Xfrog 5

for MAYA
Manual 1

Xfrog
inc

Reference Manual
Xfrog 5 for Maya

Development Team

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

Usage

Xfrog 5 for Maya is subject to copyright, available as a 30 day demo and as a for-sale licensed per-manent 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.

Liability

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

Copyright Notice

Please contact sales1@xfrog.com with any inquiries.

This Reference Manual is version 1.01, created October 13 2005, by Jan Walter Schliep

Table of Contents

1. Introduction

Welcome to Xfrog 5 for Maya.

Xfrog 5 adds new Objects to Maya which allow you to model and animate organic structures and processes, for example:

- Branching, as most plants branch

- Phyllotaxis , to recreate distributions typical for flowers and blossoms

- Tropism, plants are usually pulled downward towards earth or reach towards sunlight

Xfrog integrates many years of research into a flexible and easy to use set of well integrated Xfrog 5 components.

All parameters in Xfrog 5 can be animated, and all Xfrog 5 Objects can be combined with Objects in Maya.
You can create trees or other plants which grow, which respect sun and gravity
You can create 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 Maya.

In case you were wondering about the name "Xfrog" itself, it is an acronym: "**X**-windows based **F**inite **R**ecursive **O**bject **G**enerator". Xfrog originated on SGI IRIX, and many artistic performances (realtime CAVE for example) 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. Xfrog is a unique approach to represent natural 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 20 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 through time and space (e.g. see the film "Blueberry").

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

Xfrog is not a tree or plant generator - it is much more!
Xfrog is a modeling tool wich offers you a specific toolset inside Maya to build typical organic shapes like branching structures.
If you want to work with prebuilt models, then take a look at one of the many libraries that are available from Greenworks.

1.1 What's New

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. It is also intended for the existing users of Maya to see what is added by installation of Xfrog 5.

1.1.1 Maya features include

- Windows, Linux and Mac availability
- Customizable user interface
- Multiple viewports
- Multiple cameras
- Level Of Detail
- Software and Hardware Shading
- Additional export formats
- Display measurement units
- Library browser
- Additional primitives
- Interactive editing
- Nurbs, Polygons and Subdivision Surfaces
- Complex lighting
- Curve tools
- Boolean operations
- Metaballs
- Particle systems
- Deformers, Bones
- Floor, sky and environment objects
- Elaborate transformation tools
- Inverse and multi target kinematics
- Rendering Maya Software, Maya Vector and MentalRay
- Advanced materials and shaders
- Paint Effects
- Mel Scripting

1.1.2 Xfrog 5 tools inside Maya

- CurveNurbs
- Branch object
- Phyllotaxis
- Hydra object
- Tropism object
- Variation object

1.1.3 Xfrog 5 Improvements vs. Xfrog 3.5

- 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 Maya curve can control Xfrog Objects even text curves
- Curve Nurbs offers controls similar to the Xfrog 3.5 Horn Component, while being more flexible
- Hydra offers more control
- New Variation Object allows exceptions in the iterations of multiplier Objects
- Level Of Detail offers more control

1.2 Where to find Xfrog 5

You will find elements of it in the following places inside Maya:

- Parameters for the import of models from previous Xfrog versions are located in the "Import Options" and "Open Options" dialogue of the "File menu".
- All Xfrog 5 Objects are accessible in the Xfrog 5 Shelf.
- The Xfrog 5 Objects are linked to a model hierarchy in the "Xfrog 5 Visor"
- All parameters of Xfrog 5 Objects are accessible in the "Attributes Editor"

1.3 Migrating from Xfrog 3.5 to Xfrog 5

Xfrog 5 is a plugin that integrates seamlessly into Maya. This is an answer to the demand for integration of Xfrog into popular 3D software.

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 Maya user, we bring organic modeling and animation capabilities as new Objects, which are seamlessly integrated into Maya workflow.

The paradigm shift from standalone application to integrated plugin 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 workflow has not greatly changed and you will quickly get used to the new environment.

The adaptation to both a new Xfrog and possibly a new 3D software package is a big step - a step into the future. 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. Because of the huge changes, it is not always possible to transform Xfrog 3.5 hierarchies into a Xfrog 5 hierarchy. We are continuing our efforts to improve the XFR Importer. In the meantime we recommend that you use the Maya files included on all XfrogPlants CDs and in the XfrogPlants Basic Library of 60 Trees supplied with the Xfrog 5 CDs.

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 the changes in Xfrog 5.
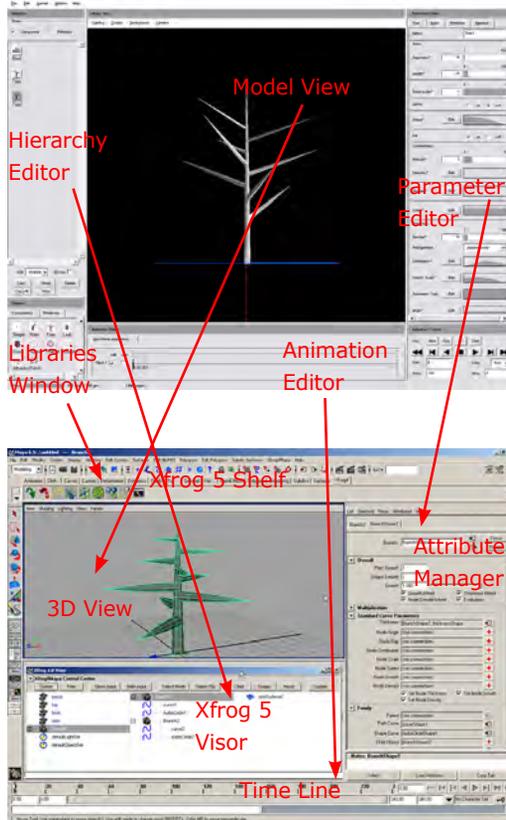
1.3.2 Object Changes

The Xfrog Components are now Maya objects, in order to better fit into the Maya naming convention. The separation of Xfrog Components and Primitives has been removed. They are all implemented as objects in Maya.

The Tree, Horn and 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 the 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 provides much more flexibility.

The PhiBall Component has become the Phyllotaxis object. We added some functionality and decided to call the object by its botanical name. The Pyllotaxis object still generates a spatial distribution of objects according to the golden section but now you can use any kind of Surface of Revolution as base 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 Maya Lattice object, which actually has the same functionality as the Xfrog HyperPatch Component.

The Variation object, which is new to Xfrog 5, allows the use 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 Maya a lot of definitions make use of curves, so we transformed the Range Definitions fa-

miliar in previous Xfrog versions - such as the Horn object curvature - into the Curve Nurbs Object. It can be combined with all Xfrog or Maya objects.

Similarly we made Tropisms available as an independent object, the Tropism object. It can be used with any Xfrog or Maya Curve.

Detailed information for all the Xfrog 5 objects is available in the Object descriptions in this manual. More information about the Lattice object can be found in the Maya documentation.

1.3.3 Modeling Changes

The method of setting up your model hierarchy is basically the same, but all linking of objects (Components) is now carried out in the Xfrog 5 Visor.

Xfrog objects are created by leftclicking the icon on the Xfrog 5 Shelf. The new Object appears inside the Xfrog 5 Visor. From here you can use the appropriate buttons (Child, Head) and so connect it to other Xfrog objects. If the newly created Object generates geometry it will immediately be visible.
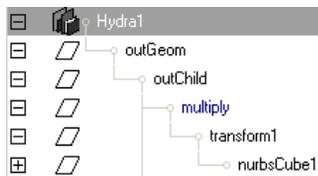
The Xfrog objects used to multiply geometry have a nested subhierarchy. This subhierarchy is used to assign certain input objects: The Objects that are iterated by the Multiplier Object are linked to the Multiplier as a subhierachy. As an example let us look at the Hydra Object multiplying a Cube.
The Cube object is linked as a child to the Hydra object.This is done by leftclicking the nurbsCube1 in the Xfrog 5 Visor, then CTRL leftclicking the Hydra and finally pressing the Child button in the Xfrog 5 Visor.

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

There is also a second possibility, to connect a child object to a parent. Leftclick the child object (nurbsCube1 in this example), CTRL leftclick the Hydra and then click the red plus sign beside Child Object (Hydra Attributes, Family settings).

Let us take 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 inside the Xfrog 5 Visor. They are both at the same level in the hierarchy and the geometry they generate falls into exactly the same place so that you will only see one trunk.

Now leftclick the Branch object that will become child object, then CTRL leftclick the future parent branch and finally click "Child" in the Xfrog 5 Visor. The image at the bottom of this page illustrates this process.

Branch2 is now linked to Branch1. The procedure of defining links is a bit different compared to Xfrog 3.5 and you have to try it a few times to get comfortable.

By the way:
If a child object should apear on top of a Branch object, then use the Head button instead of the Child button - you can find this button inside the Xfrog 5 Visor.

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 Maya to animate your Xfrog models. Every parameter can be animated independently using multiple tracks, and with the Graph Editor you can influence the interpolation between different keys. A detailed documentation of how to use the Graph Editor can be found in the Maya manual. The basic tools to set up an animation are available in the Time Manager below the 3D View.

To give you a brief overview of how to use the animation capabilities we will look at two examples: An animation of a tree growing and an animation of a tree´s 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
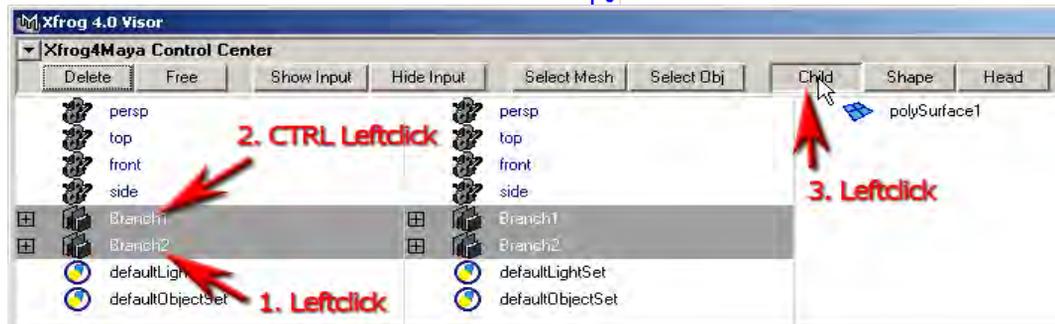
the parameters of the first Branch object. In the Xfrog 5 Visor select the first Branch Object by clicking on its name.

