
Xfrog 5
for CINEMA 4D
Manual
Xfrog
inc

Reference Manual
Xfrog 5 Plugin for Cinema 4D

Development Team

Timm Dapper	Software
Bernd Lintermann	Software
Jan Walter Schliep	Tutorials
Andreas Kratky	Reference Manual
Orio Menoni	XfrogPlants Basic Library
Stewart McSherry	Project Manager

Usage

Xfrog 5 Plugin for Cinema 4D is a copyrighted work available as a 30 day demo and as a for-sale licensed permanent version. Please note any intended use for any commercial purpose whatsoever, or any use beyond trial of the 30 day demo requires purchase of this software plugin.

Liability

We hold no liability in the use or potential misuse of this software beyond prompt replacement of any faulty CDROM or other faulty supplied material by us to you.

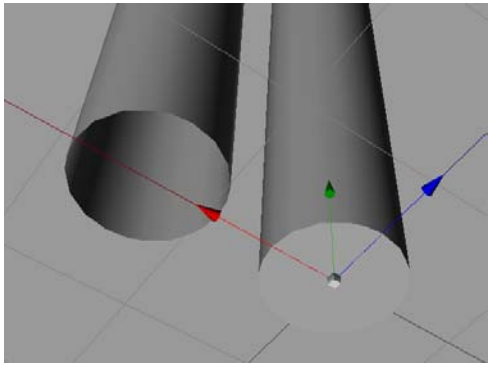
Copyright Notice

Xfrog 4.2.2 for CINEMA 4D
Reference Manual Addendum

Table of Contents

1. Branch Caps

Branch Objects can now have Caps, very much like the SweepNURBS Object. This is useful when a Solid Object is needed and when the Branch Object does not end with a Thickness of zero.

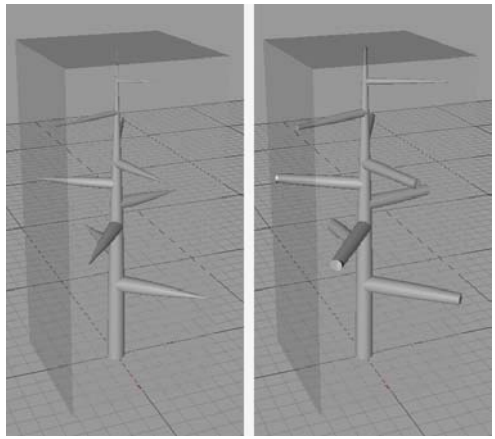


Similarly to the SweepNURBS Object, the Branch Caps have invisible Selections, called C1 and C2 so you can apply a different Material to a Branch Cap.

2. Branch Pruning

A Pruning Object can now be applied to all Branch Objects. The Branch stops growing at the point where the Branch Path intersects the Surface of the Pruning Object.

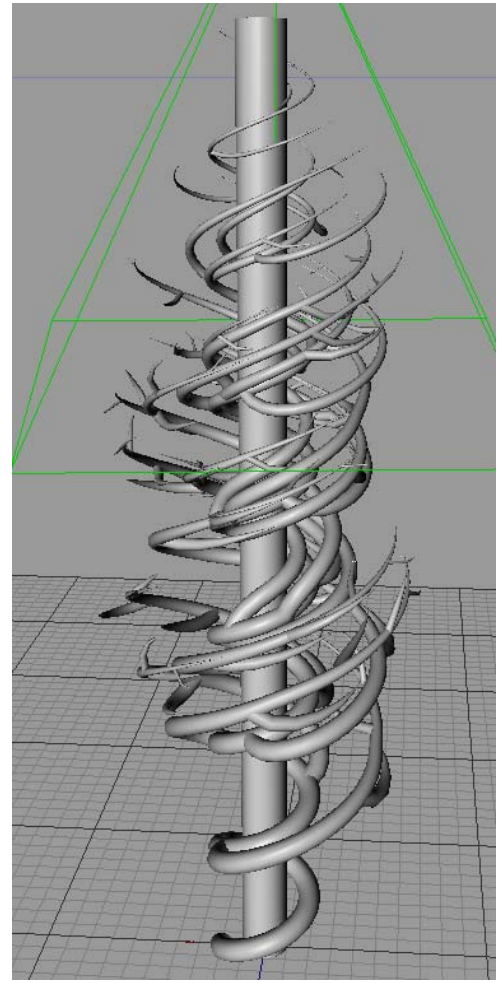
There are two Prune Modes available. With Prune Mode *Limit Growth* the Branch internally clips the Growth value so that the Branch stops growing exactly at the point where it hits the Prune Object with all implications of the Growth Parameter. Prune Mode *Cut Geometry* just cuts the Branch where it hits the Prune Object.



This is a good place to use Branch Caps with a different Material applied, to produce the look of a Branch being cut away.

3. Spline Tropism

The Tropism Object has a new Tropism Type, called *Align To Spline*. When a Spline Object is supplied as Reference, it is used to control the direction of the Tropism.

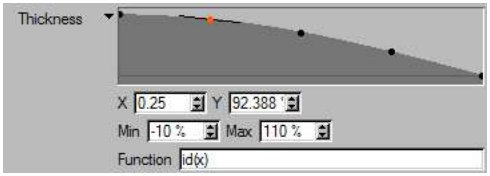


Xfrog 4.1 for CINEMA 4D
Reference Manual Addendum

Table of Contents

1. The new Curve Control

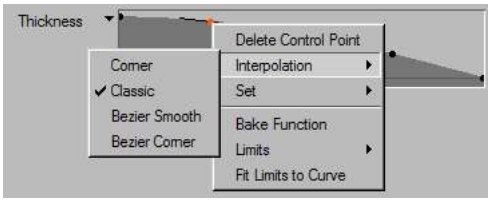
There have been several enhancements to the Curve Control Parameters. They have also been enlarged for easier viewing and editing.



You can now unfold fields for direct numerical editing of the currently selected Control Point. To do so, click on the small arrow in the upper left of the Parameter Curve. To fold it back, just click on the arrow again.

The unfolded Input fields show the currently applied Function, Min and Max values as well as the position of the currently selected point. All of these can be directly set. This makes editing of Parameter Curves more powerful and precise.

The Interpolation Type is no longer the same for the whole Curve. Interpolation types can new be different for every Control Vertex. Also two new Interpolation types have been added. To change the Interpolation Type of a Vertex, right click on it, move to the Interpolation Popup and choose the new Interpolation type. Corner and Classic represent the existing Interpolation types.



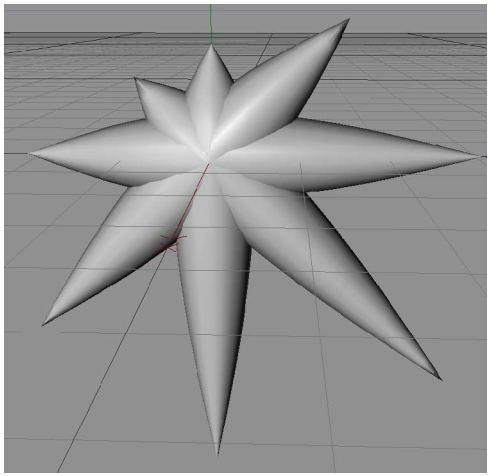
The two Bezier modes additionally give you Control over the Tangents of the Parameter Curve at the Vertices. Bezier Smooth automatically ensures that the Tangents line up so the Curve is smooth at that Control Vertex. Bezier Corner lets you control the In and Out Tangents separately. This allows you to create sharp corners with custom Tangents.

2. New Growth Controls for Hydra and Phyllotaxis Objects

Hydra and Phyllotaxis Objects have now provide more control over subsequent Branch Objects, much like Branch Objects are already able to control the Growth of subsequent Branch Objects.

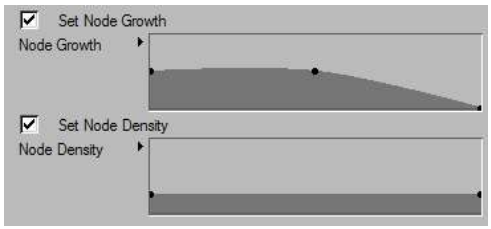
If you check "Set Node Growth" and set the Node Growth to something smaller than 100%, any Branch Objects applied to the Hydra/Phyllotaxis Object as child will act accordingly.

See below an example Hierarchy and the effect the above Node Growth setting has on this Hierarchy.



3. Optional Growth and Density Inheritance in Branch Objects

In Xfrog 4 every Branch Object always passed Node Growth and Node Density to subsequent Branch Objects. This behaviour can now be disabled by unchecking the relevant Checkboxes “Set Node Growth” and “Set Node Density”.



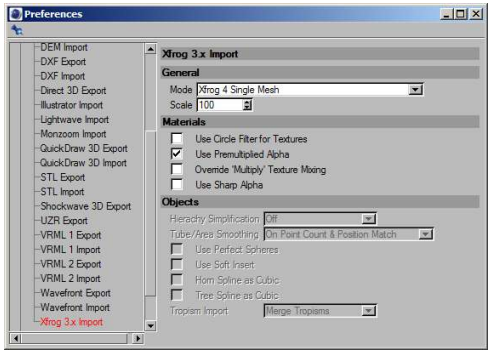
4. New Object Evaluation Mode for faster Redraw

The Object generation process has been optimized for better redraw speed. This is mostly done by internally merging Objects together, instead of handling every Object separately. This new evaluation mode is switched on by default, but in some cases you may want to swith it off. Cases in which this is important can be when you want to make a Branch with its Subbranches editable to edit every Subbranch separately or if you want to multiply non geometry objects (e.g. Light Sources) with a Branch or Phyllotaxis Object.

To switch this new internal Object merging off, check the “Single Objects” Checkbox.

5. Improved XFR Import

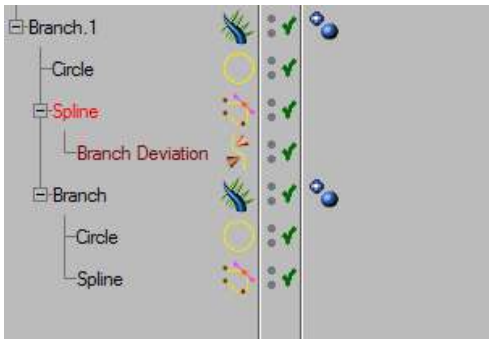
The XFR Import has been improved. Objects created with Xfrog 3.5 or earlier are now more reliably converted to Xfrog 4 Objects. Also a new Import Mode has been added that reads XFR files into a Polygon Mesh, even if the conversion into Xfrog 4 Objects is problematic.



You can switch between Conversion to Xfrog 4 Objects, the new Polygon Importer and (if installed) the Classic Xfrog Import Plugin on the Import Preferences Panel.

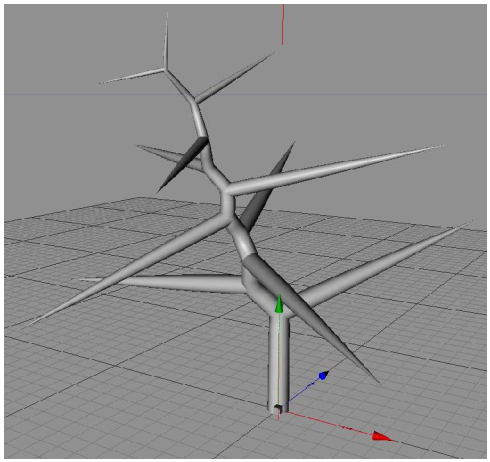
6. Deviation Object

The new Branch Deviation Object can be used to naturally bend Objects at the places where Subbranches emerge. This effect is commonly found in Nature.



The Branch Deviation Object only works when applied to the Path Spline of a Branch Object. It checks if two levels up the Hiararchy there is a Branch Object and if this is the case, uses the Node Distribution Parameter of that Branch to determine where to bend the Spline.

The example above shows how to use the Branch Deviation Object correctly. The angle the Path Spline gets bent at each Node can be defined with a Parameter Curve. The example below shows the effect of setting up a Hierarchy like the one shown above.



7. Xfrog Toolbar changes

The Xfrog Toolbar has undergone some changes. This will cause some Objects to dissapear from the Toolbar created with Xfrog 4. It is recommended to remove existing Xfrog 4 Toolbars and create a new one with Xfrog 4.1.

Table of Contents

1. Introduction

Welcome to the Xfrog 5 Plugin for Cinema 4D.

The Xfrog 5 Plugin adds new Objects to Cinema 4D which allow you to model and animate organic structures and processes, for example, branching(most plants branch), phyllotaxis(i.e. center spiral of a sunflower), gravitropism(plants are usually pulled downward towards Earth), phototropism(plants usually follow and grow towards Sunlight). You will discover, that,we have studied these Natural processes and implemented our knowledge of them inside Xfrog 5.

Xfrog 5 works like this - you to connect Objects together, and the Objects influence each other (Branch components connected together and parametrically edited by you, influence each other to create whatever type of Tree you desire to create). Some Xfrog 5 Objects create geometry, others multiply geometry in ways that mimic Nature. (Phyllotaxis, for example, can multiply Objects across/over the surface of an arbitrary shape based on the golden section, this is found in Nature as seed spirals, i.e. center of a Sunflower, for example)

Additionally all parameters in Xfrog 5 can be animated, and all Xfrog 5 Objects can be combined with all Objects in Cinema 4D, so you can create Trees or other plants which grow, and follow and/or respect Sun and gravity, leaves and branches which move in the wind, biological reactions taking place over time, architectural models which change over time, a large myriad of special effects possibilities, etc, all these things and more are possible now by combining the features of Xfrog and Cinema 4D.

In case you were wondering about the name 'Xfrog' itself, it is an acronym. XFROG, 'X-windows based Finite Recursive Object Generator'. Xfrog originated on SGI IRIX, and many (real-time CAVE for example) artistic performances of Xfrog have been achieved under IRIX.

As a side note, Xfrog is not related to fractals, or to strict mathematical approaches such as 'L-systems', and it is not a 3D modeling program like Maya(tm) or Cinema 4D(tm). Xfrog is a unique approach to solving representation of Nature processes in computer graphics.

Since you have now an understanding of what Xfrog is designed to do, and how it fits into your kit of tools, we would like to give you a little idea of what the existing customers of Xfrog are currently using it for. Here are a few common uses:

creation of trees, flowers, many other types of vegetation. (we ourselves have released 1000 models in 17 XfrogPlants libraries).

combinations of plant models to create ultra-realistic and also many fanciful landscapes

creation of unlimited unique special effects for film and video. (Disney Imagineering, ILM, Dreamworks, Sony Imageworks, Pixar, all have Xfrog and chances are you have or you will soon see models and effects recognizable as Xfrog.)

construction of experimental architectural models: iterative, organic, evolving over and thru time and space. (e.g. see the film, 'Muraya')

construction of micro-organism structural and locomotion studies, for research and educational microbiology.

creation of artwork and animations that seem to resembling living things, because of some mathematical basis but are not strict mathematical animations. e.g. see the tutorial anima-

tions on our website, the abstract animation studies built with custom functions. .

