can take a lot of time, we keep always two Mesh constructors in the stack, one cylindrical and one Meta-constructor. Like this you can easily switch between both for speed purposes. For the smaller branches Meta-Meshes are generally not necessary, since the connections are not really distinguishable.



M. Hint 2: Testbranches

It can become quite tiresome if you need to judge the branching in a later stage and rebuilding starts to take ages.

To speed up workflow, we inserted a new distributor, which generates only one branch. Turning the other branches off, rebuilding gets way faster. So you can judge all your branching settings quickly and always switch back to get an idea of the whole tree.



M. Hint 3: Do not be afraid.

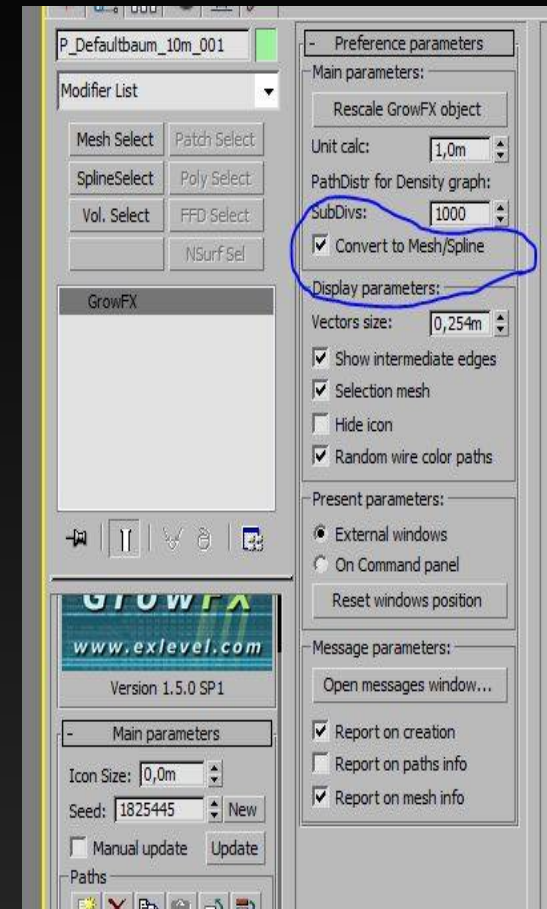
Do not be afraid of a LOT of geometry. Of course things tend to get really slow in the end, but if you really want some realistic tree, you will end up with several hundred thousand triangles. The tree here has 1.3 Million polygons.

You always could get less geometry by using alpha maps for the trees. But since we had experiences, where alphas just generated incredible rendertimes, it seems more reasonable to use highpoly alternatives.



M. Hint 4: Final Export

When you are finally done with your tree, you probably want to make a BackUp-Copy of your GrowFx Object before processing. Then you need to check the „Convert to Mesh“ Button. Only then can you turn it into a regular mesh object. Convert it into an Editable Mesh. Probably you want to turn it quickly into a Vray- or Mental Ray Proxy to improve the memory load.



Here you see some final render of the oak inmax with vray, using a proxy.

I hope this couple of hints can be of any help in your exterior projects.

Thanks for reading, Peter from Xoio