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

Make a right-click on the parameter name. In the menu that pops up, select "Set Key". The color of the numeric input field should change color. This automatically adds the first Branch

Object to the list of Objects that are part of the animation and adds a track for the "Growth" parameter. The first key is set and the parameter value stored in this key.

Now jump to the last frame at the far right of the Time Slider 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 (looks like Play button on your VCR or DVD player). You can find the sample file on the release CD: interface_growth_animation01.mb

Now we will try to animate the growth direction of the tree. This is controlled by a Curve object (by default it´s curve1, you will find it inside the hierarchy of the branch object).
Maya offers many different ways to animate parameters and objects, check the manual for more information.
One possibility would be to do this:

• rightclick curve1 in the 3D view and choose "Control Vertex" from the marking menu
• use the select tool to choose the points you

want to animate (perhaps you should switch to wireframe view by pressing "4" on the keyboard)
• press "s" on the keyboard to set a key
• move to a new frame in the Time Slider and use the Move tool to move the selected control vertices to a new position
• press "s" on the keyboard to set a new key

When you now press the "Play" button, you should see the points moving - and so the branch bending.
You can take a look at the example file "interface_curve_animation01.mb" on the release CD.

2 Xfrog Reference

2.1 Curve Parameter

Xfrog 5 uses a widget called Curve to control many different parameters. This control provides a curve defining a range of different values. It is used with parameters such as the thickness of a Branch Object.

To make use of a Curve to control a parameter, you first have to "activate" it. This is done by clicking the red plus icon.
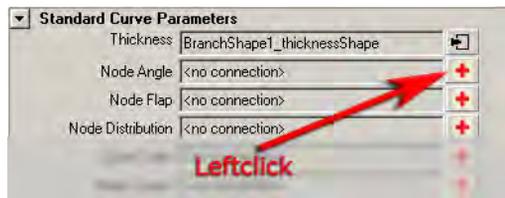
A new icon appears . Clicking this icon will open the control curve with the default settings. "Overall" contains settings to change the minimum and maximum values of the curve. It also offers a parameter called expression - here you can enter equations or functions that are used to calculate the final output of the control curve. More information on possible functions that can be used with a curve control is available in chapter 4 "Using Functions" of this manual.
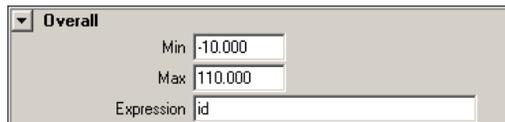

"Point parameters" contains the values of the different control points of the curve. You can change the values by entering numbers into the numeric input field, or by CTRL leftclicking/CTRL rightclicking and dragging.
It´s also possible to change the interpolation type for every point. You can choose from "Classic Smooth", "Classic Corner" (those simulate the interpolation behaviour of Xfrog 3.5) and "Bezier Smooth", "Bezier Corner" (those two offer tangents to control the shape of the curve).

For each Point X you will find at least the parameters Point X x and Point X y (height). Depending on Point X typ, you will also find fields for Point X t1x and Point X t1y - those settings control the handles of the tangent.

Point Paramters Visualization shows the control curve and it´s control points. You can select points by leftclicking and dragging a selection marquee. You can add points to the selection when you hold the Shift key pressed during this procedure, remove points by pressing CTRL - similar to most selection tools available in Maya.
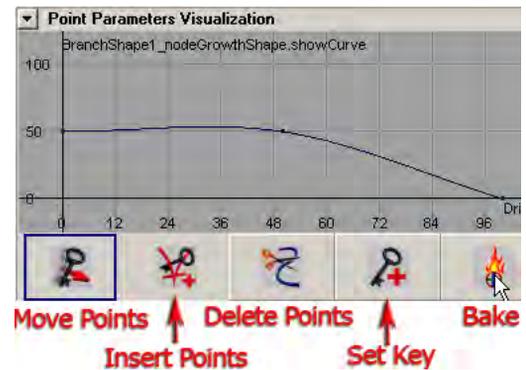
To move a control point, select it and activate the "Move Points" tool. Then drag the point to the desired position with help of the middle mouse button.

To delete points, first select them and then click the "Delete Points" icon. By selecting several points, you can delete them all at once.

To insert a new control point, you have to do the following: leftclick the control curve and turn on the "Insert Points" tool. Then click with

the middle mouse button, a new control point will be inserted at the actual position.

If you want to animate the control curve, then make use of the "Set Key" icon. Simply leftclick this button to create a key at the current frame.

If you use an expression, you can use "Bake" to create an editable control curve from the expression.

As an example of how to use the subchannels, the following paragraph shows the Thickness parameter of the Branch Object. To be precise, it shows the Point Parameters of Point_0. To open those properties, you have to leftclick the little, black arrow .
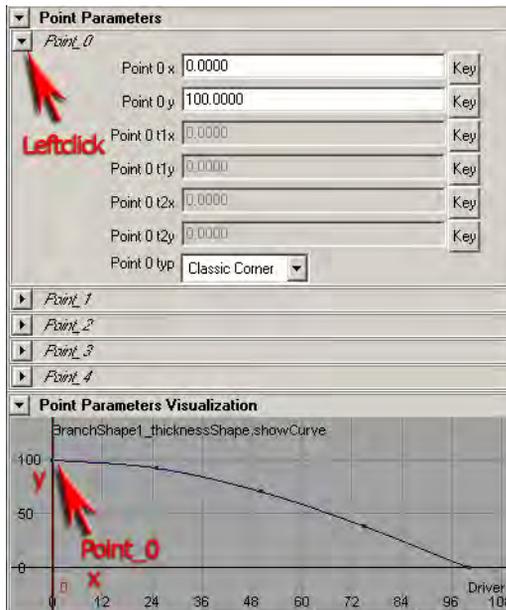
Point 0 x

Point 0 x controls the position of Point 0 along the x axis. The first point (Point 0) always has a x-value of zero - and so controls the base of the branch - the last one (in this case Point 4) has

a value of 100 and controls the top. The points in-between may use values between zero and hundred.

Point 0 y

Point 0 y defines the height. In this case this means the thickness. The higher Point 0 y, the thicker the branch will be at it´s base.

Point 0 t1x, Point 0 t1y, Point 0 t2x and Point 0 t2y

Those parameters are only active, when Point 0 typ is set to "Bezier Smooth" or "Bezier Corner". They control the position of the handles of the tangent and so have an influence on the shape of the control curve.

Point 0 typ

Point 0 typ can take several conditions: Classic Corner, Classic Smooth, Bezier Corner and Bezier Smooth.
This controls the interpolation between control points. The classic variations simulate the behaviour of Xfrog 3.5 curves. Bezier offers more control due to the use of a control tangent.

You can easily animate every control point individually. You simply have to leftclick the relevant "Key" button.

For other parameters those curve- and point-controls work exactly the same. If you want to do some fast and quick changes, simply use the control curve (Point Parameter Visualization). If you want to go with precise values, then you can make use of the above descibed Point Parameters.

2.2 Xfrog Shelf

The Xfrog Shelf gives direct access to the Xfrog Objects. If you followed the installation process described in the manual, the Xfrog 5 Shelf should be available amongst other Shelves like Animation, Curves or Rendering.
If no Xfrog 5 Shelf is created, you can do this manually by entering the "Xfrog4Maya" menu and choosing the command "Shelf".

To be able to work effectively, open the Xfrog 5 Visor by clicking the corresponding button in the Xfrog 5 Shelf. Be sure, that the Xfrog 5 plugin is loaded. If the plugin is not already loaded, simply press the green arrow in the Xfrog 5 Shelf to do so. By pressing the red arrow, you can unload the Xfrog 5 plugin.

2.3 Xfrog 5 Visor

All connections between Xfrog objects are established inside the Xfrog 5 Visor. The Visor works similar to the Outliner of Maya, but offers some additonal buttons.

With "Delete" you can delete objects - that´s easy. The "Free" button breaks the connection between active object and parent.
Primitive objects assigned as child to an Xfrog object become hidden. With "Show and Hide Input" it´s possible to make those objects visible. Select the parent Xfrog object and then press "Show Input" to show those primitives inside the 3D view.
With "Select Mesh" you can select the mesh of the active object, "Select Object" chooses the Xfrog generator nodes.
"Child" is used to create connections that Xfrog 3.5 users may know as multiple link. If you want to create a simple link - for example put

an object on top of a Branch object, then use the "Head" button.
And with "Shape" you can assign a shape to the branch object.

More details about those techniques follow in the next chapters.

2.4 Branch

The Branch creates – not surprisingly – branching structures. One Branch on its own looks in its default shape like a conical horn, similar to a trunk. This shape is achieved through the multiplication of a cross-section spline (by default a nurbs circle) which are connected to form a hull. Each cross section is scaled down sequentially along the length of the branch ending in a point.

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

To connect one Branch to a parent Branch you have to do the following in the Xfrog 5 Visor:
•leftclick the child Branch object
•CTRL leftclick the parent Branch (add select)
•leftclick the "Child" button in the Xfrog 5 Visor

A connection between both Branch objects is established and you will see the result in the 3D view.

To remove a child Branch object from the parent, leftclick the child Branch object, then leftclick the "Free" button in the Xfrog 5 Visor.

If you open the hierarchy of a branching structure inside the Xfrog 5 Visor, then you will find several objects inside the Branch object. curve1 defines the path of the Branch, nurbsCircle1 is responsible for the profile.
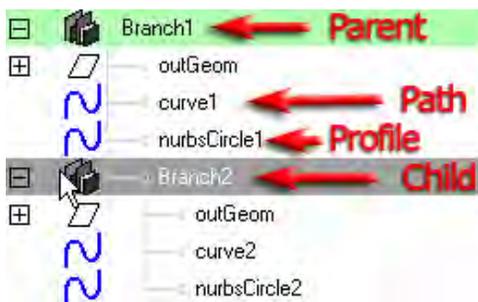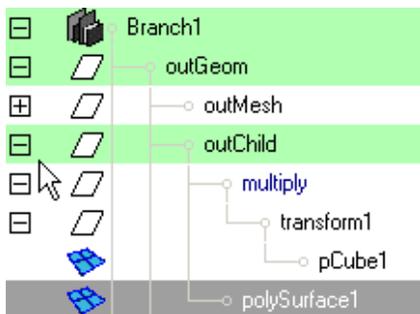
The Branch object can be used to create trees and bushes, but also may serve as stalk for a flower. If you want to create a leaf, then simply replace the profile of the Branch object (nurbs circle) by a more or less flat curve.