1.1 What's New

The What´s new section is intended for users of Xfrog 3.5 and earlier releases, as a list of the new features available in Xfrog 5, and for the existing users of Cinema 4D, to see what is added by installation of the Xfrog 5 plugin. The features here are simply a list of what is available and more information follows inside this reference manual and also inside the Cinema 4D reference manual.

1.1.1 Cinema 4D features include

- PC and Mac availability
- Customizable user interface
- Multiple viewports
- Multiple cameras
- Level Of Detail
- Customizable toolbars and shortcuts
- Software and hardware shading
- Additional export formats
- Display measurement units
- Library browser
- Additional primitives
- Interactive editing
- HyperNURBS modeling
- Complex lighting
- Multiple spline types
- Text tools
- Boolean Operations
- Metaballs
- Particle Systems
- Deformers
- Bones
- Floor, Sky and Environment Objects
- Audio
- Elaborate transformation tools
- Inverse and multi target kinematics
- Polygonal modeling
- Many Plugins available
- Rendering including raytracing
- Advanced materials and shaders
- Extended animation features

many additional features are available with additional modules from Maxon, such as Radiosity and HDRI (Advanced Render Module) and complex particle system control (Thinking Particles Module)

1.1.2 Xfrog 5 Improvements vs. Xfrog 3.5

- Time units - mins, secs, frames
- Multi-track animation
- Soft Insert animation feature; removes 'popping'
- Phyllotaxis (Phiball) can place objects on any Surface Of Revolution
- Phyllotaxis now allows direct control of the orientation of multiplied objects
- Branch Object combines the power of the Tree, Leaf and Horn Components
- Branch Object, mesh resolution is independent from number of branches
- Branch Object can now generate several branches per node
- Curve Control now uses spline curves to control parameters and allows for much

better control
- Any Cinema 4D spline can control Xfrog Objects even text splines
- Curvature Spline offers controls similar to the Xfrog 3.5 Horn Component, while offering more options
- Hydra offers more control
- New Variation Object allows exceptions in the iterations of multiplier Objects
- Level Of Detail offers more control

1.2 Getting Started

Your need to extract the contents of the Xfrog 5 .zip (PC) or Xfrog 5.sit (Mac) file into your Cinema 4D Plugins folder, to use Xfrog 5.

We have prepared a separate document, which is supplied in your Xfrog 5 package called 'Installing and Licensing Xfrog 5' Please use this document now to install and license Xfrog 5, if you have not already done so.

1.2.1 Where to find Xfrog

Xfrog 5 Plugin seamlessly integrates into Cinema 4D. You will find elements of it in the following places:

- Parameters for the import of models from previous Xfrog versions are located in the Preferences dialogue of the Edit menu in the section "Import/Export".
- All Xfrog 5 Objects are accessible in the Plugins menu in the Xfrog section. A toolbar with these objects can be used.
- The Xfrog 5 Objects are linked to a model hierarchy in the "Object Manager"
- All parameters of Xfrog 5 Objects are accessible in the "Attributes Manager"

1.3 Migrating from Xfrog 3.5 to Xfrog 5 Plugin

Xfrog 5 is a Plugin that integrates seamlessly into Cinema 4D. This was in answer to demand for integration of Xfrog inside popular 3D software programs. We chose to port to Cinema 4D, before other platforms, because of the strong performance / price ratio of the Cinema 4D base product, and the open architecture which allowed easy migration of Xfrog 5 as a plugin inside Cinema 4D. As well we were able to establish a good relationship with Maxon computer and offer sidegrade and upgrade pricing to our existing Xfrog users. This was a compelling reason to work together.

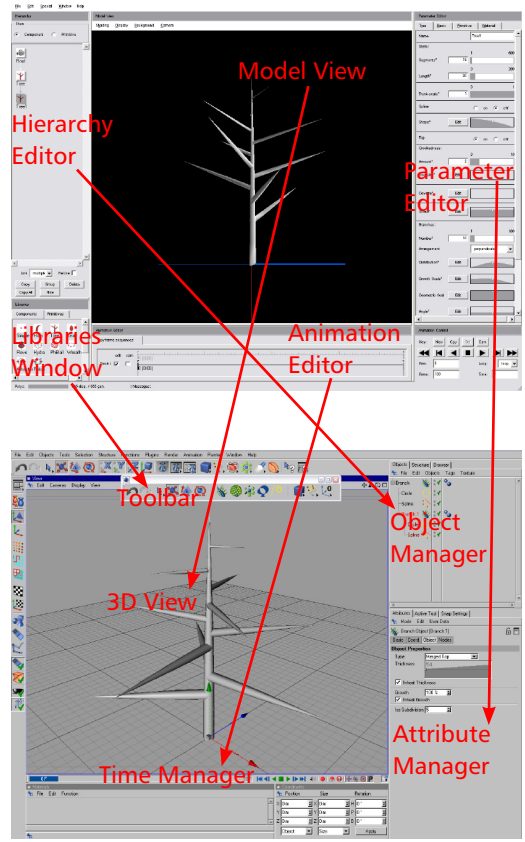
In any event, this new Plugin approach gives the existing Xfrog 3.5 user a powerful new animation and rendering system, as well as many new abilities not possible with Xfrog 3.5. To the existing Cinema 4D user, we bring organic modeling and animation capabilities as new Objects, which are seamlessly integrated into Cinema 4D work.ow.

This new plugin approach has produced several changes in the interface compared to previous versions of Xfrog. To familiarize you with these changes and make the transition easy we provide this Migration Guide. You will probably discover that the fundamental Xfrog work.ow is not greatly changed and you will quickly get used to the new environment. However, the adaptation to both a new Xfrog and possibly a new 3d software package (Cinema 4D) is quite a bit to absorb. But we decided the best possibility is to provide you

with a flexible application that will be able to fulfill your growing demands now and in the future. The new version offers many more features and more flexibility of use. The downside is that at the current time while we are doing our very best to solve all downward-compatibility issues the Importer may have problems with some XfrogPlants models. This is because XfrogPlants models often make use of tricky workarounds because of limitations in former Xfrog versions. We are contining our efforts to improve the XFR Importer, and it's speed. In the meantime we recommend that you use the .c4d files included on all XfrogPlants DVDs and in the XfrogPlants Basic Library of 60 Trees supplied with the Xfrog 5 DVDs.

1.3.1 Interface Changes

The elements of the interface and their functions remain the same, what has changed are the names and their placement. The following images illustrate what is changed, and where, in the Xfrog 5 Plugin.



1.3.2 Object Changes

The Xfrog Components are now known as Cinema 4D Objects, in order to better fit into the Cinema 4D naming convention. The separation of Xfrog Components and Primitives has been removed. They are implemented as Objects in Cinema 4D.

The Tree Component, the Horn Component and the Leaf Component in Xfrog 3.5 had a very similar architecture, so we decided to integrate them into one single Object, the Branch Object. Perhaps you have had a desire to use a function assigned to a slider in the Horn Component, inside the Tree Component - This is now possible. The integration of the three Components into one creates much more flexibility.

The PhiBall Component has become the Phyllotaxis Object. We added some functionality and decided to call the Object by it's botanical name. The Pyllotaxis Object still generates a spatial distribution of Objects according to the golden section but now you have the possibility to use any kind of Surface Of Revolution as basis for the distribution. It is also possible to influence the orientation of the Objects on the Surface of Revolution.

The Hydra Component and Wreath Component are now integrated together as the Hydra Object, which has the functionality of both Components.

The Hyper Patch Component is replaced by the Cinema 4D FFD Object, which actually has the same functionality as the Xfrog Hyper Patch Component.

The Variation Object, which is new to Xfrog 5, allows alternatingly iterations of different geometric Objects with the same multiplier Object. You can have them alternate regularly, randomly or you can assign exceptions for specific positions.

In Cinema 4D a lot of definitions are made with use of splines so we transformed the range definitions that you know from previous Xfrog versions such as the Horn Object curvature into the Curvature Spline Object. This object makes parameters like Rotation and Translation accessible as an independent Object. It can be combined with all Xfrog or Cinema 4D Objects.

Similarly we made Tropisms available as an independent Object, the Tropism Object. It can be combined with any Xfrog or Cinema 4D Object which is defined by a Spline.

Detailed information for all the Xfrog 5 Objects is available in the Object descriptions in this manual. More information about the FFD object can be found in the Cinema 4D documentation.

1.3.3 Modeling Changes

The method of setting up your model hierarchy is basically the same. As you have seen in the overview of the interface changes the linking of Objects (Components) is now carried out in the Cinema 4D Object Manager.

Xfrog Objects are created by selecting the name from the Plugins/Xfrog4 menu or by clicking on the corresponding icon on the Xfrog 5 Toolbar. The new Object appears at the very top of the Object Manager. From here you can drag it and link it to the hierarchy. If the newly created Object generates geometry it will immediately be visible, as if – in previous Xfrog versions – it were linked to the Camera Component.

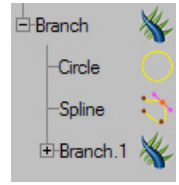
The Xfrog Objects used to multiply geometry have a nested subhierarchy. This subhierarchy is used to define certain parameters: The Objects that are iterated by the Multiplier Object are linked to the Multiplier as a subhierachy. As an example let us look at the Hydra Object multiplying a Cube. The Cube Object is linked as a child to the hydra Object.

The subhierarchy can be opened and collapsed by clicking the plus or minus sign in front of the Object name.

Let us have a look at another example: Linking two Branch Objects together in order to build a tree. The first step is to create two Branch Objects which will appear at the top of the Object Manager. They are both at the same level in the hierarchy and the geometry they generate fall into exactly the same place so that you will only see one trunk. Now click the plus sign in front of the first Branch Object to expand the subhierarchy of this Object. You will see two Spline Objects defining the outline and the curvature of the trunk. At the third position in this subhierarchy you will need to link the second Branch Object to make it a new branching level. To connect an Object to the third position you drag the second Branch Object towards the second position occupied by the Spline. When the Branch Object is floating over the Spline name you will notice the cursor changing into an arrow pointing downwards.

When you move the Branch Object a bit further down below the Spline name the cursor will change to an arrow pointing to the left.

Now release the mouse button and the Branch Object will be linked in the third position creating the branches. The procedure of defining links is a bit different now and you have to try it a few times to get comfortable. By the way: If you released the mouse button when the cursor arrow points downwards you would have linked the Branch Object as a child of the Spline Object which does not create any meaningful output in our example.



For more information on this linking process you may wish to read the Cinema 4D manual Chapter titled 'Object Manager'.

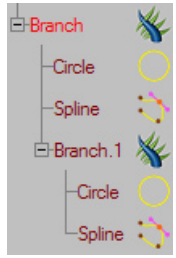
1.3.4 Animation Changes

A lot of changes have occurred in the way you create your animations. Now you can use the elaborate animation tools that come with Cinema 4D to animate your Xfrog models. Every parameter can be animated independently using Multiple tracks, and with the F-Curves (function curves) you can influence the interpolation between different keys. A detailed documentation of how to use F-Curves can be found in the Cinema 4D manual chapter entitled 'F-Curves'. The basic tools to set up an animation are available in the Time Manger below the 3D View.

To give you a brief overview of how to use the animation capabilities we will look at two examples: A growth animation of a tree and an animation of a trees curvature. Based on the previous example in which you linked two Branch Objects we will have a look at the Growth parameter of the first Branch Object, the trunk of your tree.

By default the time slider at the far left of the Time Manager is set to frame number 0. With this slider you can scrub through the animation. The default position will let you access the beginning of the animation.

In order to create an animation of the growth of the tree you need to access the parameters of the first Branch Object (it passes the relevant information through the hierarchy). In the Object Manager select the first Branch Object by clicking on its name.



In the Object Properties (displayed in the Attribute Manager) of the first Branch Object locate the Growth parameter and set the parameter value to 0%. The tree is now scaled down to 0.

Make a right-click on the parameter name. In the menu that pops up, select "Animation/Add Keyframe". This automatically adds the first Branch Object to the list of Objects that are part of the aniamtion and adds a track for the Growth parameter. The first key is set and the parameter value stored in this key.

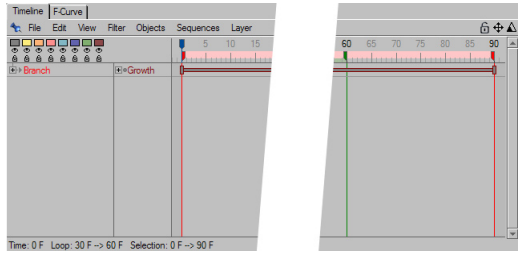
Now move the time slider to the far right of the Time Manager to define the end of the animation.

Set the Growth parameter to 100% and add another keyframe using the same method.

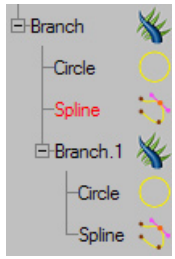
To view the animation you just created you can click the Play button (the green triangle in the Time Manager).

The animation of the curvature works in a very similar way. First make sure that the Timeline is visible. You can either invoke the Timeline Window by selecting "Timeline" from the "Window" menu or by selecting "Animation" from the "Layout" submenu of the "Window" menu.

When you continue with the animation that you just created, you will see the following in the Timeline Window:



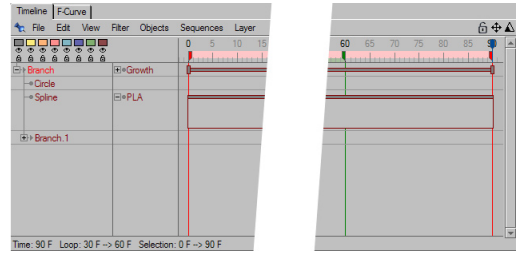
To animate the curvature of the tree you need to select the second position in the subhierarchy of the first Branch Object, which is occupied by a Spline Object defining the curvature of the Branch Object.



For this animation you will have to deal with the control points of the Spline Object. Therefor you need to activate the PLA (Point Level Animation) mode that allows keyframing of the changes made to the control points by clicking the orange button with a key icon (look to the immediate right of the full orange button below) .

Select "Points" from the "Tools" menu to switch to the tool that allows you to edit the control points of the Spline Object. (Note - For easier viewing you can set the display mode of the 3D View Window to "Isoparms" in the Display menu inside the 3D View Window).

In the Timeline Window bring up the "File" menu and select "New Track" and "PLA". This creates a new track to record the control point changes. In order to see this new track in the Timeline Window you can click the plus sign in front of the Branch objects representation.



Move the timeslider to the far left. Hold down the Ctrl key and left click the mouse at the far left of the PLA track to set the first key on this track. Then do the same for the last key of the track. So far no changes have been made to the point structure and nothing changes compared to the previous animation.

Move the timeslider to a frame somewhere in the middle of the animation, e.g. frame 60. As you switched to the Isoparm display mode you can see the three control points of the Spline Object. Select the point in the middle of the Spline Object and drag it to the side. Then again Ctrl left mouse click on the PLA track to create an intermediate key.

When you play the animation now you will see your tree bending sideways while it is growing.

2 Xfrog Reference

2.1 Curve Control

In addition to the control widgets that you know from Cinema 4D (see section 1.6) the Xfrog 5 Plugin uses a widget called curve control. This control provides a curve defining a range of different values. It is used with parameters such as the thickness of a Branch Object (see below).

By default this curve has one point on the left representing the value of the first iteration and a second point on the right representing the value of the last iteration of the Object. The intermediate values are interpolated. New control points can be inserted by right-clicking on the curve at the place where the new control point is to be inserted and selecting "Insert New Control Point" from the popup menu. Right-clicking an existing control point gives you the possibility to delete the point. When you click at points while holding down the shift key you can select several points at the same time, move them and delete them.

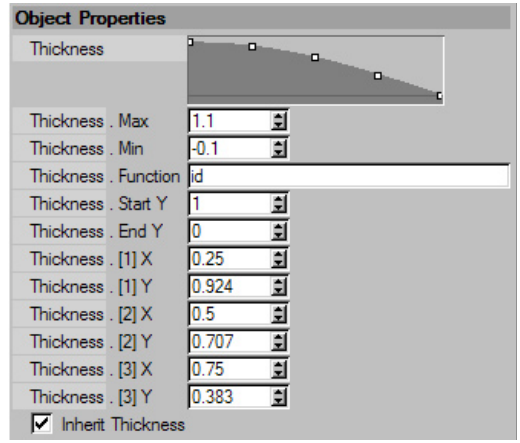
By right-clicking the curve you also have the possibility to bake a function that is assigned to the curve control. (The interpolation of the values between the control points of the curve can be influenced by a mathematical function). A function can be defined in the subchannels (see below). With the command "Bake Function" it is possible to translate the assigned mathematical function into the resulting curve. After this the curve can be edited, new control points can be inserted etc. The possibility of changing the mathematical function itself is not available anymore after baking it.

Right-clicking the curve allows you to switch the interpolation characteristic between the

control points from either a smooth (Classic) to a linear (Linear) interpolation. Finally the right-click options of the curve allow to change the limits of the display of the curve. It is possible to select between fixed limits, limiting the display to the currently visible range; further it is possible to select between modes that adapt the displayed range of the curve according to the max, the min, or both of the curve.

By right-clicking the parameter name you can access the subchannels (Show Subchannels) of the parameter. The subchannels allow access to every parameter numerically (for further info, please see the Cinema 4D manual). In the case of the curve control the subchannels give access to the X/Y coordinates of every control point of the curve as well. They allow you to assign a mathematical function that is used to interpolate the intermediate values between the control points. More information on possible functions that can be used with a curve control is available in chapter 4 "Using Functions".

As an example of how to use the subchannels, the following paragraph shows the Thickness parameter of the Branch Object.



Thickness.Max

By right-clicking the word "Thickness" in the attributes manager and selecting "Show Subchannels" you get access to the numeric values defining the points of the Thickness curve. Thickness.Max lets you define a maximum value for the curve display in the control. The value only influences the display, not the actual model.

Thickness.Min

Again this parameter is only available by right-clicking the word "Thickness" in the attributes manager and selecting "Show Subchannels". Thickness.Min lets you define a minimum value display of the Thickness curve.

Thickness.Function

This subchannel of Thickness is used to assign a function to the Thickness curve that is

used to calculate the scaling factor for each iteration. By default this parameter is set to id (identity) using the Thickness curve as-is to calculate the Branch Objects contour. It is possible to use other mathematical function instead of id to achieve different shapes or effects. For more information on the usage of functions refer to the chapter 4, "Using Functions".

Thickness.Start Y

A Thickness subchannel defining the first point of the Thickness curve. Since this point is the beginning of the curve it only has one coordinate, the Y coordinate.

Thickness.End Y

A Thickness subchannel defining the last point of the Thickness curve. Since this point is the ending of the curve it only has one coordinate, the Y coordinate.

Thickness.[1] X

When the Thickness curve has more control points than the first and the last one this parameter allows you to specify the X coordinate of intermediate points. The intermediate points are numbered according to their position on the curve.

Thickness.[1] Y

This parameter allows you to specify the values for the Y coordinate of intermediate control points of the Thickness curve.

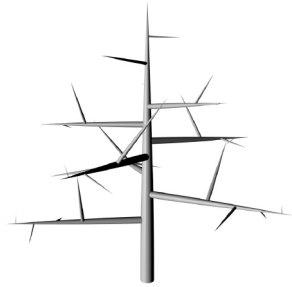
2.2 Xfrog Toolbar

The Xfrog Toolbar gives direct access to the Xfrog Objects. (Plugins -> Xfrog 5 -> Create Toolbar)

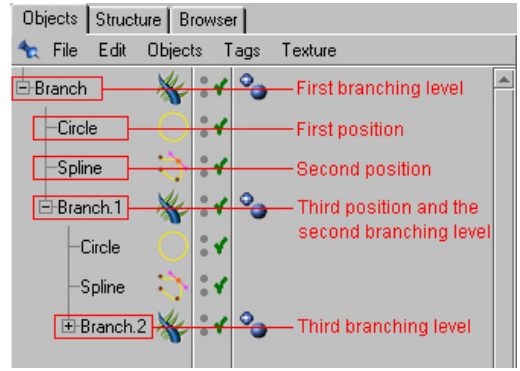
2.3 Branch Object

The Branch Object creates – not really surprisingly – branching structures. One Branch Object on it's own looks in it's default shape like a conical horn that could remind you of a trunk. This shape is achieved through the iteration of segments along a line which are all triangulated together forming a hull. The segments are continuously scaled down until the last segment is ending in a point

Linking several Branch Objects together creates a tree structure with several branching levels. Each Object represents one branching level.



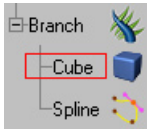
In order to create subsequent branching levels the Branch Objects need to be linked in a specific hierarchy. The Branch Object expects a number of different Objects linked to it, similar to the C4D Sweep NURBS Object. The different positions in the subhierarchy are reserved in the following way:



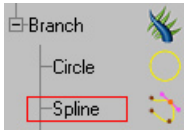
The first position in the subhierarchy contains the Object that is multiplied by the Branch Object and is arranged in the characteristic conical shape. This position can be occupied by different Objects. The default Object in the first position is a Circle that defines the profile of a tubular hull forming the trunk or the branches. The shape of the hull is controlled by an algorithm as explained in the section "Object Properties".

The Circle can be made editable by selecting "Make Editable" from the Edit menu in order to deform the profile.

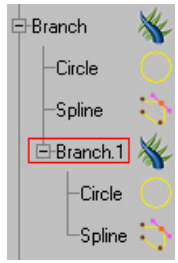
By inserting other Objects in the first position of the Branch Object subhierarchy it is possible to multiply cubes or any given other type of geometry along the Branch Object. The amount of multiplied instances can be controlled through the Number parameter of the spline in the second position of the Branch Objects subhierarchy. This parameter defines the amount of intermediate points for this spline.

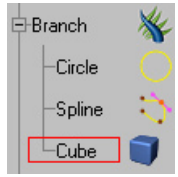


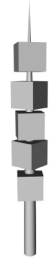
The second position in the Branch Object's subhierarchy contains a Cubic Spline that defines the curvature of the trunk or the branches. By editing the control points of this Spline the curvature of the Branch Object can be influenced



The third position in the Branch Object subhierarchy contains the Object that is multiplied along the trunk or branch. New instances of this Object are generated at the branching nodes along the Branch Object. Connecting another Branch Object creates a new branching level with smaller branches coming out of the trunk or the branches of the previous branching level. Other objects like a Cube connected to the third position would create cubes instead of branches:

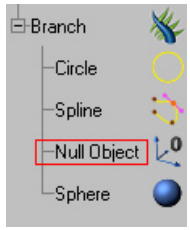


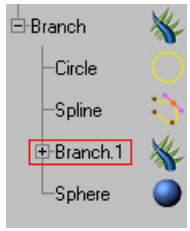


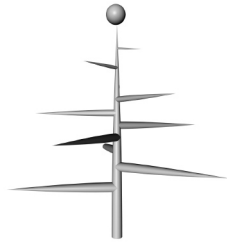


The fourth position in the Branch Object subhierarchy is reserved for the Object that is linked to the end of the Branch Object. This creates for example a possibility to put a leaf at the end of a branch, or using the Branch Object to create a stem of a flower, this would be the place where the blossom needs to be linked to appear at the end of the stem.

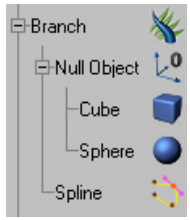
In certain cases you might only want to create an Object at the end of the Branch Object. In this case you can link a single Null Object in the third place of the Branch Objects subhierarchy to access the fourth place of the subhierarchy.

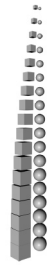




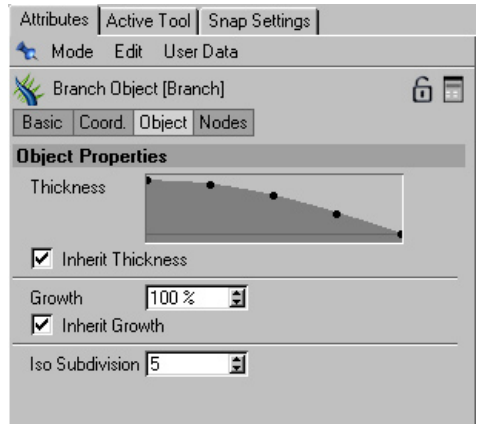


It is also possible to create more than one Object in the same position of the Branch Object subhierarchy. This hierarchy must be linked to a Null Object which forms a nested subhierarchy. This can be used to multiply several Objects along the same Branch Object. You might need to transform the Objects linked to the Null Object in order to make them visible (by default they are all generated in the same location with the same orientation).





Object properties



Thickness

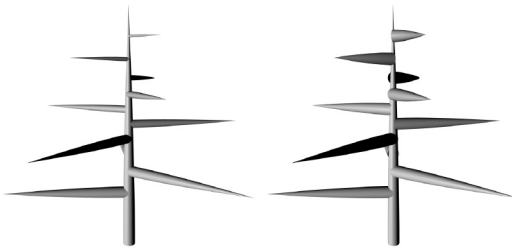
The Thickness parameter is controlled with a curve control. This control provides a curve defining the thickness of the Branch Object. It defines a scaling factor for the segments that are iterated along the Branch Object. The curve represents the length of the iterating Branch Object. More information about the Curve Control widget is available in chapter "1.1.1 Curve Control" at the beginning of this manual.

It is also possible to define the thickness of a Branch Object by editing the Radius of the Spline at the first position in the Branch Objects subhierarchy. In this setting the thickness information is not transmitted to the subsequent Branch Object in the hierarchy. Thus the subsequent branches cannot be adapted to the thickness of their parent Branch Object.

Inherit Thickness

This option checked, which is the default setting, specifies that the thickness parameter from the parent Branch Object is inherited by the selected Branch Object. This setting makes sure that the thickness of all instances of the selected Branch Object are adjusted to match the thickness of the previous Branch Object.

When this parameter is not checked the instances of the subsequent Branch Object all have their default thickness no matter where they are placed.



Growth

The Growth parameter allows you to specify different degrees of growth between 0% and 100%. It is basically a scaling factor that influences the length as well as the thickness of branches.

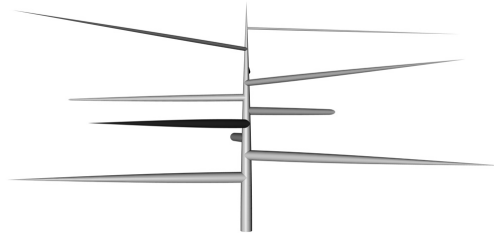
When several Branch Objects are linked together in order to create several branching levels of a tree the subsequent branches need to match the thickness of the branch or trunk they are connected to. This means that the subsequent instances of the Branch Objects have to be scaled according to their placement relative to the previous Branch Object. For this purpose a scaling factor is passed from one Branch Object to the next, controlling the size of the different instances.

The growth parameter also provides you with an easy way to animate the growth of a tree. It is basically a scaling factor that allows you to gradually change the size of a Branch Object between 0% and 100%. By default this parameter is set to 100%. This would represent a fully grown tree. This parameter being transferred through the model hierarchy allows you to animate the growth of the trunk (the first Branch Object) from 0% to 100% with all branching levels of the tree being scaled accordingly. This creates the impression that the tree actually grows and branches develop and grow over time.

Inherit Growth

This option checked, which is the default setting, specifies that the growth parameter from the parent Branch Object is inherited by the selected Branch Object. This setting makes sure that the length of all instances of the selected Branch Object are adjusted to match the size of the previous Branch Object.

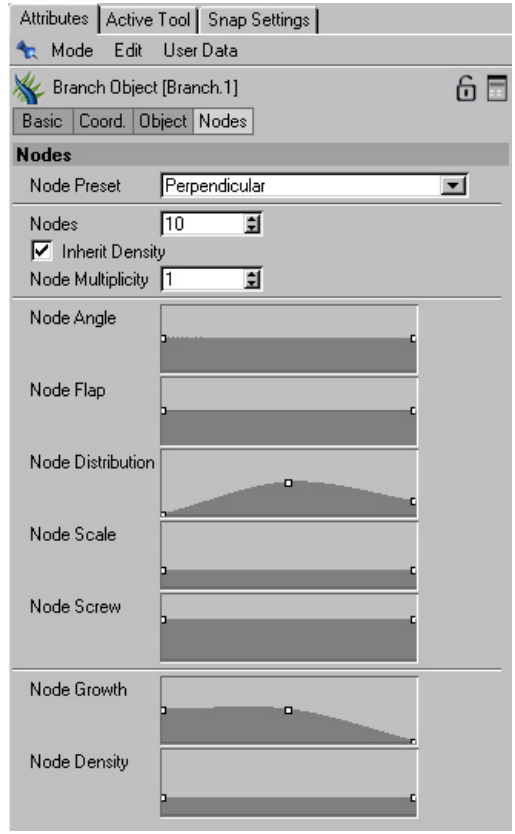
When this parameter is not checked the instances of the subsequent Branch Object all have their default length no matter where they are placed.



Iso Subdivision

This parameter specifies the way the Branch Object is displayed when the display mode is set to isoparms.

Nodes Properties



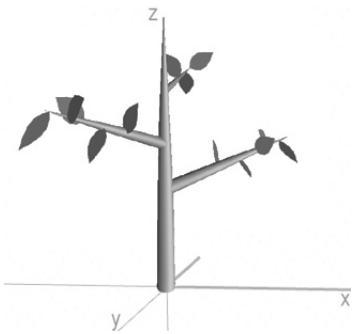
Node Presets

This pull-down menu provides a number of presets that are reflecting common structures of how branches and leaves often are arranged in nature. The specific arrangement is in many cases characteristic for a certain species.
The presets only provide an easy access to constellations of the parameter like Screw etc. These parameters can also be freely defined with the controls of the Branch Object. In the

case of a free definition that is not one of the presets the Node Preset menu will switch to "Custom". The following passage will give you an overview over the different presets. Three parameters of the Branch Object are influenced by the Node Presets: Screw, Flap and Node Multiplicity. When at least one of these parameters is changed manually the preset will change to Custom. One example: The Node Multiplicity parameter is normally set to 1 but with all presets for pair arrangements, this parameter is automatically set to 2. For a closer look at the parameter changes you can have a look at the subchannels of the corresponding parameters.