You are not limited to Branch objects when creating children. Almost any shape can be used.

In this example, the pCube1 object was connected to the parent branch by following the same procedure as already described earlier:
leftclick the pCube object, then CTRL leftclick the branch object and finally leftclick the "Child" button inside the Xfrog 5 Visor.

If you want to use several child objects, simply repeat those steps for every child.

If you want to put an object at the top of a Branch object - think of a blossom on top of a stalk - then you have to establish a simple link. Actually this works almost the same way as with regular child objects:
leftclick the object you want to be at the top of the branch, CTRL leftclick the desired parent (Branch) object and then leftclick the "Head" button.

Repeat this procedure to add several objects to the Head of a Branch object. The file "example_multiple_child_objects.mb" shows a structure with several child objects.

The possibility of using several child objects is very important, if you are trying to recreate realistic plants. There you will find many examples of structures, where two objects - for example a leaf and a blossom - emerge from the same knot.

In case you added several child objects to a parent branch, but only see one child, then you might need to transform the objects linked to the parent to make them visible (by default they are all generated in the same location with the same orientation).
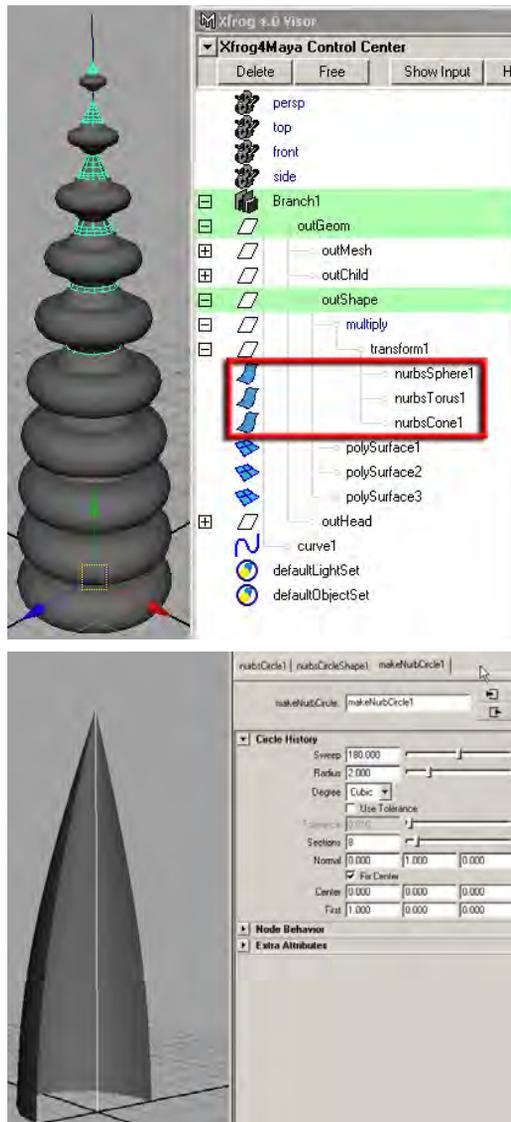
As soon as you create a control curve for a parameter, the curve will also be listed inside the hierarchy of the relevant object.

If you want to work on the profile, simply left-click the nurbsCircle in the Visor and go to the Attributes. The lower image on the left shows a branch, where the Sweep of the nurbs Circle was set to 180° (default: 360°), also the radius was changed to two.

You can use all Maya curve tools to work on the nurbsCircle. For example you could delete the History and then work on the control vertices of the profile.

If you want to delete the default profile of the Branch object, do the following:
Leftclick the profile (nurbsCircle), then leftclick the "Free" button inside the Xfrog 5 Visor.

To use your custom curve as profile for the Branch object, you have to follow these steps:
Leftclick the custom curve, then CTRL leftclick the Branch object. Now go for the "Family" settings of the branch inside the Attribute Editor. Leftclick the red plus sign of "Shape Curve", to assign the custom curve as profile.

It is also possible to use "massive" objects for the shape. This works similar to assigning a custom profile. Leftclick the desired object, CTRL leftclick the branch and go for the "Family" settings. There you have to click the red plus sign of "Shape Object". It´s also possible to assign several Shape Objects, simply repeat the above described procedure for each object that you want to use as Shape Object.

There is also an alternative and faster way, to assign objects as "Shape Object". You can make

use of the "Shape" button inside the Xfrog 5 Visor. Leftclick the custom object, CTRL leftclick the Branch object, then click the "Shape" button - the object should be used as shape from now on.
To delete an assigned Shape object, leftclick it in the hierarchy, then choose "Free" or "Delete" in the Xfrog 5 Visor.

The placement of the Shape Object(s) along the Branch depends on the "Path Curve". Objects are placed on the control vertices of the "Path Curve". So if you need more or less Shape Objects along the Branch, then you have to increase or decrease the number of control vertices of the Path Curve.

Of course you also can work on the Path Curve. Inside the Xfrog 5 Visor, open the hierarchy of the Branch object and leftclick "curve1". Now you can use Maya´s curve editing tools. You also could rightclick the curve in one of the view panels and switch into "Control Vertex" mode. This enables you to directly move the control vertices of the Path Curve.

Like with the profile, you can also use a custom curve for the path. First "Free" the default curve (or delete it with the "Delete" button of the Visor) then assign the new curve by:
Leftclick the new path curve, CTRL leftclick the Branch object, then leftclick the red plus icon of "Path Curve" inside the Family settings of the Branch object.
In many cases it´s a good idea to use curves with as few control vertices as possible. You can make use of the "Path Smooth" feature of Xfrog 5 described later on to avoid a jagged appearance.

Overall Settings
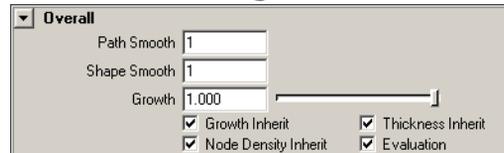
Path Smooth

"Path Smooth" controls the smoothness of your path. This enables you to work with a low resolution path curve and adjust the smoothnes of the Branch with "Path Smooth".
The default value is set to one. When increasing "Path Smooth" to a value of two, the resolution will be doubled, a value of three will triple the resolution.

Shape Smooth

"Shape Smooth" works similar to "Path Smooth" - only on the profile of the Branch object.

Growth
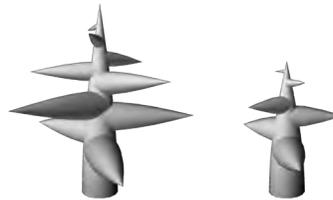
The "Growth" parameter controls the overall growth of the Branch object, the values range from zero to one. By default, "Growth" is passed to child branches. It is basically a scaling factor that influences the length of branches. The smaller "Growth", the shorter the Branch object will be and the less child objects will be produced.

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.

Growth Inherit

By default, growth information is passed from parents to child Branch objects. If a child branch shouldn´t be affected by "Growth" or "Node Growth" of parent objects, then you have to uncheck the option "Growth Inherit".
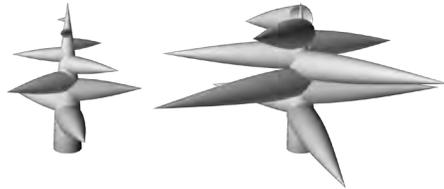
Thickness Inherit

Like growth information, the thickness is passed from parent branches to child branches. If the "Thickness" of a child Branch object should be independent from parent Branch objects, then turn off "Thickness Inherit". 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. The picture above shows a simple branching structure with Growth- and Thickness Inherit turned on (left) and off (right).

Density Inherit

The same as described above, just for the "Node Density" parameter.

Evaluation

With this option you can control, if Xfrog objects should be built or not. If "Evaluation" is turned off, then the branching structure is not generated. "Evaluation" only works on the topmost parent Branch object and turns the complete hierarchy of this parent off.

The "Evaluation" option is also important, when importing Xfrog files (*.xfr) as Xfrog 5 object. The importer translates the structure of the Xfrog file to a Xfrog 5 object. Because Xfrog objects can be very detailed and cpu intensive, "Evaluation" is turned off during import. This enables you to check the imported structure, adjust settings if needed and then turn "Evaluation" on.

"Evaluation" is also usefull, when working with several Xfrog 5 structures. You can turn off unneeded objects and so speed up the editor view.

Multiplication

Update Tesselated Mesh

By default this option is active and is typically what you need. In some special cases though, it might be useful to deactivate this option.
Try this: create a Branch object and use it to multiply a sphere. Change Multiplication Type to "Copy" and deactivate the option "Update Multiplied Transforms". Use the selection tool to activate one of the multiplied spheres, you should be able to (re-)move it. This you could use to create handmade variations without the use of the Variation object.
Update Texture does about the same, only related to textures.

Node Number

With "Node Number" you control the amount of child objects, that are produced by this Branch object. On the first sight this is a very simple parameter, but the number of child objects is also influenced by "Gowth", "Node Growth" and "Node Density".
If the "Growth" of a Branch object is set to one and the parent Branch object passes a "Node Growth" and "Node Density" of 100%, then the Branch object will exactly produce the amount of children that you specify with "Node Number".

Node Preset

This pull-down menu provides a number of presets that reflect structures of branch and leaf arrangements commonly found in nature. The specific arrangement is in many cases characteristic for a certain species. The presets provide easy access to constellations of the parameter like Node Screw etc. These parameters can also be freely defined with the controls of the Branch Object. The following passage will give you an overview over the different presets. Three parameters of the Branch Object are influenced by the Node Presets: "Node Screw", "Node Flap" and "Node Arrangement". To make your "Node Preset" decision happen, you have to press the Update button after your choice..