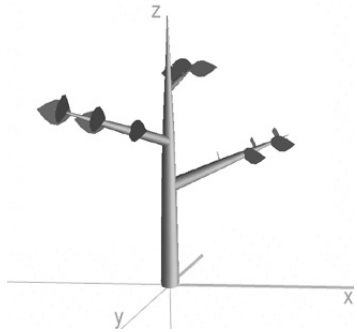
Perpendicular

Defines how the multiplied instances are arranged along the parent Branch Object. The instances are connected in a freely spread manner around the parent Object. Their orientation is perpendicular to the axis of the parent Branch Object.



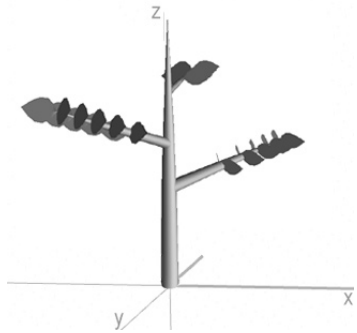
Alternate Perpendicular

Arranges the multiplied instances along the two sides of the parent Branch Object. The arrangement is alternating so that the first instance is created on one side and the next instance on the other side. The orientation is again perpendicular to the axis of the parent Branch Object.



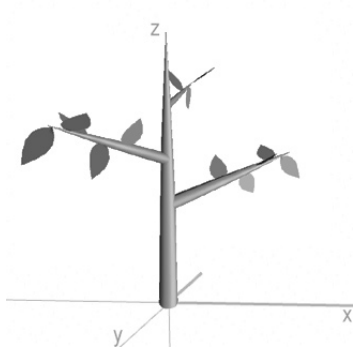
Pair Perpendicular

Arranges the multiplied instances along the two sides of the parent Branch Object. The arrangement is paired so that two instances are always created at both sides. The orientation is perpendicular to the axis of the parent Branch Object.



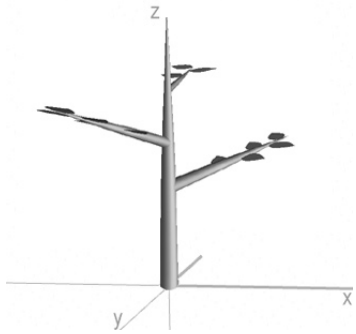
Lateral

Arranges the multiplied instances along the parent Branch Object in a freely spread manner around it. The instances are oriented parallel to the axis of the parent Branch Object.



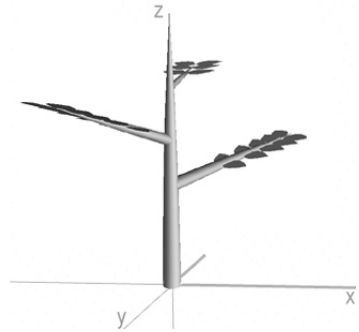
Alternate Lateral

Arranges the multiplied instances along the two sides of the parent Branch Object. The arrangement is paired so that two instances are always created at both sides. The orientation is parallel to the axis of the parent Branch Object.



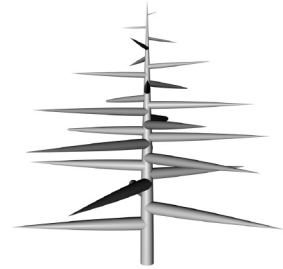
Pair Lateral

Arranges the multiplied instances along the two sides of the parent Branch Object. The arrangement is paired so that two instances are always created at both sides. The orientation is parallel to the axis of the parent Branch Object.



Nodes

When the active Branch Object is linked to another Branch component as it's successor (or child Object) it is multiplied along the previous Branch Object (the parent Object). With the number parameter you can specify the amount of instances that are created by this multiplication.



Inherit Density

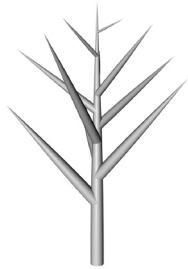
When this option is checked, the Node Density value is passed through the hierarchy and influences the subsequent Branch Objects. When the option is not activated the Branch Object does not inherit the Density parameter from it's parent Branch Object (see also the description of the Node Density parameter).

Node Multiplicity

In order to multiply and arrange it's subsequent Objects the Branch Object generates along itself a number of nodes where the iterated instances are created. By default it produces only one instance in each node but by setting the Node Multiplicity parameter to a higher value it is possible to create more instances in each node.

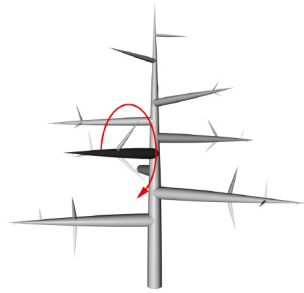
Node Angle

Defines whether a branch is coming out of previous Branch Object in a shallow or steep angle. The inclination of the iterated branches is defined in respect to the previous Branch Object. The curve displays the values over the length of the parent Object.



Node Flap

Rotates the branches around their longitudinal axis. The curve represents the values for the different iterations along the parent Branch Object.



Node Distribution

Specifies the distribution of branches along the parent Branch Object. The curve defines a statistical function that distributes the number of branches specified in the Number parameter according to the max and min of the curve.

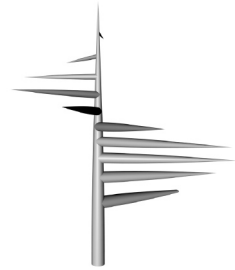
A maximum creates a dense distribution of branches while in relation to this the minima receive a loose distribution. This function is only statistical which means that a linear curve always produces an equal distribution no matter how high the values of the curve are. The actual amount of instances is not influenced by the Node Distribution curve.

Node Scale

Defines a general scaling factor for the different iterations of the branches. The factor is added on top of the scaling inherited from the previous Branch Object. Thus it might happen that with higher values for the Node Scale, branches can actually exceed the thickness of their parent Object. The curve of the Node Scale control represents the length of the selected Branch Object.

Node Screw

Rotates the branches around their parent Branch Object. The curve represents the values for the different iterations along the parent Branch Object.



Node Growth

This parameter defines a scaling factor for the different iterations of the branches which is similar to the Growth parameter of the Object properties. The thickness of the branches is still determined by the Growth parameter inherited from the previous Branch Object. The Growth Scale mainly influences the length of the branches. The curve displays the values over the length of the parent Object.

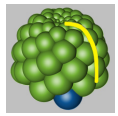
Node Density

This parameter controls how many branches are produced by the Object linked to the currently selected Object. The effect may appear somewhat indirect on the first hand view but you will quickly notice it's usefulness. Imagine you are applying the Node Density parameter to the trunk of a tree. You can specify for example that your tree will have many branches at the top, making for a complex crown. For other species it might be desirable to concentrate most of the branches in the lower part of the tree – you have one easy control to define these characteristics.

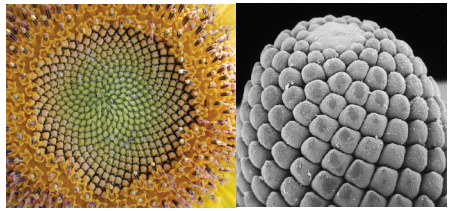
Tip:

In Cinema automatically generated Texture Coordinates are usually created so that the texture origin is somehow in the upper left. The Branch Object's y Texture Coordinate has it's origin at the bottom. This may seem counter-intuitive at first, because textures will appear bottom up when first applied. You can correct this by setting Y Texture Scale of the Texture tag to a negative value. This has been done because it may happen that in growth animations the object can „grow through texture space, either when using the „Growth" Parameter or by hand animating the Texture Scale. Usually for plants the origin of a branch is the place where you would wish the texture to be attached to, which would not automatically be the case for the texture coordinate origin being at the top.

2.4 Phyllotaxis Object

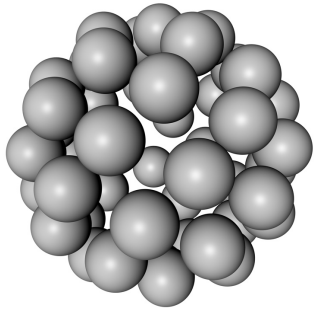


The Phyllotaxis Object distributes Objects on arbitrary surfaces of revolution. It's name comes from the botanical term "phyllotaxis" describing the arrangement of plant organs, such as seeds, in a flower.



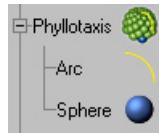
Some examples of phyllotaxis

By default it multiplies an Objects that is linked to it and distributes on the surface of a sphere.



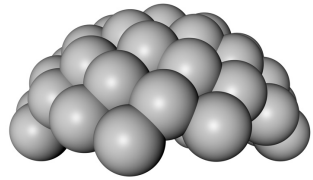
The Object provides a number of parameters that control this arrangement so that the Object is useful for many different structures that are based on this spherical distribution concept. Another example could be the arrangement of the petals of a flower.

Similar to the Branch Object the positions in the subsequent hierarchy are fulfilling specific functions:



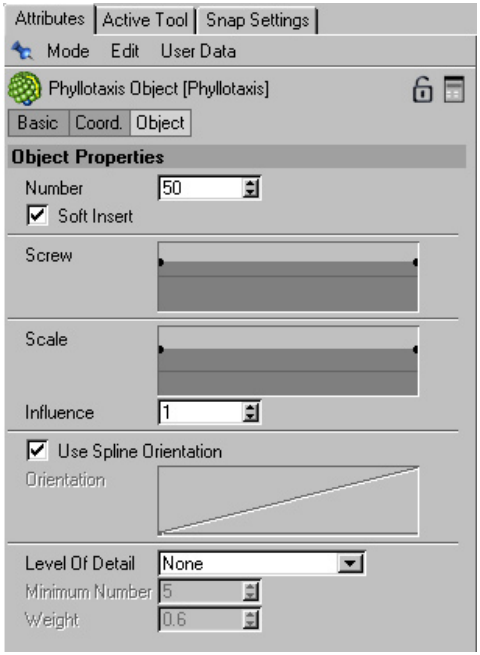
The first position in the subsequent hierarchy is occupied by the Spline defining the outline of the spherical surface which is the basis for the arrangement of the iterated Objects. By default this Spline is an Arc. Different Splines can be assigned to create many different kinds of surfaces of revolution.

This Spline defines the radius of the sphere that is by default created with the Arc. It also specifies the area of the spherical surface that is covered by iterated instances.



The second position in the subhierarchy contains the Object that is multiplied by the Phyllotaxis Object. This can be any given Cinema 4D Object. It is also possible to create a nested subhierarchy in this position.

Object properties



Number

Specifies the number of instances that are created of the multiplied Object.

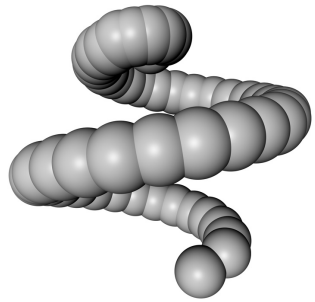
Soft Insert

This option is important for the animation of a Phyllotaxis Object. When the Number parameter changes over time the new instances are "faded in" or existing instances are "faded out". This means that new instances are smoothly scaled from zero to their target size when they are inserted and scaled down from their initial size to zero when they are removed.

When the Soft Insert option is disabled new instances pop in at their target size or existing ones are removed without any smooth transition.

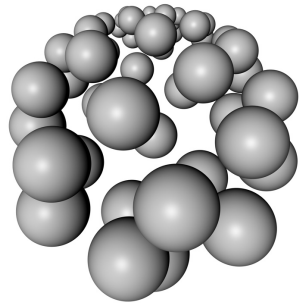
Angle

Defines the arrangement of the iterated instances on the spherical surface. By default the instances are arranged according to the golden section that can often be found in Nature. The Angle parameter specifies a rotation angle in relation to the center point of the Phyllotaxis. The leftmost point corresponds to the first iteration and the rightmost point to the last iteration.



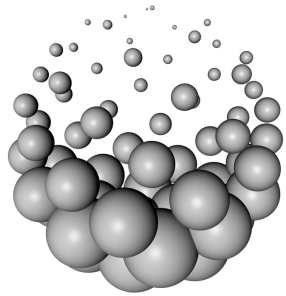
Scale

This parameter allows you to specify a scaling factor to the iterated instances. The leftmost point corresponds to the first iteration and the rightmost point to the last iteration.



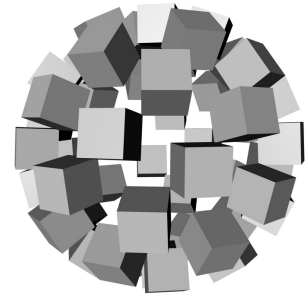
Influence

Specifies how strongly the position of the iterated instances are influenced by their size. When, for example, the first iterations are scaled down and the others remain at their initial size the surface appears more packed in the area where the big iterations are, whereas in the other part holes seem to appear. When the Influence parameter is set to a higher value this unevenness can be reduced and the bigger iterations are pushed towards the smaller ones in order to create an even density over the surface. Setting the Influence parameter to 1 corresponds to full influence, whereas 0 disables the influence.



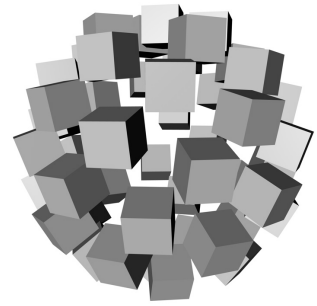
Use Spline Orientation

This checkbox determines whether the orientation of the iterated instances follows the tangent of the spline describing the outline of the overall shape or not. When it is unchecked and the instances are not set to follow the Spline the Orientation parameter is available. In the other case this parameter is greyed out.



Orientation

Specifies a rotation value for the iterated instances. The leftmost point corresponds to the first iteration and the rightmost point to the last iteration.



Level of Detail

The Level of Detail pull-down menu allows you to define the way in which the complexity management is handled. The following options are offered:

None

Disables the influence of Cinema 4D's complexity management on the selected Phyllotaxis Object.

Drop&Resize

Xfrog 5 Objects have their own LOD algorithms which analyze the geometry and reduce the resolution without changing the overall shape of the Object. Briefly, the resolution is reduced by the removal of iterated instances from the spherical surface while adjusting the placement of the remaining instances and enlarging them. This ensures that the overall volume is maintained and the model does not appear to be "thinned out". The Drop&Resize option enables this complexity management.