Perpendicular

Defines how the multiplied instances are arranged along the parent Branch Object. The instances are distributed around the parent object, with the only constraint being that they are oriented perpendicular to the axis of the parent Branch object.
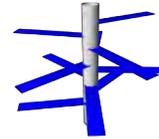
Alternate Perpendicular

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, but displaced along the axis. The orientation is again perpendicular to the axis of the parent Branch Object.
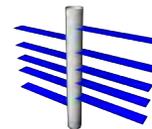
Pair Perpendicular

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 and at the same distance along the parent Branch object axis. 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 distributed manner around it. The instances are oriented parallel to the axis of the parent Branch Object.
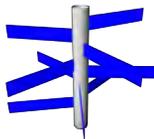
Alternate Lateral

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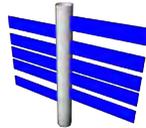
Pair Lateral

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.
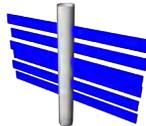
Of couse you also can work on all settings manually and create custom arrangements.

Node Arrangement

With "Node Arrangement" you can control, if multiplied instances should be placed in a ring at each node, or pairwise.

By default, "Node Arrangement" is set to "Ring" and this gives access to a second parameter: "Node Multiplicicity". With "Node Multiplicity" you can control how many instances are placed at a node. The default value is one. If you want to create arrangements typical for pine trees for example, then you can increase "Node Multiplicity" to a higher value.

The next image shows an example with "Node Arrangement" set to Ring and "Node Multiplicity" set to five.
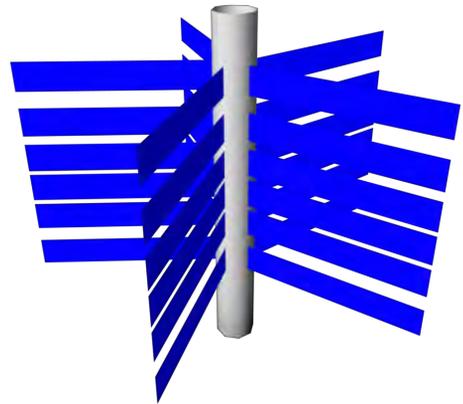
If you set "Node Arrangement" to "Pair Mirror", then you always will get two instances per node. There´s no acess to the "Node Multiplicity" parameter in this case. "Pair Mirror" is very typical for ferns for example and also for palm fans.

Multiplication Type

"Multiplication Type" let´s you choose from three different options.
Tesselated Mesh is the default setting and means, that a tesselated mesh is generated from the Xfrog hierarchy.
Instance creates instances from one child object, and Copy creates copies. This parameter is mainly important when creating a mesh from the Xfrog hierarchy or when working with "non-typical" child objects.
If you want to multiply a point light for example, then you have to change the "Multiplication Type" to Copy or Instance, otherwise you will get an error message. Don´t forget to press the Update button after switching the "Multiplication Type"!

Try this: create a Branch object, set "Multiplication Type" to "Instance" and press the "Update" button. Now create a point light. Leftclick the point light in the Visor, CTRL leftclick the branch and press the "Child" button. The point light should appear several times inside the branch. Now leftclick the Branch object in the Xfrog 5 Visor and press the "Select Obj" button to select the object, then press the "Delete" button. This makes an editable object out of the procedural Xfrog structure. You should get the mesh of the branch and inside several point lights. When you now choose one of the point lights and change it´s parameters, then all others will change too - because they are all instances. If you repeat the same procedure but with "Multiplication Type" set to "Copy", then the lights will be independent from each other.

Standard Curve Parameters
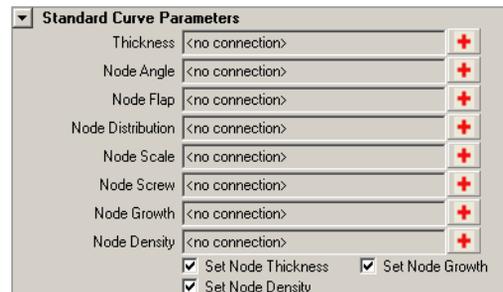
The Standard Curve Paramteres contain properties that are controlled with curves, already described earlier in this manual.
By default, no curves are present. To create a control curve for a parameter, you have to left-click the red plus icon.
A new icon should be visible it represents an "Input Connection". If you click this button, then you will get access to the control curve. Alternatively you can take a look at the Xfrog 5 Visor. After creating a control curve, it should be visible in the Visor too. Simply leftclick the entry in the Visor to get acces to the controls.
This procedure is the same for all other Curve Parameters of Xfrog.

Thickness
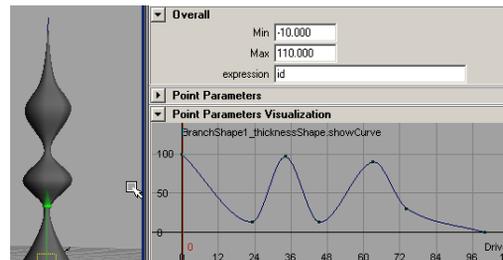
The Thickness parameter is controlled with a Curve Parameter. This control provides a curve defining the thickness of the Branch object. It

also defines a scaling factor for the segments that are multiplied along the Branch object. The curve represents the length of the Branch object. The left side of the curve controls the base of the Branch object. The part in the middle has an influence of the middle of the branch and the right part of the curve has an influence on the upper area of the Branch object. More information about the Curve Parameter widget is available in chapter "2.1 Curve Parameter" at the beginning of this manual.

The picture below shows the control curve of the "Thickness" parameter and the resulting Branch object.

It is also possible to define the thickness of a Branch Object by editing the Radius of the nurbsCircle inside the subhierarchy of the branch. In this setting the thickness information is not transmitted to the subsequent Branch object in the hierarchy. Thus the subsequent branches do not react to the thickness of their parent Branch Object.

Node Angle

This parameter defines the angle in which a child object grows out of the parent branch.

The curve displays the values over the length of the parent Object. If you need better control over the angle, add more control points to the curve.
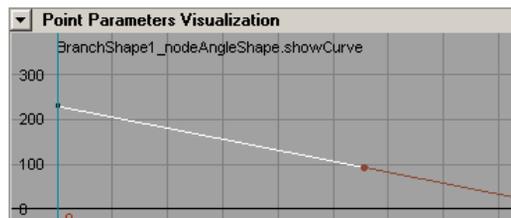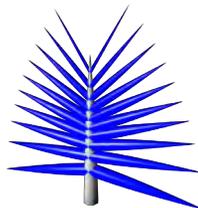

Node Flap

With "Node Flap" you can specify the rotation of multiplied objects around their length axis.

The picture on the right shows an example, where the parent Branch object uses "Node Flap" to rotate the child branches. The curve represents the values for the different iterations along the parent Branch object. This parameter is also used to create arrangements like lateral or perpendicular when working with Node Presets.
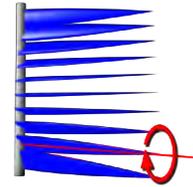
Node Distribution

Specifies the distribution of child objects 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
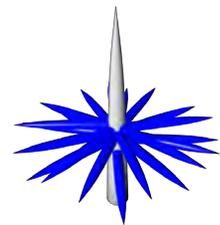
The following picture shows the Distribution curve that was used to create the accumulation of branches at the center of the parent Branch object.
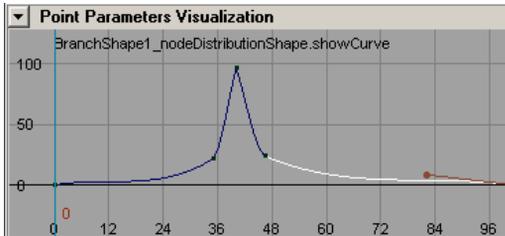
the length of the child branches. The curve displays the values over the length of the parent Object. "Node Growth" also has an influence on the number of grandchild objects, that are generated by the children of the parent Branch object. The following picture shows an example with high values at the base of the parent Branch object and lower values in the upper areas.

Node Scale

"Node Scale" controls - what a surprise - the scale of child objects. This is of course completely different to the growth parameters. "Node Scale" always works on the complete geometry of the child objetcs.

Node Screw

Rotates the nodes around the axis of the Branch Object. The curve represents the values for the different iterations along the parent Branch Object.


By default Xfrog tries to recreate a typical arrangement found in nature. An important keyword is "Golden Section".

The "Golden Section" has a value of about 1.62. Proportions that follow the ratio of the Golden Section are often considered as aesthetically pleasing. Many distributions inside the world of plants can be described with this number.

Node Growth

This parameter defines a growth factor for the different multiplications of the branches which is similar to the Growth parameter of the "Overall" settings. The "Node Growth" mainly influences
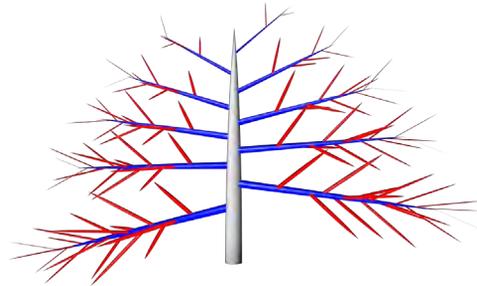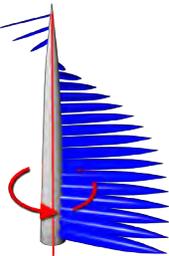
Node Density

This parameter controls how many branches are produced by the child Branch object linked to the currently selected object. Imagine you are applying the Node Density parameter to the trunk of a tree. You can specify for exam-

ple 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.

The picture shows the same branching structure as seen in the "Node Growth" paragraph. But this time the "Node Density" of the parent Branch object is set to higher values in the upper area.
Because of the higher density values, the second branching level (blue color) produces more grandchild objects (red color).

Set Node Thickness, Set Node Growth, Set Node Density

These three options can be turned on and off, by default they are turned on.
When turned off, the relevant parameter is not passed to the child Branch objects. You can use this to override the control curves that you created**.**

Family

The Family settings mainly contain information about curves that are used for path or profile purpose. Here you also can assign new child objects or head objects.
In many cases you will be able to connect objects inside the Xfrog 5 Visor. But some procedures have to be done manually with help of the Family settings.
Below you will find a list of the different options and what they can do for you.

Parent

This shows the parent object, of course only if existent.

Path Curve

This option contains the link to the Curve object that defines the path of the Branch object.

Shape Curve

Similar to the above parameter, establishes the link to the curve responsible for the shape (profile of the branch).

Child Object

With this parameter you can create connections between a parent and child object. If you don´t use the easy way inside the Xfrog 5 Visor, then follow these steps:
Leftclick the future child object, CTRL leftclick the future parent object, then leftclick the red plus icon of the "Child Object". The connection between both objets should be established.

Shape Object

Additionally to the "Shape Curve", you also can make use of an object to create the shape of the Branch object. Leftclick the desired object, CTRL leftclick the Branch object and finaly leftclick the red icon of "Shape Object" inside the Family settings to assign the object.
Objects that are made "Shape Object" are placed accordingly to the knots of the path curve. It´s possible to use several "Shape Objects".

Shape Primitive

Contains the Transform node of the relevant "Shape Object".

Head Object

This option let´s you place objects at the top of the Branch object. You can use any kind of object, just be sure to use the proper Multiplication Type settings.

Head Primitive

Contains the Transform node of the relevant "Head Object".

2.5 Phyllotaxis
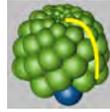
 The Phyllotaxis Object distributes Objects on arbitrary surfaces of revolution. Its name comes from the botanical term "phyllotaxis". It describes the arrangement of plant organs, such as seeds in a flower.
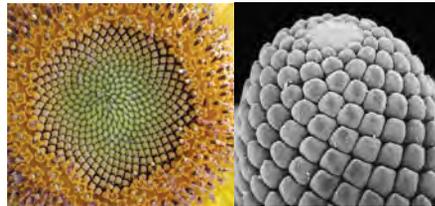
Some examples of phyllotaxis

By default it multiplies Objects that are linked to it and distributes on the surface of a sphere.
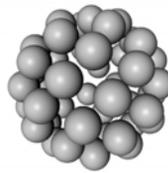
 The Object provides a number of parameters that control this arrangement so that the Object is useful for many different structures that are based on a spatial distribution on a surface of revolution. Similar to the Branch Object the positions in the subhierarchy fulfill specific functions:
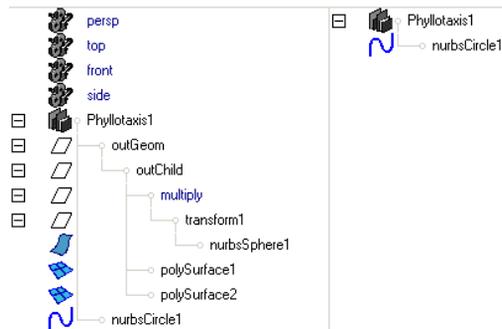
Inside the subhierarchy of the Phyllotaxis you can find a nurbsCircle object defining the surface of revolution. To create many different kinds of surfaces of revolution any Curve can be assigned.

By default the nurbsCircle is set to a Radius of one and Sweep of 180°. This defines an Arc which specifies the area of the spherical surface that is covered by iterated instances.
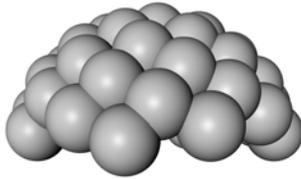
The picture above shows a Phyllotaxis with a Sphere as child object. The Sweep parameter of the nurbsCircle - you can find the parameter inside the makenurbsCircle settings - is set to a value of 90°.

Overall Settings
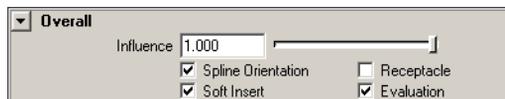
Influence

Specifies how strongly the position of the iterated instances are influenced by their size. When, for example, the first instances are scaled down and the others remain at their initial size the surface appears more packed in the area where the big instances 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 instances 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.
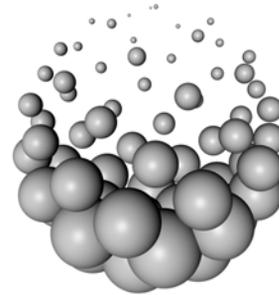
Spline Orientation

As soon as this option is active, child objects will be oriented accordingly to the (virtual) surface normal of the surface of revolution.
If inactive, then it is possible to make use of Curve parameter "Orientation", discussed a little bit later.

Soft Insert

This option is important for the animation of a Phyllotaxis Object. When the "Node 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 full size when they are inserted and scaled down from their normal size to zero when they are removed.
When the Soft Insert option is disabled new instances pop in at their full size or existing ones are removed instantly.

Evaluation

Similar to the Branch object, you can turn the evaluation of the Phyllotaxis on or off.

Multiplication

This section controlls the number of child objects and of what type the resulting object will be.

Update Tesselated Mesh

By default this option is active and is exactly what you need. In some special cases though, it might be useful to deactivate this option.
Try this: create a Branch object and use it to multiply a sphere. Change Multiplication Type to "Copy" and deactivate the option "Update Multiplied Transforms". Use the selection tool to activate one of the multiplied spheres, you should be able to (re-)move it. This you could use to create handmade variations without the use of the Variation object.
Update Texture does about the same, only related to textures.

Node Number

This parameter defines the number of child objects.

Multiplication Type

"Multiplication Type" let´s you choose from three different options.
Tesselated Mesh is the default setting and means, that a tesselated mesh is generated from the Xfrog hierarchy.
Instance creates instances from one child object, and Copy creates copies.

Standard Curve Parameters

Orientation

By default child objects will be oriented according to the (virtual) surface normal of the surface of revolution of the Phyllotaxis.
When this option is inactive, then you can make use of the curve parameter "Orientation".
"Orientation" specifies a rotation value for the instances. The leftmost point corresponds to the first instance and the rightmost point to the last instance. The following illustration shows an example of the Orientation parameter and the resulting shape:
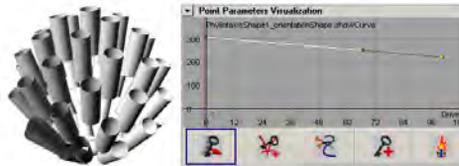
Node Angle

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.
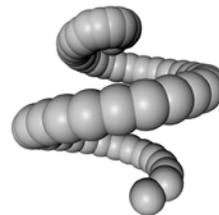
Node Scale

This parameter allows you to specify a scaling factor for the multiplied instances. The leftmost point corresponds to the first instance and the rightmost point to the last instance.
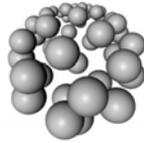
Node Growth

This is one of the new features of Xfrog 5. If you are using Branch objects as children of the Phyllotaxis, you can influence them with the "Node Growth" parameter.
The leftmost point corresponds to the first instanceand the rightmost point to the last instance.

Set Node Growth

An additional option that you can use to turn Node Growth on or off.

Family settings

Parent

Shows parent object, if existent.

Path Curve

Shows the Curve that is used to define the surface of revolution.

Child Object

Contains the link to the child object.

2.6 Hydra

An Object linked to the Hydra Object is multiplied and distributed in a circle. Creating connections between parent and child works similar to Branch and Phyllotaxis:
Leftclick the child, CTRL leftclick the Hydra object then leftclick "Child" in the Xfrog 5 Visor.

The Hydra Object can multiply any object, for example also other Xfrog Objects. Below you can see a Hydra with a Branch object as child. The Branch object uses cubes for the shape.
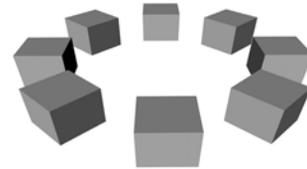
Overall Settings
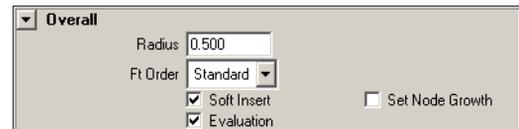
Radius

With "Radius" you can control the dimensions of the virtual circle in which child objects are arranged.

Ft Order

"Ft Order" offers two possibilities, "Standard" and "Classic".
If you want to simulate the behaviour of the Hydra inside Xfrog 3.5, then you have to switch to "Classic".
There is a slight difference in the behavior

of the Twist parameter, when you compare "Standard" to "Classic". The Standard behaviour creates more predictable results while 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.

Soft Insert

This option is important for the animation of a Hydra Object. When the "Node Number" parameter changes over time , new instances are "faded in" or existing instances are "faded out". When the Soft Insert option is disabled, new instances pop in at their full size or existing ones are removed instantly.

Evaluation

Turn the Hydra on or off.

Set Node Growth

As soon as this option is active, you can use the "Node Growth" parameter to control subsequent Branch objects.

Multiplication

Update Tesselated Mesh

By default this option is active and is typically what you need. In some special cases though, it might be useful to deactivate this option.
Try this: create a Branch object and use it to multiply a sphere. Change Multiplication Type to "Copy" and deactivate the option "Update Multiplied Transforms". Use the selection tool to activate one of the multiplied spheres, you should

be able to (re-)move it. This you could use to create handmade variations without the use of the Variation object.
Update Texture does about the same, only related to textures.

Node Number

With "Node Number" you control the number of child objects, that are produced by the Hydra.

Multiplication Type

"Multiplication Type" let´s you choose from three different options.
Tesselated Mesh is the default setting and means, that a tessalted mesh is generated from the Xfrog hierarchy.
Instance creates instances from one child object. When you "bake" the Xfrog object, then you will get the proper number of child objects as instance from the original child object. When you change one of them, then all others will receive the same changes.
When set to Copy you will get separate duplicates of the original child object. This time the copies are independent from each other.
If you want to multiply a point light for example, then you have to change the "Multiplication Type" to Copy or Instance, otherwise you will get an error message.
Don´t forget to press the Update button after switching the "Multiplication Type"!