Minimum Number

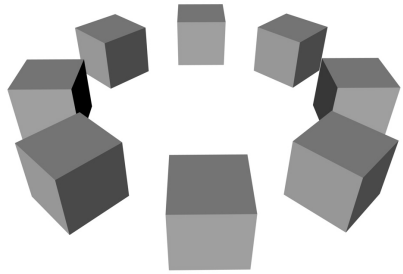
Allows you to specify a minimum amount of iterated instances that the complexity management will not surpass.

Weight

Allows you to specify how strong the remaining elements will be enlarged. This parameter set to 1 will result in a direct mapping: Half of the elements are removed and the remaining ones are doubled in their size. The parameter set to 0 disables the scaling. The default value is 0.6 which delivers the best results for most cases.

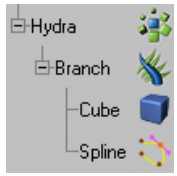
2.5 Hydra Object

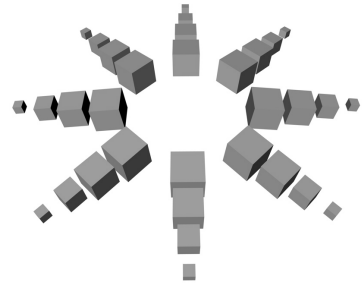
An Object linked to the Hydra Object is multiplied and distributed in a circle.



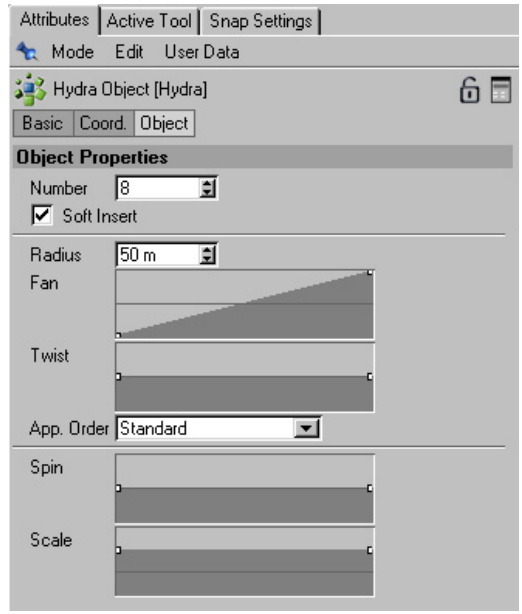
The Object that is multiplied needs to be linked in the first position of the subhierarchy of the Hydra Object.

The Hydra Object can multiply any given geometry, for example also other Xfrog Objects:





Object Properties



Number

Specifies the amount of instances that are created of the multiplied Object.

Soft Insert

This option is important for the animation of a Hydra Object. When the Number parameter

changes over time the new instances are "faded in" or existing instances are "faded out". This means that new instances are smoothly scaled from zero to their target size when they are inserted and when instances are removed they are scaled down from their initial size to zero.

When the Soft Insert option is disabled new instances pop in at their target size or exisitng ones are removed without any transition.

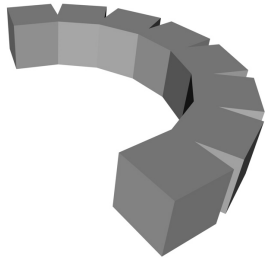
Radius

Specifies the radius of the circle the iterated instances are arranged on.

Fan

Defines the segment of the circle that is occupied by iterated instances. The Fan parameter defines the degree of rotation around the X-axis of the Hydra Object. The leftmost point corresponds to the first iteration and the rightmost point to the last iteration. The intermediate values are interpolated. Additional control points can be inserted by right-clicking the curve.

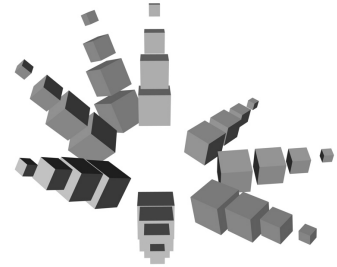
The numerical values of the control points can be accessed in the subchannels of the Angle parameter.



Twist

Specifies a degree of rotation around the Z-axis of the Hydra Object. The leftmost point corresponds to the first iteration and the rightmost point to the last iteration. The intermediate values are interpolated. Additional control points can be inserted by right-clicking the curve.

The numerical values of the control points can be accessed in the subchannels of the Angle parameter.



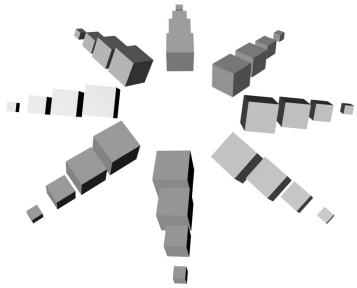
Application Order

There is a slight change in the behavior of the Twist parameter in the previous versions of Xfrog and the Xfrog 5 Plugin. The Application Order parameter allows you to switch between the two behaviors. The Standard behaviour creates the best results but the Classic behaviour is maintained for backwards compatibility. In some cases when you are importing models that were created in a previous version of Xfrog you might want to use the Classic behaviour.

Spin

Specifies a rotation of the iterated instances around their local longitudinal axis. The leftmost point corresponds to the first iteration

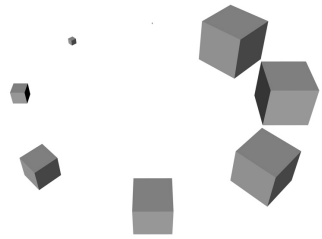
and the rightmost point to the last iteration. The intermediate values are interpolated. Additional control points can be inserted by right-clicking the curve.



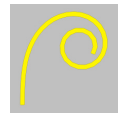
Scale

Specifies a scaling factor for the iterated instances. The leftmost point corresponds to the first iteration and the rightmost point to the last iteration. The intermediate values are interpolated. Additional control points can be inserted by right-clicking the curve.

The numerical values of the control points can be accessed in the subchannels of the Angle parameter.

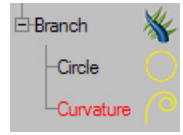


2.6 Curvature Object



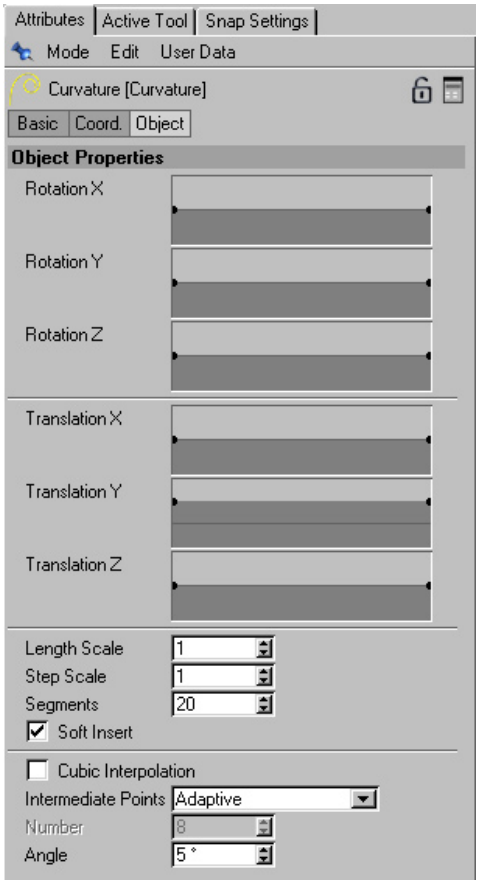
The Curvature Object is a spline Object. The difference to other splines, is that it is not predefined, or defined, by interactively drawing control points, as it's definition is implicit. The Curvature Object consists of a number of segments that can be influenced parametrically. Parameters that define rotation and translation values for the splines curvature are provided. It is also possible to use mathematical functions to define the curvature.

The Curvature Object can be used with any Xfrog or Cinema 4D Object that can use a Spline. One example would be the use together with the Branch Object as shown in the following model hierarchy:



The above hierarchy is equivalent to the former Horn Component of previous Xfrog versions.

Object Properties



Rotation X

Specifies a rotation around the X-axis of the spline segments that the Curvature Object consists of. The leftmost point corresponds to the first spline segment and the rightmost point to the last spline segment.

Rotation Y

Specifies a rotation around the Y-axis of the spline segments that the Curvature Object consists of. The leftmost point corresponds to the first spline segment and the rightmost point to the last spline segment. The intermediate values are interpolated.

Rotation Z

Specifies a rotation around the Z-axis of the spline segments that the Curvature Object consists of. The leftmost point corresponds to the first spline segment and the rightmost point to the last spline segment.

Translation X

Specifies a translation (a displacement) along the X-axis of the spline segments that the Curvature Object consists of. The leftmost point corresponds to the first spline segment and the rightmost point to the last spline segment.

Translation Y

Specifies a translation along the Y-axis of the spline segments that the Curvature Object consists of. The leftmost point corresponds to the first spline segment and the rightmost point to the last spline segment.

Translation Z

Specifies a translation along the Z-axis of the spline segments that the Curvature Object consists of. The leftmost point corresponds to the first spline segment and the rightmost point to the last spline segment.

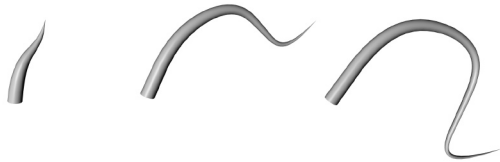
Length Scale

Defines a scaling factor for the length of the segments of the spline.

Step Scale

This parameter allows you to specify an additional factor for the length and angle of the

spline segments. The observation that led us to implement this parameter is coming from the natural growth processes of plants. As an example you may think of a growing leaf: In the beginning the leaf is very small and has only a light curvature. As it grows it does not only get longer, also the curvature gets more accentuated. The Move Scale parameter lets you gradually apply this characteristic to your model. When the parameter is increased it makes the individual spline segements of the Curvature Object longer and the angle between the different segments steeper. This behaviour simulates the described natural phenomenon.

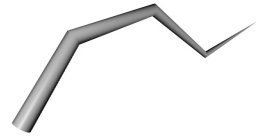


Branch Object with a Curvature Spline in different Step Scale stages (Step Scale 0.3, Step Scale 1.0, Step Scale 1.6)

Segments

Defines the amount of spline segments the Curvature Object consist of. This parameter determines the shape of the Curvature Spline. The more segments are used for the definition, the smoother the resulting curves will be. However, the amount of segments should be kept as low as possible because the option to apply a Cubic Interpolation to the Spline offers a much more efficient way to create a smoothly curved Spline. The Segments parameter should really be regarded as a modeling parameter used to define the actual shape of the Spline.

In this sense it is also not useful to apply a Level Of Detail management to the Segments parameter, as this may influence the shape of the curve. For Level Of Detail management it is better to control the way how intermediate points are generated and in which density (see parameters "Intermediate Points, Number, Angle").



Branch Object with a Curvature Spline consisting of 4 Segments.

Branch Object with a Curvature Spline consisting of 4 Segments and Cubic Interpolation.

Branch Object with a Curvature Spline consisting of 20 Segments without Cubic Interpolation. The curve is slightly different than above, although the Spline parameters are the same.

Soft Insert

This option is important for the animation of a Curvature Object. When the amount of spline segments changes over time the new segments are "faded in" or existing segments are "faded out". This means that new segments are smoothly scaled from zero to their target size when they are inserted and when segments are removed they are scaled down from their initial size to zero.

When the Soft Insert option is disabled new segments pop in at their target size or existing ones are removed without any transition.

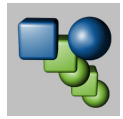
Cubic Interpolation

This option creates a smooth curving of the Spline even if the amount of Segments is very low. The smoothing is achieved by inserting additional intermediate values, that are calculated with an interpolation algorithm, between the existing points defining the curve. . For illustration please view the images in the description of the Segments parameter on the previous page.

Intermediate Points, Number, Angle

These parameters define the curving of the Spline. They provide different strategies of interpolation between the control points that actually define the Spline (vertices). For a detailed description of the interpolation strategies please refer to the section 7, chapter "Spline Primitives" in the Cinema 4D documentation.

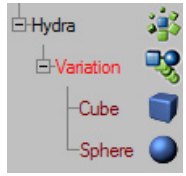
2.7 Variation Object

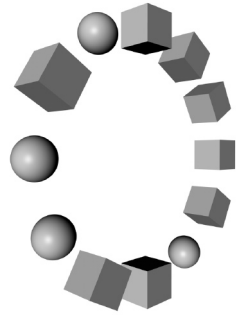


The Variation Object is used with other multiplier Objects. It allows you to alternatingly iterate different Objects with the same multiplier Object. The Object works only together with Xfrog Multiplier Objects, such as Branch, Hydra and Phyllotaxis.

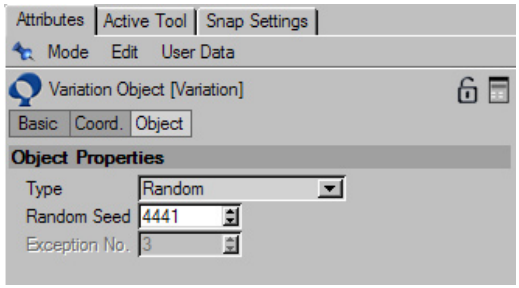
The Variation Object is linked to the Multiplier Object and the Objects that are to be iterated are linked as children to the Variation Object. The iterated Objects can alternate regularly, randomly and it is possible to assign exceptions for specific positions

In the example hierarchy below the Variation Object is used with the Hydra Object and iterates a Cube and a Sphere Object.





Object Properties

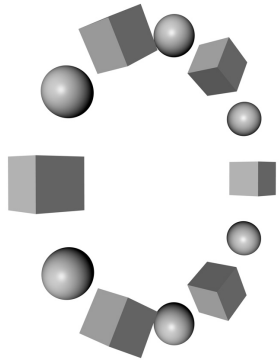


Type

This pulldown menu allows you to select from different ways of how the alternation between the iterated Object is calculated.

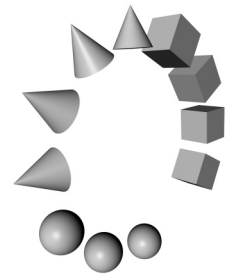
Sequential

Generates a sequential alternation between the iterated Objects. In the case of our example it would create the following arrangement:



Spread

Distributes the iterated instances one after the other evenly on the iteration positions.

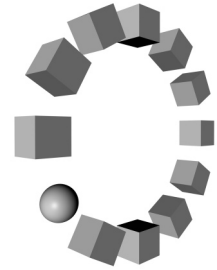


Random

Generates a randomly changing alternation of the iterated Objects. As this alternation is based on a numerical random seed value it is possible to reproduce random alternations. The first example picture shown above is an example for a random alternation.

Exception

Allows you to define one specific exception for one of the iterations where the second of the iterated Objects is used. The exception is defined numerically refering to the number of the iteration.



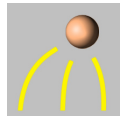
Random Seed

This parameter is only available when the "Type" parameter is set to "Random". It allows you to specify a number that is used as a basis to generate a random number. The use of this seed value allows you to reproduce the arrangements that are created by the Variation Object.