Standard Curve Parameters

Fan

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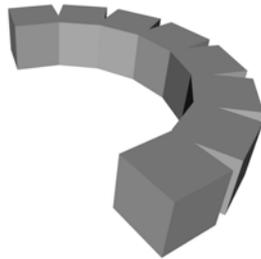
Twist

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.
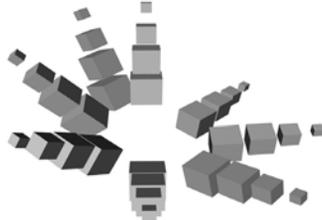
Spin

Specifies a rotation of the instances around their local longitudinal axis. The leftmost point corresponds to the first instance and the rightmost point to the last instance.
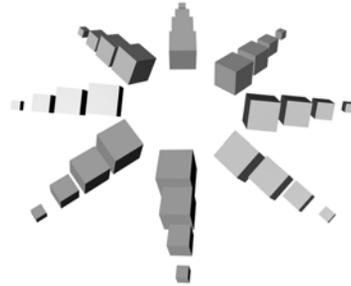
Scale

Specifies a scaling factor for the instances. The leftmost point corresponds to the first instance and the rightmost point to the last instance.
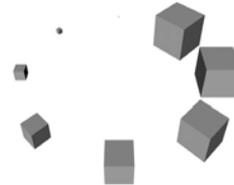
Node Growth

The growth of subordinate Branch objects can be controlled with this curve parameter.
The leftmost control point controls the growth of the first iteration, the rightmost control point specifies the growth of the last instance.

Family Settings

Parent

Contains the parent object of the Hydra object.

Child Object

Here you can create parent - child connections.

2.7 CurveNurbs

The CurveNurbs Object is a new curve Object introduced by Xfrog. The difference to other curve objects, is that it offers a set of parameters to provide controls that you might know from Xfrog 3.5 Horn components. With the CurveNurbs Objects parameters the curvature along the curve is defined.The curved spline is created step by step and for each of these steps a rotation and translation is defined. Mathematical functions can also be used to influence the curvature.

The CurveNurbs Object can be used with any Xfrog or Maya object that can make use of 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.

Overall
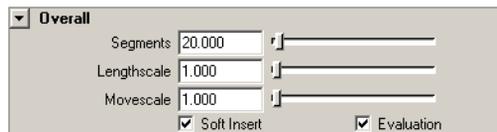
Segments

Defines the amount of Control Vertices of the CurveNurbs Object. This parameter determines the shape of the CurveNurbs. 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 Path Smooth option to the CurveNurbs (see Branch object, Overall settings) offers a much more efficient way to create a smoothly curved object. 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 make use of the Path Smooth option of the Branch object.

The above image shows a Branch object with a CurveNurbs as path. In both cases the number of "Segments" of the CurveNurbs is the same, but on the right side the "Path Smooth" option of the Branch object was used to smooth the appearance.

Lengthscale

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

Movescale

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 comes from the natural growth processes of plants. As an ex-

ample 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 segments of the Curvature Object longer and the angle between the different segments steeper.

Soft Insert

Ensures that Branch objects using a CurveNurbs are added smoothly, when the number of branches is animated.

Evaluation

You can turn the evaluation of the CurveNurbs object on or off.

Standard Curve Parameters

X-Rotation

Defines the rotation around the X-axis for the segments the CurveNurbs object. The beginning of the curve corresponds to the beginning of the CurveNurbs, the rightmost part of the control curve corresponds to the end.
Example: If all control points of the curve are set to a value of 314 (which stands for PI) , each segment is curved so that over its whole length the Spline bends 180 degrees (which is the same as PI wich means a half circle).

Rotation Y

Similar to Rotation X but applied to the Y-axis.

Rotation Z

Similar to Rotation X but applied to the Z-axis.

Translation X

Defines a tranlation (move) along the X-axis of each CurveNurbs segment. The left part of the curve corresponds to the base of the CurveN-urbs object, the rightmost part of the curve corresponds to the end.
Example: A constant value of 2000 units means that each segment is moved so that over the whole length of the CurveNurbs object the segments add up to a distance of 2000 units.

Translation Y

Similar to Translation X but applied to the Y-axis.

Translation Z

Similar to Translation X but applied to the Z-axis.

2.8 Variation

The Variation object is used with multiplier objects like Branch, Hydra or Phyllotaxis. It allows you to multiply different subobjects with the same multiplier object. The object only works together with Xfrog Multiplier objects.

The Variation object is
linked to the Multiplier object and the objects that are to be multiplied are linked as children to the Variation object.
Linking a Variation object to a multiplier like the Branch object works the same way as already described before: leftclick the Variation object, CTRL leftclick the future parent and leftclick the Child button.
Now you can link several child objects to the Variation object.

Important note:
To link a future child object to the Variation object works the same way as described above
- almost. If you link several objects to the Variation object and you use the Child button every time, then you will create a group of objects. Sometimes this will make sense, but in most cases this is not what you want.
If the Variation object should be possibe to choose between several objects - to create variations - then you have to add the second, third and the nth child object manually. This means that you leftclick the child object, CTRL leftclick the Variation object and then go for the Family settings of the Variation object.
There you have to manually assign the child object by leftclicking the red icon of "Child Object X". By default there are two slots available, if you need more, then you have to change the "Child Count" inside the Multiplication settings.

Overall Settings
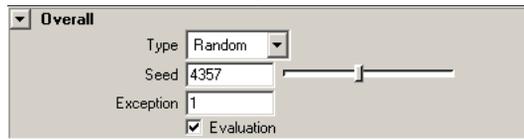
Type

Random

Here you can choose out of several different options. If set to "Random", the Variation object will choose randomly out of the child objects linked to the Variation object. Think of a tree - you could distribute healthy and some dead leaves over the branches. If you want that more healthy leaves appear, then link several healthy leaves to the Variation object, but only one dead leaf. If Type is set to "Random", then you can make use of "Seed" to change the random seed.

Sequential

If Type is set to "Sequential", then the child objects will be repeated in the same order that they were linked to the Variation object. On the right you can see a repeating Sequence of a cube, torus and sphere.

Spread

If Type is set to "Spread", then the instances will be distributed evenly. The picture below shows an ex-

ample of Type set to Spread. This is a helpful setting, when you need older branches at the base and the youngest and most green branches at the top.

Exception

The fourth option of Type is "Exception". Exception allows you to define one specific exception for one of the instances where the second of the Subobjects is used. The exception is defined numerically referring to the number of the multiplication. The example shows a Hydra with Variation object as child. The first child object of the Variation object is a sphere, the second one - the exception - is a torus. If Type is set to "Exception", then you can use the numeric input field to specify the instance wich should be replaced by the exception.

2.9 Tropism
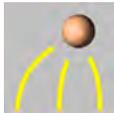
The Tropism Object allows you to define force fields to bend curves into a certain direction. The Tropism Object can be used, for example, to bend curves towards a (fictitious) light source. This phenomenon is called phototropism. Likewise you can define a gravitropism which would cause the branches of a tree to hang down towards the ground as if they were attracted by the gravitation of the earth.

The Tropism has to be connected to the curve that it should deform. The example on the left shows a Tropism Object used to bend the branch object along the Z axis. Below you will find the hierarchy of the branch object. You can see the Tropism inside the Curve object.
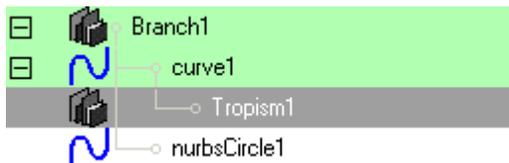

The default setting of the Tropism Object directs the branches along the Z axis.

To connect a Tropism object to a curve, you have to follow this procedure:
Leftclick the curve (inside the Branch), CTRL leftclick the Tropism object. Then go for the Family settings and click the red plus icon of "Curve" to assign the Tropism object to the curve.

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
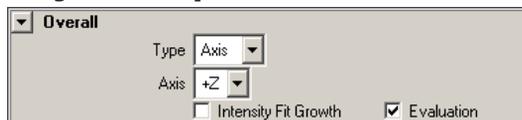
Type

This pulldown menu allows you to switch between two definitions of the force field.
Type: *Axis*
Turns the deformed curve in direction of one of the three axes of the coordinate system. When you use the Axis setting together with a reference Object, the curve 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.
To assign a Reference object please do the following: leftclick the future Reference object, CTRL leftclick the Tropism object, then click the plus icon of "Reference" inside the Family settings of the Tropism object.
Type: *Point*
Bends the deformed curve towards the position of another Object that you can specify as "Reference" inside the Family settings of the Tropism object. This option allows you to use a light source existing in your scene as the point

towards which the Spline turns.

Axis

This pulldown menu allows you to select the axis and direction into which the curves turn when the Tropism Type is "Axis". The default value is +Z which bends the Spline in the positive direction of the Z-Axis. In order to define a gravitropism, select -Y to cause the Spline to bend towards the ground.

Intensity Fit Growth

This option can be used to adjust the behaviour of the Tropism object. If activated, the effect of the Tropism object on branches with lower Growth values is higher.

Evaluation

Use the checkbox to turn the evaluation of the Tropism object on or off.

Standard Curve Parameters

Intensity

This control curve can be used to control the strength of the Tropism object. If you use negative values for the control curve, then the Tropism object acts into the oposite direction specified in the "Axis" pulldown menu.
By adding more control points, it´s possible to bend branches down at the base and back upwards towards the end of the branches.

Family Settings

Parent

The parent object of the Tropism is displayed in this field.

Curve

Use this slot to assign the Tropism object to a curve.

Reference

This field is used to define an arbitrary object as a reference for the force field. To do so, leftclick the reference object, CTRL leftclick the Tropism and leftclick the icon of "Reference" inside the Family settings of the Tropism object. If Type is set to "Point", the curves are bent towards the position of the reference object. If Type is set to Axis the branches are bent towards the respective axis of the reference object.
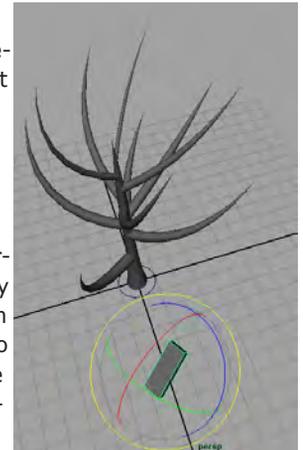
2.10 Import Settings
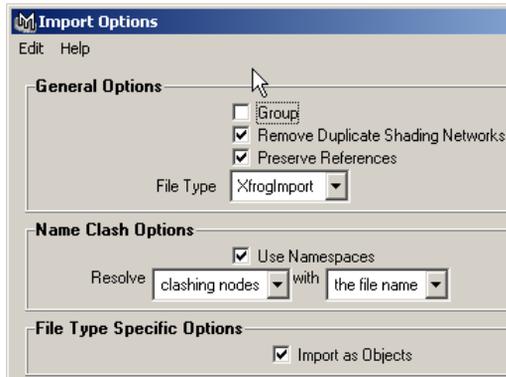
Xfrog 5 also adds an option to the file dialogues of Maya.

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 Maya. However, as the model hierarchy is handled in a slightly different way in Maya, you can control the way how the Xfrog 3.5 hierarchy is interpreted and transformed into a Xfrog 5 hierarchy.

When you open the "Options" during opening or merging scenes, you will see the "Xfrog Import" options.
If the option "Import as Objects" is active - you can find it inside the section "File Type Specific Options" - then the importer tries to translate the Xfrog 3.5 hierarchy into a Xfrog 5 hierarchy. Because Xfrog 5 is a complete rewrite it works very different from the classic Xfrog standalone softare, therefor this translation won´t work in all cases.
If the option "Import as Objects" is unchecked, then Xfrog files are importes as a mesh. If you don´t want to change the model, then this will be the better solution.

Important note:

After importing Xfrog files as Xfrog 5 object, the hierarchy will be visible in the Xfrog 5 visor, but the model won´t be visible in the 3D view. This is because the parent Xfrog object is turned off by default!
Simply leftclick the parent Xfrog object, go for the "Overall" settings and activate "Evaluation" !

The Xfrog import options only will be available, when the plugin is loaded!

3. Xfrog 5 and Maya

This chapter will provide some tips on how to make the best out of the combination of Maya 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.
Maya has a Level Of Detail (LOD) System built in and the Xfrog Objects can make use of this. You could build several copies of a tree for example, each with it´s own resolution - one for close ups, one for medium distance and a third for the background for example.

Level of Detail Group

The easiest way is to build a low resolution structure and then use the "Smooth Shape" and "Smooth Path" options to create higher resoluti- on copies. The highres tree could also have many small leaves, whereas the low resolution tree has less leaves but with bigger size - this way the overall "volume" keeps more or less identical.
When you build the different resolutions, simply select all objects and choose the command "Edit->Level of Detail->Group" from the main menu.

A lodGroup is created. The first object in the hierarchy should be the detailed and high reso- lution object. The next should be less detailed and so on.
When now navigating inside the scene, always the appropriate Xfrog object is displayed. Are you close to the tree, then the LOD algorithm will choose the high resolution tree. If the tree is only visible in the far distance, then the low resolution tree will be displayed. It´s even pos- sible to set a distance from which the object is completly turned off.
Therefor leftclick the lodGroup in the Visor and go for the "Lod Attributes". Activate the option "Min Max Distance" and enter the desired va- lues in the numeric input fields below. If you are further away from the Xfrog object then specified in the Max settings, then the object will be turned off completely.

So as you can see, it´s very simple to combine Xfrog 5 objects with Maya´s LOD capabilities. The Level of Detail is set accordingly to the ca- mera distance.
Please also take a look at the example file "LOD_LODgroup01.mb"

If the Xfrog object is not animated, then it makes sense to create static meshes. This nor- mally will increase the display speed in the 3D views.
The procedure could look like this:
Leftclick the toplevel Xfrog object, then click the "Select Obj" button in the Visor. Now simply click the "Delete" button inside the Xfrog 5 Vi- sor and you will get your static mesh.

When working with Subdivision Surfaces as child objects of a Xfrog component, then it´s also possible to easily change the resolution. Head for "Subdivision Surface Display" of the child object and change the "Resolution" to the desired level. After that leftclick the Xfrog parent object, the display should be updated

with the new resolution of the subdivision child. Please keep in mind: this only works, if "Multiplication Type" of the Xfrog object is set to "Instance" or "Copy".

3.2 Detail Editing of Objects
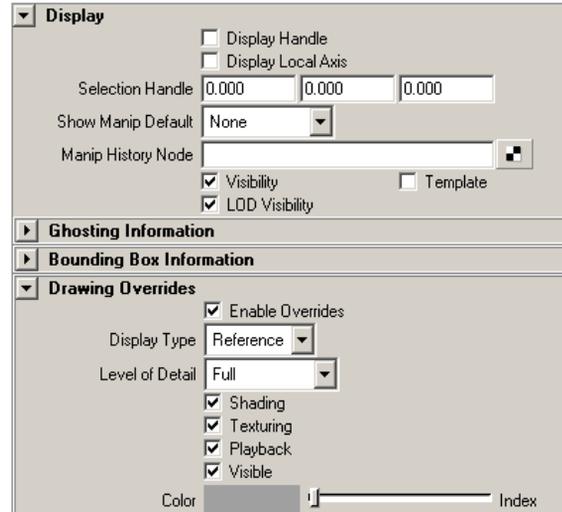
The great power of Xfrog lies in its capability to control complex structures with a small number of parameters. When you create a tree you are not dealing with every branch separately but controlling the entire branching level with one set of parameters. This allows of course for a quick and simple 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 create a static mesh from the Xfrog object. Therefore you have to follow this procedure:
Leftclick the toplevel Xfrog object, then leftclick "Select Obj" inside the Xfrog 5 Visor. The Xfrog generator nodes of the object are selected. Now simply click the "Delete" button of the

Visor, the remaining object only contains the geometry that was formerly generated by the Xfrog generator nodes.

If you now want to work on the polysurface on vertex or face level, then you have to take a look at the Drawing overrides.

Select the desired polysurface object and open the "Display" attributes. Inside "Display" you will find "Drawing Overrides". The "Display Type" of this parameter set is set to "Reference". Use the pulldown menu to change "Display Type" to "Normal".

After this change you should be able to use all appropriate modeling tools of Maya to work on the shape of the (former) Xfrog object.

As the Object looses its parametrical definition through this conversion you loose the possibility to edit the Object through the Xfrog parameters. This is an irreversible process. So be sure you are done with your Xfrog work before you switch to a static mesh.

3.3 Xfrog 5 and Paint Effects

Because of the tight integration of Xfrog 5 into Maya, it is possible to combine Xfrog 5 objects with Paint Effects.

For example you could create the stem and branches with a Xfrog 5 branching structure and then use Paint Effects to add leaves to the tree. The following section will show, how to add Paint Effects to a Xfrog branching structure.

Step 1

At first you should start building the tree. So add several Branch objects to the scene and use the standard procedure to link them together: leftclick future child object, CTRL leftclick the future parent object and finally click the "Child" button of the Xfrog 5 Visor.

Use parameters like "Thickness", "Node Angle", "Node Growth" and "Node Distribution" to shape the tree. When you are done, you have to set the "Multiplication Type" of the branch that will multiply the Paint Effetcs strokes to "Instance".

Step 2

Create a Paint Effects stroke near the origin of the scene. Then make the stroke parent of the stroke curve - by dragging the stroke curve with middle mouse button on top of the stroke (inside the Visor). Now leftclick the stroke, CTRL leftclick the Xfrog multiplier (Branch, Hydra, Phyllotaxis) and press the "Child" button inside the Xfrog 5 Visor. Please keep in mind, this only works if "Multiplication Type" is set to "Instance"! The creation of Xfrog structures is a very complex task and similar is true for instances inside Maya, especially when it´s instances from Paint Effects strokes. So try to keep the complexity of objects in a reasonable range.

3.4 Xfrog 5 and Visibility

Normally primitive objects assigned as child to an Xfrog object become hidden and work further only as source for geometry. To edit them you need to make them visible again. This can be done in the visor. If you select an Xfrog object, in the right column all primitive geometry-sources appear. By selecting one of them, its attributes appear in the attribute-editor, where visibility can be toggled.

This whole process can be done easier using the "Show Input" button of the Visor. Select the object, which primitive child should be visible for editing and then make them visible using the Show Input-button. To make them disappear again after editing use the Hide Input-button.

It is also possible to work with layers that will override the internal visibility-settings. Best you will add the primitive to a layer, before you assign it as a child to an Xfrog object. After the assignment, the primitive will also disappear, but after re-enabling its visibility using the Show Input-button, visibility is steered by the layer. This way you can control the visibility now by the layer-settings.

The same is valid for the created geometry of Xfrog objects. Often they disturb while editing the primitives. You can easily select the top-level Xfrog object, afterwards use the Select Mesh-button to select just the created geometry. After that you can add all selected objects to a new layer. This layer steers the visibility of the Xfrog objects created geometry, while the first layer controls the input-primitives-geometry. This way it is possible to visualize just the components you need for editing, to have a smoother access to the input.If you have your Xfrog object with already assigned primitives, it is also possible to make the input-objects visible using the described way and then to add them to a separate layer later.

3.5 Xfrog 5 and MEL Scripting

In this section you will find some examples, how you can create Xfrog structures with the help of MEL.

In the last chapter of this manual, you will find a complete list of MEL commands supported by Xfrog 5.

You will notice, that even complex examples are easy to realize using Xfrog-MEL-commands, which integrate well in the Maya-workflow. Also standard-MEL-commands may be used. If the standard-MEL-commands are used together with Xfrog-MEL, the structure provided by Xfrog-MEL should be used. Although Xfrog-MEL commands should function with other structures, unexpected results may occur.