Exception Number

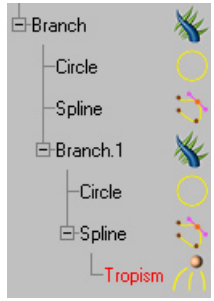
This parameter is only available when the "Type" parameter is set to "Exception". It allows you to specify the number of an iteration where the second Object of the iterated Objects is created. This Object is linked in the second position of the subhierarchy of the Variation Object.

2.8 Tropism Object



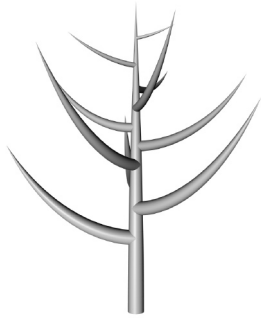
The Tropism Object allows you to define force fields to bend splines into a certain direction. The Tropism Object can be used, for example, to create light sources and make splines turn towards the light source – this phenomenon is called phototropism. In the same way you can define a gravitropism which would cause the branches of a tree to hang down towards the ground as if they were attracted by the gravitation of the earth.

The Object works as a Cinema 4D Deformer Object (see Cinema 4D manual, section 7.11) with the difference that it only influences splines. The Tropism is connected to the spline which it is deforming. The following example shows a Tropism Object used to bend the branches of a tree towards a virtual light source.

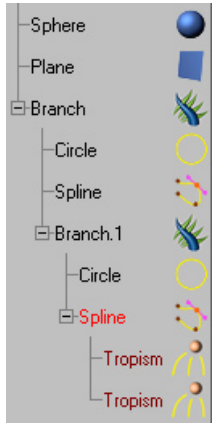


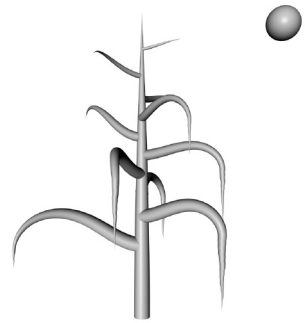
The default setting of the Tropism Object directs the branches upwards, as if a virtual sun was right above the tree – in more technical

terms: The Branches are pointing in the direction of the Y-Axis.



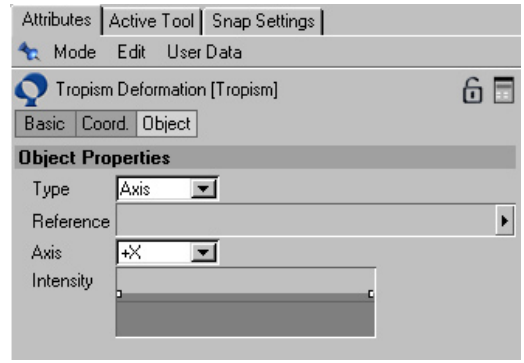
It is also possible to combine several Tropism Objects and add their respective influences. The following example shows one Tropism Object set to "Point" Type combined with a second Tropism set to "Axis" Type (for a description of the Types see the Object Properties section on the next page).





It is possible to combine as many Tropism Objects as you wish. The influence is added according to the order they are linked together: The first Tropism is applied first, etc.

Object Properties



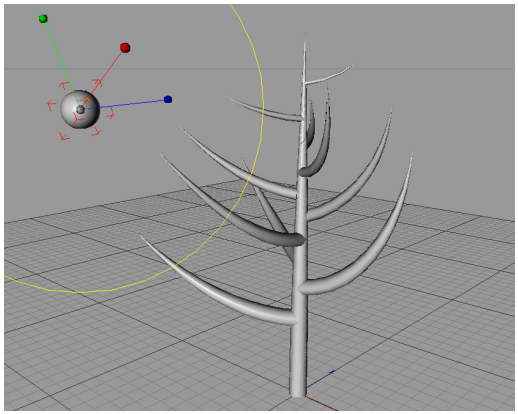
Type

This pulldown menu allows you to switch between two definitions of a force field.

Axis

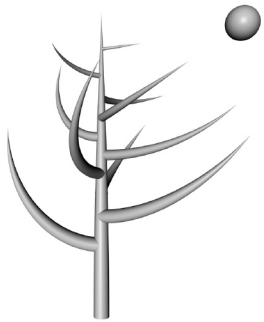
Turns the deformed Spline in direction of one of the three axis of the coordinate system (see

picture above). When you use the Axis setting together with a reference Object the Splines will be bent in the direction of the corresponding axis of the reference Object. This means when you rotate the reference Object the axis for the Tropism will be rotated as well.



Point

Directs the deformed Spline towards another Object which is used as the reference of the Tropism Object (in the example below it is a Sphere Object – but any Object in Cinema 4D can be used). This option would allow you to use a lightsource existing in your scene as the point towards which the Spline turns.

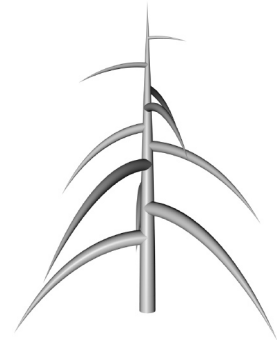


Reference

If the Type of the Tropism Object is set to "Point" you can specify the reference towards which the Splines should turn in this field. Grab the reference Object in the Object Manager and drag it to the "Reference" field. The icon along with the Objects name will appear.

Axis

In this pulldown menu allows you to select the axis and direction into which the Splines turn when the Tropism Type is "Axis". The default value is +Y which makes the Spline point in the positive direction of the Y-Axis. In order to define a gravitropism you would select -Y in order to cause the Spline to point along the negative direction of the Y-Axis.



Intensity

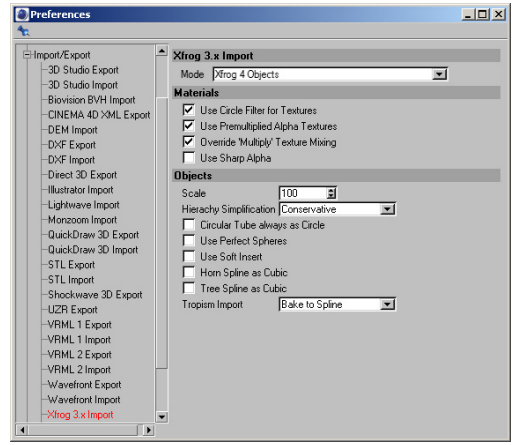
Specifies the intensity of the deformation. Assigning a positive value makes the Splines point into the specified direction, whereas negative values makes them point into the opposite direction. Setting the curve first to a positive value and then, for the values on the left side of the curve, to a negative value creates the very common phenomenon of a combination of Phototropism and Gravitropism. At the beginning of the branches they point upwards as they grow towards the sun, while the ends of the branches point to the ground because they are thin and tend to hang down.

2.9 Import/Export Settings

The Xfrog 5 Plugin also adds an option to the Import and Export settings in the Edit Preferences Menu.

In order to provide compatibility to the standalone versions of Xfrog (such as Xfrog 3.5) it is possible to read existing .xfr files into Cinema 4D. However, as the model hierarchy is handled in a slightly different way in Cinema 4D, you can control the way how the Xfrog hierarchy is interpreted and transformed into a Cinema 4D hierarchy.

When you expand the Import/Export section you will see the Xfrog Import options. If the classic Xfrog Plugin is installed as well as the Xfrog 5 plugin, you can choose which plugin you want to use. We recommend currently for PC users that you install both the classic Cinema 4D Xfr import plugin and the Xfrog 5 Plugin for Cinema 4D. Between the 2 plugins, you can read any .xfr file, including animated ones. For Mac users (and PC users) all Xfrog-Plants CDs, contain 60 .c4d models, which are optimized for your use.



Mode

Xfrog 5 Objects

Converts the Components of the imported Xfrog 3.5 file into Xfrog 5 Objects. This option uses the new Xfrog 5 Plugin and gives access to the Xfrog 5 functionality inside Cinema 4D.

Classic Plugin Object (with Animation)

Uses the old Xfrog Plugin to import a Xfrog 3.5 file as a single Object. If the imported model is animated, this animation is imported as well and will playback in Cinema 4D.

Classic Polygon Mesh (no Animation)

Uses the old Xfrog Plugin to import a Xfrog 3.5 file as a polygon mesh. Animation is not imported.

Materials

This option allows you to specify som egeneral settings for the imported materials.

Use Circle Filter for Textures

Sets the interpolationfilter of the textures of the imported materials to Circle. A detailed of this filter can be found in the Cinema 4D manual in section 18.5.2.

Use Premultiplied Alpha Textures

Switches the Premultiplied channel of the imported textures on. See also Cinema 4D manual, section 18.6.2.

Override 'Multiply' Texture Mixing

Deactivates the "Multiply" option of the mix mode of imported textures. This can be helpful with textures that have a color assigned in addition to the texture. See also Cinema 4D manual, section 18.5.2.

Use Sharp Alpha

Adjusts the Alpha channel in a way that its contour characterisitic is sharp. See also Cinema 4D manual, section 18.6.1.

Objects

This option is available, when the new Xfrog 5 Plugin is used for import.

Scale

Allows to specify a scaling factor that defines the size of the imported model in Cinema 4D.

Hierarchy Simplification

This option is available, when the new Xfrog 5 Plugin is used for import.

Always

This option will always interpret the Xfrog hierarchy in a way that the resulting Cinema 4D hierarchy will be as simple as possible.

To give you an example for the simplification think about the following case: To create a sphere in Xfrog 3.5 you needed to create a Simple component and then assign a sphere primitive to it. In a strict translation to Cinema 4D this would result in a Null Object and a sphere linked to it. As in most cases the Null Object is not needed, it will be removed by this import option.

Only when the addition of the transformation assigned to the Simple and its primitive would lead to a clearly wrong result, the separation of the two elements is preserved in the import.

Conservative

This option also tries to interpret the Xfrog 3.5 hierarchy in the simplest way. The difference to the "always" option is – to use the same example case – that any time when different translations are assigned to the Xfrog

component and its primitive the separation of the two will be preseved. This respects that in some cases the two elements get different translations assigned just for convenience of the workflow, although the shape of the model is not influenced.

Off

This option imports the hierarchy in exactly the same way it used to be in Xfrog 3.x. The resulting hierarchy in Cinema 4D might be a bit less elegant but like this the model will definitly be as it used to be in Xfrog 3.5. Caveat - this may create a lot of polygons if every point in a circle is forced to be created, instead, for example, a spline

Cicular Tube alsways as Circle

When this option is activated all "Tube" primitives from Xfrog 3.x are imported as Circle Splines. The "Tube" primitive is, when no manual changes of its profile were made, a circle. In most cases it is best replaced by the Circle Spline when imported into Xfrog 5 – this is the default setting. In cases where a different shape is needed, this option needs to be deactivated.

Use Perfect Spheres

Cinema 4D makes a distinction between perfect and normal spheres. The perfect spheres are rendered always in a perfectly round shape no matter what their segment resolution is. Normal spheres are dependent on the segment resolution. Similar to the Circle Spline mentioned in the paragraph above, this option lets you define whether the "Sphere" primitive of Xfrog 3.x should be replaced by a perfect or a normal sphere when imported into Xfrog 5. By default the Xfrog 3.x "Sphere" pimitive create a diamond shape. To import this shape as it was, the option needs to be deactivated.

Use Soft Insert

Activates the "Soft Insert" option for the Xfrog 5 objects that the imported model is translated into. Soft Insert smoothly fades additional elements in or out when created during an animation sequences.

Horn Spline as Cubic

Uses the curving of the imported "Horn" components as basis for a cubic interpolation of a Curvature Spline. There might be certain differences in comparison with the original. When the option is deactivated, a linear Spline is used and the curving is reproduced as in the original. This may result in a high amount of control points of the Spline and can be difficult to manipulate.

Tree Spline as Cubic

Uses the curving of the imported "Tree" components as basis for a cubic interpolation of a Curvature Spline. Like with the "Horn" there might be certain differences in comparison with the original. When the option is deactivated, a linear Spline is used and the curving is reproduced as in the original.

Tropism Import

IIn Xfrog 3.x tropisms were parameters integrated into the parameter sets of certain components, the Phototropism and the Gravitropism were available. In the translation to Xfrog 5 they can either be translated as two separate Tropism objects or as one.

Bake to Spline

Combines both Xfrog 3.x tropism parameters into the Spline of one Tropism object.

Separate Tropisms

Gravitropism and Phototropism are translated into two separate Tropism objects.

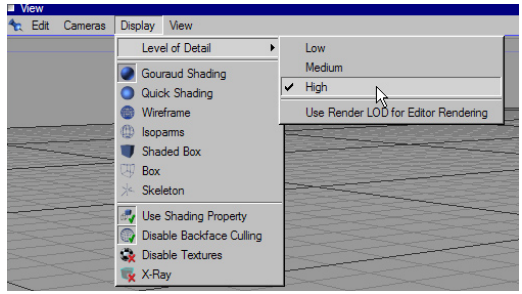
3. Xfrog 5 and Cinema 4D

This chapter will provide some tips on how to make the best out of the combination of Cinema 4D and Xfrog 5.

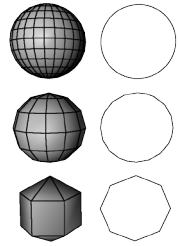
3.1 Controlling Level Of Detail

Often when you build complex models there is a need to have it generated with different Detail so an Object which is small on the screen or in the background doesn't use as many Polygons as your Hero Objects in the foreground.

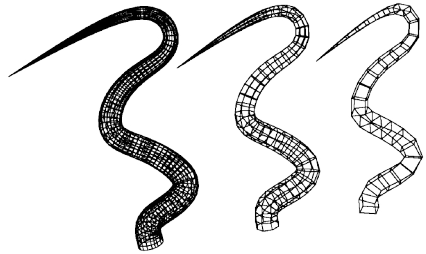
Cinema 4D has a Level Of Detail (LOD) System built in and the Xfrog Objects make use of this and also provide their own means of Polygon Reduction. There are two global Level Of Detail values defined for the Scene, one for the Viewports and one for Rendering. The Level Of Detail Value is a percentage (initially 100%) which is passed to every Object at generation time, it is up to the Object how to make use of it. Be aware that some Objects may not make any use of it at all or depending on certain Object Settings. It is also worth noting that the Level Of Detail Value may exceed 100%. The Level Of Detail for the Viewports can be set in the Display Menu of each Viewport (see Screenshot), the Render LOD can be set on the Render Preferences Panel.



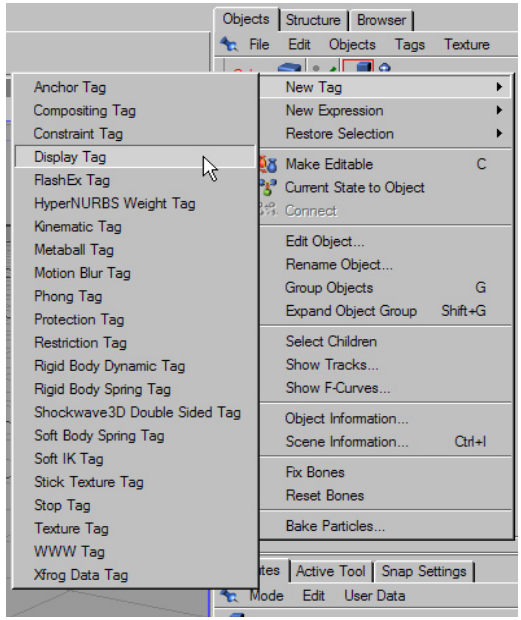
To account for Level Of Detail Procedural Objects usually internally vary Parameters connected to their visual fidelity. A Sphere Object for example internally varies the Number of Segments if the LOD is different from 100%, Spline Curves vary their Intermediate Points Settings. The Screenshots show examples of a Sphere and a Spline Object at 100%, 50% and 25% Level Of Detail.

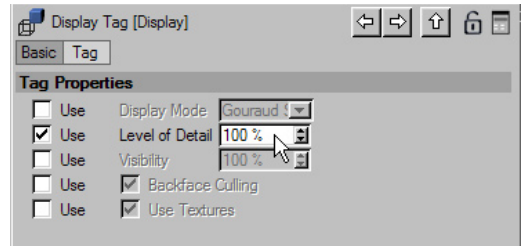


Xfrog Objects can either directly make use of this. When the Input Splines for a Branch Component automatically reduce their Level Of Detail the simplified Objects automatically get used to generate the Mesh. To see the effect have a look at the Wireframe of an example Branch Component below at 100%, 50% and 25% Level Of Detail. Some Xfrog Objects are able to react directly to Level Of Detail, see the Object Descriptions for more details.

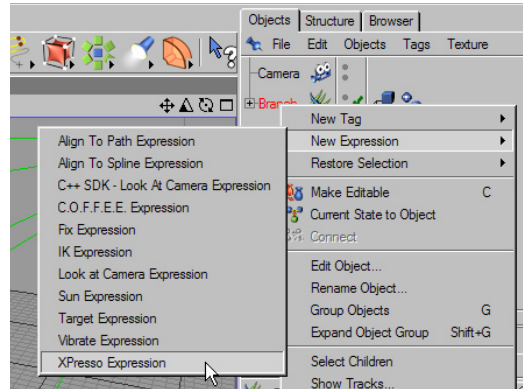


Instead of setting a Level Of Detail globally, you can also specify a Level Of Detail for every Object in the Scene separately. To control the Level Of Detail of an Object you have to assign a 'Display Tag' to the Object by right-clikking on the Object in the Object Manager and choosing 'New Tag' -> 'Display Tag'. In the Display Tag Properties activate the 'Use' Checkbox in front of the 'Level of Detail' Value to manipulate that value for the Object the Tag is applied to and all of its Subobjects (if they don't have a Display Tag on their own).



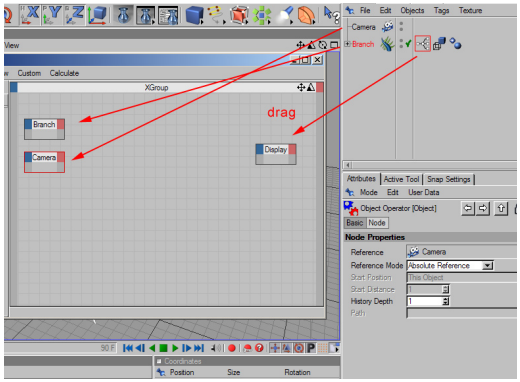


This Level Of Detail Value can be edited and animated like any other parameter in Cinema 4D, but you may want to have your LOD controlled by the distance of the Object to the Camera. This can easily be archived by setting up a simple XPresso Expression. To do so, apply an XPresso Tag to the Object you want to control the LOD of by right-clicking on the Object in the Object Manager, then selecting 'New Expression'->'XPresso Expression'.

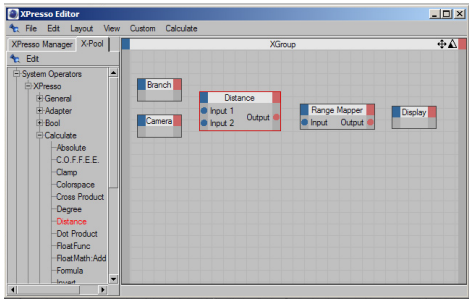


The XPresso Editor should open up right away, (later you can open it by double-clicking on the XPresso Tag). Drag the Object you want to control the LOD of (a Branch Object in this example), it's Display Tag and the Object you want to use the distance of (Camera) into the XPresso Editor. If, like in this example, your Level Of Detail is somehow controlled by a Ca-

mera, you need to check the 'Camera Dependent' Checkbox in the Properties Panel of the XPresso Tag(!)

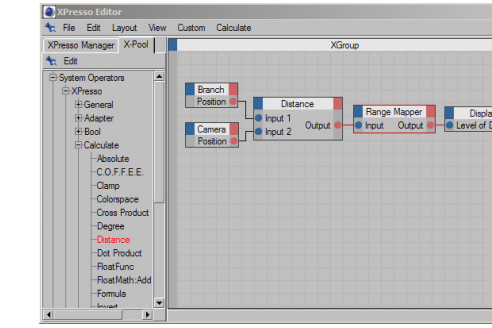


Now we want to use the Distance between Camera and Branch Object to Control the LOD Value in the Display Tag. For that purpose you need to drag two more nodes into the XPresso Editor. Select the X-Pool Tab on the left side of the XPresso Editor and look for the nodes 'System Operators'->'XPresso'->'Calculate'->'Distance' and 'System Operators'->'XPresso'->'Calculate'->'Range Mapper' and drag them both into the Editor.

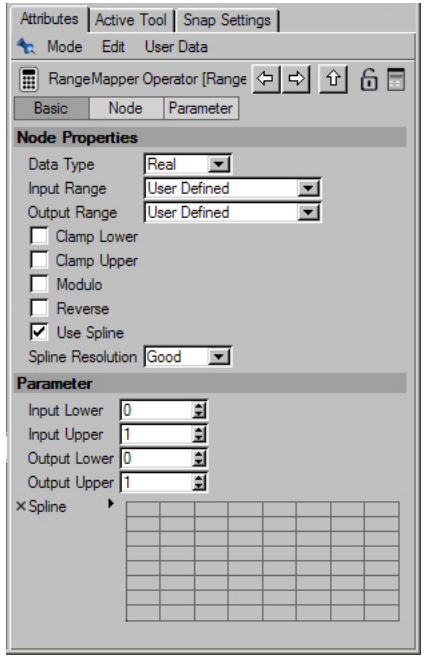


The next step is to connect all these nodes. The Branch and Camera Objects don't have any Output yet and we need the position of both to calculate their distance. To create an output port left-click the red rectangle in the upper right of the Branch Node and Choose 'Coordinates'->'Global Position'->'Global Position'. Do the same for the Camera Object. The Display Object needs the LOD Value to be available as input port. Click the blue rectangle in the upper left of the Display Tag Node and choose: 'Tag Properties'->'Level of Detail'.

To connect the ports click on one of the ports and drag the connection line to the port you want to connect it to. Use this to connect the Position of Branch and Camera with Input 1 and Input 2 of the Distance Node, the Output of the Distance Node with the Input of the Range Mapper and the Output of the Range Mapper with the Level Of Detail of the Display Node. Congratulations! Your XPresso Expression is now up and running.



The only thing that remains is using the Range Mapper for fine tuning. For this step, click on the Range Mapper, which will let it's properties show up in the Attribute Editor. The purpose of the Range Mapper is to decide what LOD value you want to get used at what distance. At first set the 'Output Range' to 'Percent'.



'Input Lower' and 'Input Upper' are the mininum and maximum distances in-between which you want your LOD to change. Choose a distance value for the camera at which you want a LOD of 100% and enter it as 'Input Lower' and enter a distance at which you want your LOD to be reduced to a minimum as 'Input Upper'. A good value to start with is around twice the height of your object as 'Input Lower' and five to ten times as much as 'Input Upper'. You'll probably need some tweaking to achieve good results.

'Output Lower' and 'Output Upper' are the LOD values the Range Mapper fit inside of. Currently it would create an LOD of 0% at minimum distance and an LOD of 100% at maximum distance which is the opposite of what you want. You may also not want to go all the way down to a LOD of 0%, 10% is usually a better minimum. To archive this set ‚Output Lower' to 100% and ‚Output Upper' to 10%.

You may want to use the 'Clamp Lower' and 'Clamp Upper' checkboxes to limit the LOD to the range you specified (10%-100%), otherwise these values may be exceeded if the distance between your objects is not in the range specified by 'Input Lower' and 'Input Upper'.

With the Spline Curve you could also create a nonlinear transition between the LOD values while the camera moves away. This often makes sense, because when a camera moves away from an object at a constant speed the objects size on screen is reduced faster at the beginning and slower at the end. You can do the same with your LOD. To do this left-click several times somewhere into the spline window to create control points (9 times is good), then right click and choose 'Root'.

When you now move the Camera Object around, the Level Of Detail of the Object is dynamically updated based on distance. Please refer to the appropriate sections in the Cinema 4D Manual for more information on Level Of Detail and XPresso.

3.2 Detail Editing of Objects

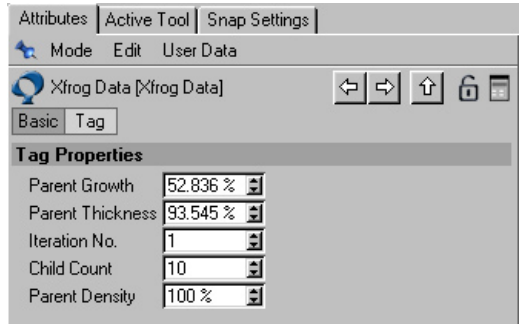
The great power of Xfrog lies in its capability to controll complex structures with a small number of parameters. When you create a tree you are not dealing with every branch separatly but controlling the entire branching level with one set of parameters. This allows of course for a quick and simple editing process – but there are also the cases where you have an already very convincing tree that is basically satisfying all your needs, except that you want this one branch to be longer and curved in a different way. This is possible when you choose the option "Make Editable" in the "Objects" menu of the Object Manager. This option converts the parametrically defined Branch Object (the function works with any parametrical Object - the Branch Object is only used as an example) into a polygonal Object that is defined by the points of its polygon mesh (more information on this function is available in the Cinema 4D manual in the section called 'Make Editable').

The points of the converted Object can then be edited independently as in the example below:

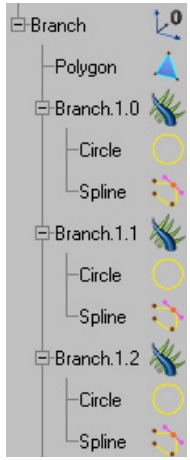
Example of a Branch Object that is converted into the polygonal formatand then edited with the Magnet tool.

As the parametrical Object looses its parametrical definition through this conversion you loose the possibility to edit the Object through these parameters. This is an irreversible process. (In order to realize the powerful modeling tools provided by Xfrog, the Xfrog Objects transmit parameters to subsequent Objects in the hierarchy. This transmission is stopped when you convert one of the Objects in the hierarchy into the polygonal format).

Looking at the Branch Object as an example, this means that you have to start the conversion process at the very top of the hierarchy and then move down to the linked Objects. This approach preserves the shape of your model because the parameters that would normally be directly transmitted to the next Object are stored in a specific tag, the Xfrog Data Tag. This Tag contains the data that are normally passed from the parent to the child and makes them available to the child Object although they are not transmitted from the parent anymore. It is necessary that you work from the top of the hierarchy down to the subsequent Objects because otherwise the Xfrog Data Tag cannot be generated correctly and the shape of the tree gets altered.

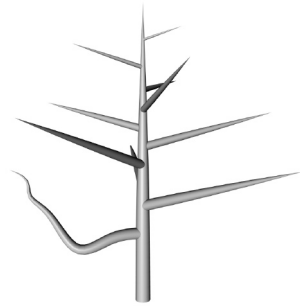


After the conversion of the Branch Object into the polygonal format the subsequent Branch Objects are separated out into their iterations and instead of one single Branch Obejct describing a whole branching level each iteration is independently linked to the parent Object.



The first Branch Object is converted into the polygonal description and the child is separeted into its iterations

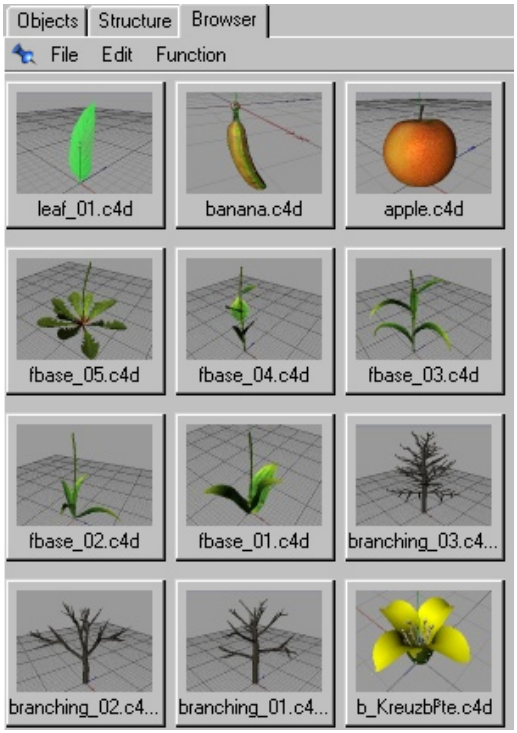
As only the first Branch Object is converted the subsequent Objects remain in their parametrical description. They can still be edited parametrically and each Object can be edited indepent from the other Obejcts on the same branching level.



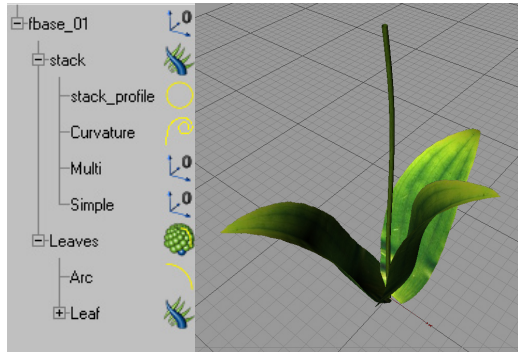
3.3 Using Prefabricated Model Parts

With the Library Browser, Cinema 4D provides a powerful tool to optimize your workflow. It gives you access to often used scenes, materials, textures, pictures, etc. (more information about this Browser can be found in the C4D manual).

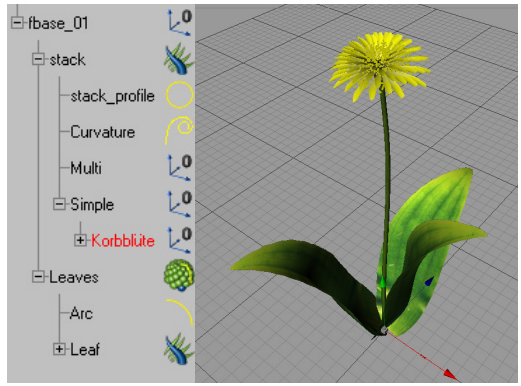
As certain structures, like hierarchies with several Branch Objects linked together, often reappear in Xfrog models, it is useful to build a library of prefabricated generic parts that you can easily recombine to create new models. With the Xfrog 5 CDs we include a collection of prefabricated generic parts that you can use to quickly build your own models.



Just drag one of the icons in the Browser into the Object Manager and the hierarchy of this model part will appear in your scene.

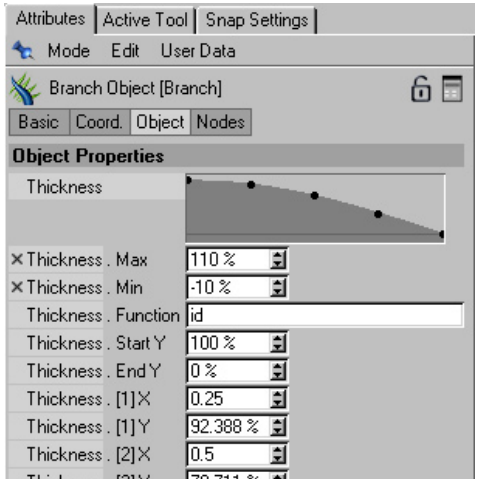


When you drag another Model part from the Browser into the Object Manager and link it to the correct position of the first part it is possible to create a wonderful flower in just two steps. Based on this scene you can play with the parameters and add more Objects to create the specific species that you need.

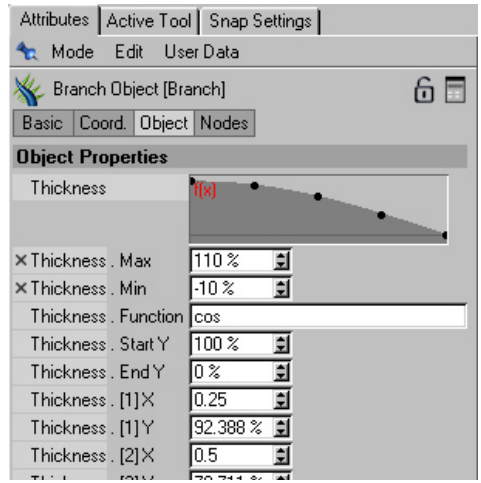


4 Using Functions

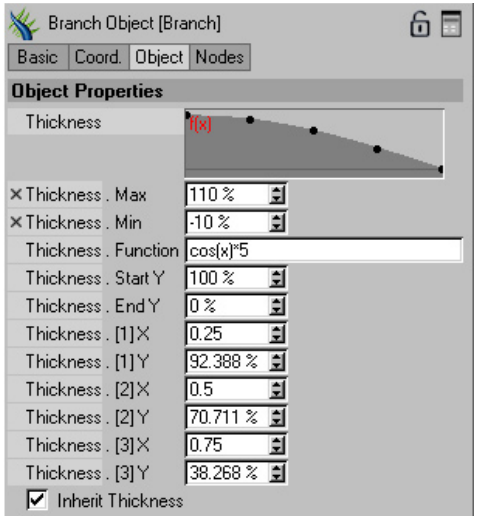
Functions can be used in Xfrog 5 to achieve some specific shapes and effects. It is possible to use functions with all parameters that are defined with a curve. Functions can be entered in the Function field in the subchannels of such a parameter. The examples below show the Thickness parameter of the Branch Object.



The pictures above show the function "id", which uses the x values defined by the Thickness curve identically as they are shown in the curve. (Identity function)

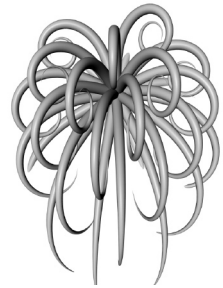


The pictures above show the function "cos", which uses the x values defined by the Thickness curve as the input for a cosine function (a graph of the cosine function is shown on the next page) and calculates the thickness of the Branch Object based on this function. The normal syntax for functions would require to insert the x, representing the value of the thickness curve: cos(x). In the case where a function is used alone and does not have any additional terms, the "(x)" expression can be omitted.

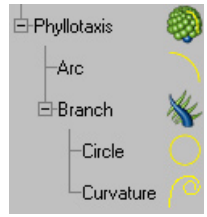


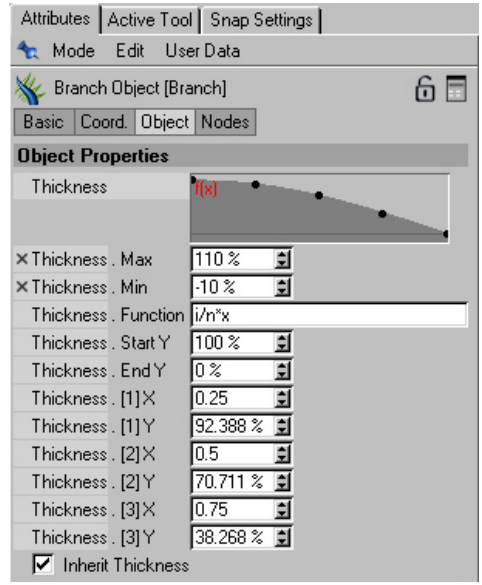
The pictures above show again the cosine function. In this expression the function uses an additional term to multiply the cos(x) by 5. In combinations of several terms it is necessary to write the whole expression of the function including the "x" parameter.

Besides the "x" variable it is possible to refer to values like the iterationnumber of Objects that are iterated by a multiplier Object ("i") or the amount of child Objects of one parent Object ("n"). Below you find an example for the use of "i" and "n" together to achieve certain modeling effects. In this example the expression i/n*x is apllied to the Rotation X parameter to produce the following form with a gradually changing curvature.



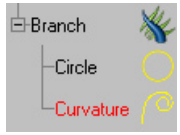
The function is applied to the Curvature Spline defining the bending of the tentacles made with a Branch Object.

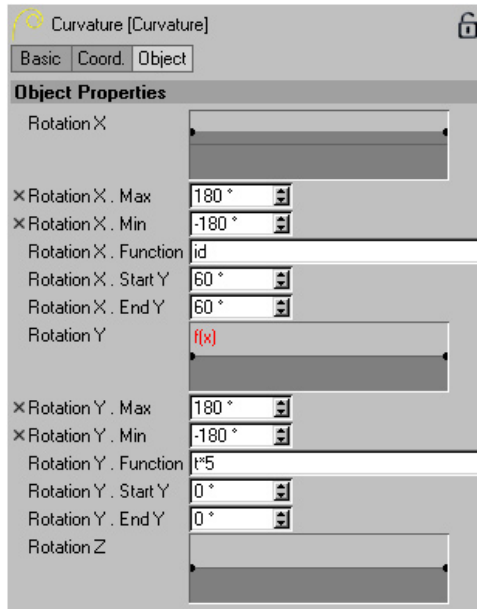




In the Animation context it is of great interest to include time into your functions. This is possible with the variable "t", refering to the time in seconds or with "f" referring to the frame number.

The Example below shows a Branch Object that uses "t" to calculate the rotation in the y-axis. In order to produce a visible effect we also set the Rotation X parameter of the Spline controlling the bending of the Branch Object.





This following shows all functions that are provided in the Xfrog 5 Plugin with their graphs and their definition-ranges. The functions can be assigned to all parameters that are defined by a curve. Functions change the calculation of the intermediate values between the control points of the curve.

A certain mathematical understanding is required to predict the result of the use of functions in a given context. Functions allow you to define shapes with a “mathematical perfection”. Certain functions are only defined within a certain range of values and deliver no useful result beyond this definition range. The following graphs will give you an idea in which way the different functions remap the values of the sliders they are assigned to and within which range of values they are defined. The Input values specify the range of values that can be passed to the function and the Output values specify the range of values that are returned by the function.

Variables:

Variables refer to certain values that can be used as the basis for further calculation. The basic variable is "x", which is referring to the values defined by the parameters curve (in our example the thickness curve). Other variables can referr to the time and allow you to integrate functions into your animations to achieve specific effects.

x – referring to the current value of the curve.

t – referring to the animation time in seconds.

f – referring to the frame number

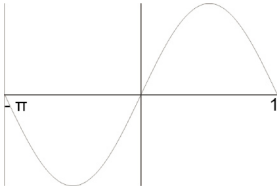
i – referring to the number of the iteration of a multiplied Object (e.g. the third instance)

n – referring to the amount of child Objects of a parent Object.

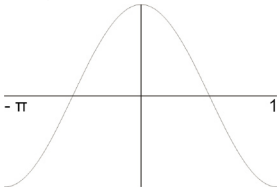
Functions:

Function: id
Description: identity
Leaves the values unchanged (y = x)

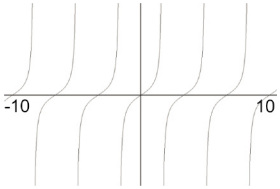
Function: y = sin x
Description: sine
Input values: -∞ to ∞
Output values: - 1 to 1



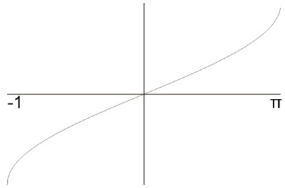
Function: y = cos x
Description: cosine
Input values: - ∞ to ∞
Output values: - 1 to 1



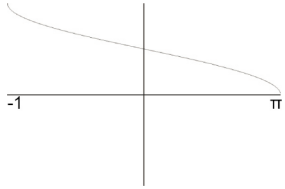
Function: y = tan x
Description: tangent
Input values: - ∞ to ∞
Output values: - ∞ to ∞



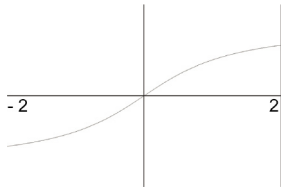
Function: y = asin x
Description: arc sine
Input values: - 1 to 1
Output values: - π to π



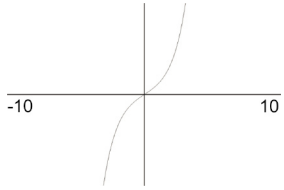
Function: y = acos x
Description: arc cosine
Input values: - 1 to 1
Output values: - π to π



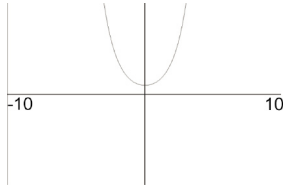
Function: y = atan x
Description: arc tan
Input values: - ∞ to ∞
Output values: - π/2 to π/2



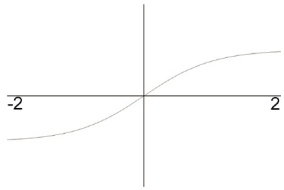
Function: y = sinh x
Description: sine hyperbolic
Input values: - 1 to 1
Output values: - π to π



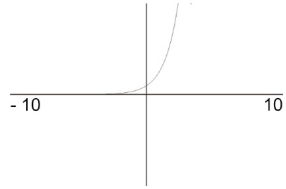
Function: y = cosh x
Description: cosine hyperbolic
Input values: - ∞ to ∞
Output values: 0 to ∞



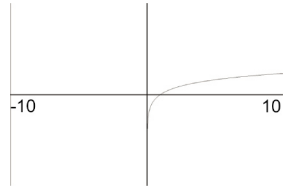
Function: y = tanh x
Description: tangent hyperbolic
Input values: - ∞ to ∞
Output values: - 1 to 1



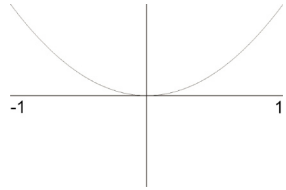
Function: y = exp x
Description: exponential
function to basis e=2.7182818
Input values: - ∞ to ∞
Output values: 0 to ∞



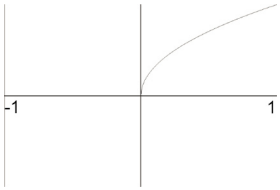
Function: y = log x
Description: logarithm to
basis e=2.7182818
Input values: 0 to ∞
Output values: - ∞ to ∞



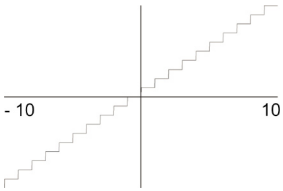
Function: y = sqr x
Description: square x (x2)
Input values: - 1 to 1
Output values: 0 to π



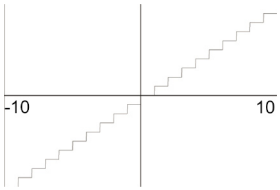
Function: y = sqrt x
Description: square root of x
Input values: 0 to ∞
Output values: 0 to ∞



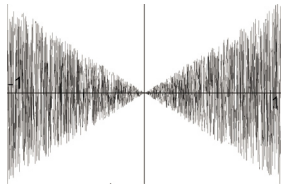
Function: y = ceil x
Description: the smallest integer greater or equal than x
Input values: - ∞ to ∞
Output values: - ∞ to ∞



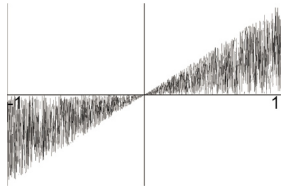
Function: y = floor x
Description: the greatest integer smaller or equal than x
Input values: - ∞ to ∞
Output values: - ∞ to ∞



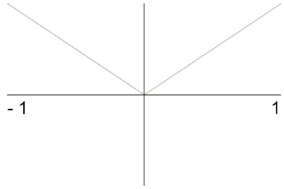
Function: y = rnd x
Description: random number between [-x..x]
Input values: - ∞ to ∞
Output values: 0 to ∞



Function: y = rndabs x
Description: x*abs(rnd(1))
Input values: - ∞ to ∞
Output values: 0 to ∞

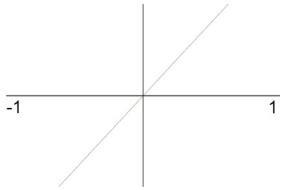


Function: y = abs x
Description: abs(x)
Input values: - ∞ to ∞
Output values: 0 to ∞

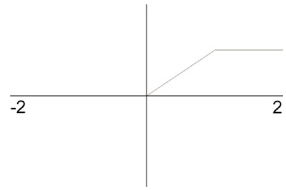


Function: y = rad x
Description: x*3.1415927/180
Input values: - ∞ to ∞
Output values: - ∞ to ∞

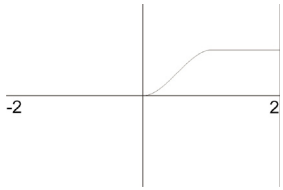
Function: y = phi x
Description: x*1.618034
Input values: - ∞ to ∞
Output values: - ∞ to ∞



Function: y = clamp x
Description: clamp in 0..1
Input values: - ∞ to ∞
Output values: 0 to ∞



Function: y = smooth x
Description: smooth clamp in 0..1
Input values: - ∞ to ∞
Output values: 0 to ∞



Function: y = pi x
Description: x*3.1415927
Input values: - ∞ to ∞
Output values: - ∞ to ∞