Adding CurveParam:

```
$branch1 = `XFCreateBranch`;
$curveEmulation1 = `XFCreateCurveEmulati-
on`;
string $branchKids[] = `listRelatives
$branch1`;
XFConnectCurveAttr $curveEmulation1 $bran-
chKids[0] thickness;
```

Connecting Xfrog object to another as child:

```
$branch1 = `XFCreateBranch`;
$branch2 = `XFCreateBranch`;
XFConnectChild $branch2 $branch1 0;
```

or shorter:

```
XFConnectChild `XFCreateBranch` `XFCreate-
Branch` 0;
```

Connecting primitive object to Xfrog object as child:

```
$branch1 = `XFCreateBranch`;
$primitive1 = ` torus -p 0 0 0 -ax 0 1 0 -ssw 0
-esw 360 -msw 360 -r 1 -hr 0.5 -d 3 -ut 0 -tol
```

```
0.01 -s 8 -nsp 4 -ch 1`;
XFConnectChild $primitive1[0] $branch1 1;
```

Connecting CurveNurbs to Xfrog object:

```
$branch1 = `XFCreateBranch`;
delete `listConnections ($branch1+".inPathCur-
ves[0]")`;
$curve1 = `XFCreateCurveNurbs`;
XFConnectCurve $curve1 $branch1 0;
```

Connecting Tropism to curve of previous example:

```
$branch1 = `XFCreateBranch`;
delete `listConnections ($branch1+".inPathCur-
ves[0]")`;
$curve1 = `XFCreateCurveNurbs`;
$tropism1 = `XFCreateTropism`;
XFConnectCurve $curve1 $tropism1 0;
XFConnectCurve $curve1 $branch1 0;
```

Complex scripting example:

The following example creates a Branch with CurveNurbs as path. A Tropism object is assigned to the path. Finally a branching structure is created.

```
$branch1 = `XFCreateBranch`;
delete `listConnections ($branch1+".inPathCur-
ves[0]")`;
$curve1 = `XFCreateCurveNurbs`;
$tropism1 = `XFCreateTropism`;
setAttr ($tropism1+".axis") 3;
XFConnectCurve $curve1 $tropism1 0;
XFConnectCurve $curve1 $branch1 0;
$primitive1 = `sphere -p 0 0 0 -ax 0 1 0 -ssw
0 -esw 360 -r 1 -d 3 -ut 0 -tol 0.01 -s 8 -nsp 4
-ch 1`;
XFConnectChild $primitive1[0] $branch1 2;
$branch2 = `XFCreateBranch`;

XFConnectChild $branch1 $branch2 0;
```

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 "expression" field inside the "Overall" settings of a control curve. The examples below show the Thickness parameter of the Branch Object.
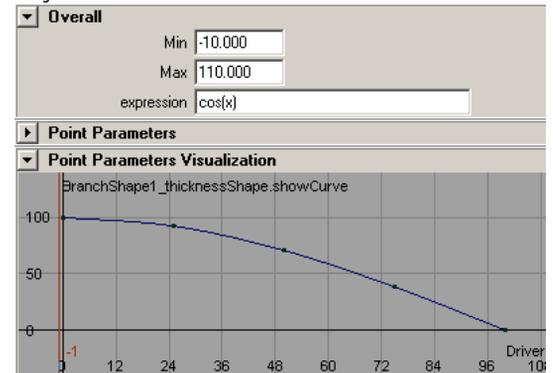
Select the Branch object and go for the "Standard Curve Parameters". Access the control curve for Thickness (create a curve if neccessary, by clicking the red plus icon) and take a look at "Overall".
There you can find the "expression" field wich contains "id" as default function. "id" simply returns the values of the control curve.

The example above shows the funtion "cos". If you want this function to work on the control curve values, then you have to enter "cos(x)" into the expression field.

If you do this for the Thickness parameter of a Branch object, then the result should look similar to the picture on the left.

Now let us work a little bit on this example. Switch to the "Overall" settings of the Branch object and increase the "Smooth Path" value to about 20.

Then access the parameters of the Thickness control curve (BranchShape1_thicknessShape) and take a look at the Point Parameters Visualization. Select the control points in the middle of the curve and delete them, so that we only have a point at the left and one at the right. Set "Point 0 y" to a value of null, "Point 1 y" to a value of 628.

See the picture below:

If the expression field of the Thickness parameter still shows "cos(x)", then the resulting shape of the Branch object should look like the picture on the left. Now you also will understand, why we increased the value of "Path Smooth" - we need a mesh with higher resolution to get a nice wave because of the cos(x) function inside the Thickness parameter.

Now we will increase the frequency of the function. Head over to the expression field and enter "cos(10*x)" as new function. The resulting shape should look similar to a high voltage insulator.

Because of the nature of the cosine function, the "Thickness" parameter takes values between -1 and 1. To avoid a negative thickness we will add the abs() function to the expression. abs() limits the output to positive values. So the expression would look like this: abs(cos(10*x)). But we even can improve this result. The thickness never should reach a value of zero. So perhaps you want to try the following expression: 0.2+abs(cos(10*x)) Because of the term "0.2", thickness will have a minimum value of 0.2. On top of this, the wave of the cosine function is added. The final shape of the Branch object should lool like in the image on the right.

So to summarize: the values of the control curve are passed to the expression. The func-

tion then calcualtes the final result and generates the Thickness of the Branch object.
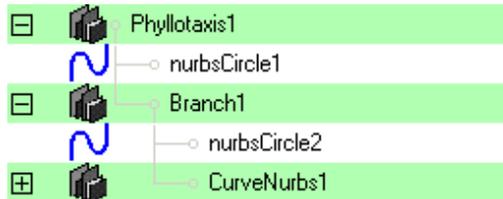
Besides the "x" variable it is possible to refer to values like the iteration number of Objects that are multiplied by a multiplier Object ("i") or the amount of child objects of the parent Object ("n"). Below you find an example for the use of "i" and "n" together to achieve certain modeling effects. In this example the expression i/n*x is applied to the Rotation X parameter to produce the following shape with a gradually changing curvature.

The expression is applied to the Rotation X parameter of the CurveNurbs object inside the Branch object. Because of "i" which stands for iteration, the first Branch recieves zero as value for "i", the second Branch recieves one, the third one recieves two and so on. So the higher the iteration number of the instance, the

stronger the branch will be bent by the expression.

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.

Due to a limitation, "f" and "t" inside the expression editor are only evaluated, if the control curve is (at least slightly) animated.

The following pages show all functions that are provided in Xfrog 5 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 "mathematical perfection". Certain functions are only defined within a certain range of input 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 calculations. 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 refer to the time and allow you to integrate functions into your animations to achieve specific effects.

x – refers to the current value of the curve

t – refers to the animation time in seconds

f – refers to the frame number

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

n – refers to the amount of child Objects of the parent Object

# 4.1 Functions Overview:

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

Function: y = sin x
Description: sine
Input values: -∞ to ∞
Output values: - 1 to 1
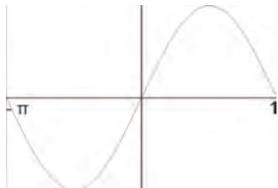
Function: y = cos x
Description: cosine
Input values: - ∞ to ∞
Output values: - 1 to 1
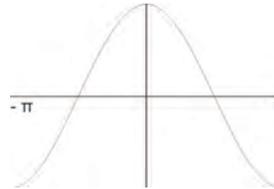
Function: y = tan x
Description: tangent
Input values: - ∞ to ∞
Output values: - ∞ to ∞
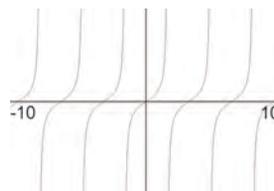
Function: y = asin x
Description: arc sine
Input values: - 1 to 1
Output values: - π to π
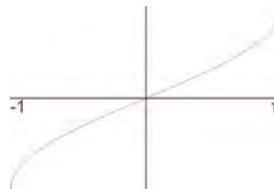
Function: y = acos x
Description: arc cosine
Input values: - 1 to 1
Output values: - π to π
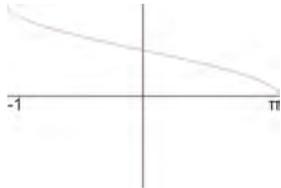
Function: y = cosh x
Description: cosine hyperbolic
Input values: - ∞ to ∞
Output values: 0 to ∞
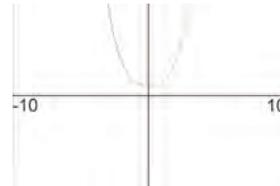
Function: y = atan x
Description: arc tan
Input values: - ∞ to ∞
Output values: - π/2 to π/2
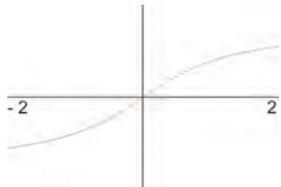
Function: y = tanh x
Description: tangent hyperbolic
Input values: - ∞ to ∞
Output values: - 1 to 1
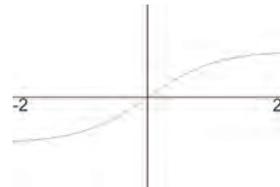
Function: y = sinh x
Description: sine hyperbolic
Input values: - 1 to 1
Output values: - π to π
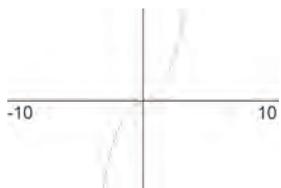
Function: y = exp x
Description: exponential
function to basis e=2.7182818
Input values: - ∞ to ∞
Output values: 0 to ∞
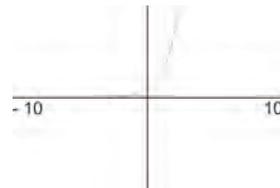
Function: y = log x
Description: logarithm to
basis e=2.7182818
Input values: 0 to ∞
Output values: - ∞ to ∞
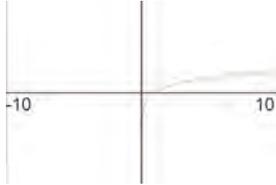
Function: y = sqr x
Description: square x (x2)
Input values: - 1 to 1
Output values: 0 to п
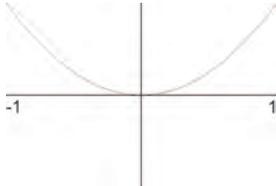
Function: y = sqrt x
Description: square root of x
Input values: 0 to ∞
Output values: 0 to ∞
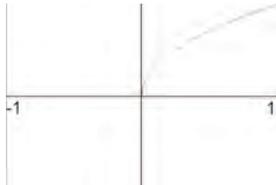
Function: y = ceil x
Description: the smallest integer
greater or equal than x
Input values: - ∞ to ∞
Output values: - ∞ to ∞
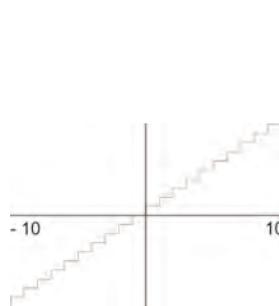
Function: y = floor x
Description: the greatest integer
smaller or equal than x
Input values: - ∞ to ∞
Output values: - ∞ to ∞
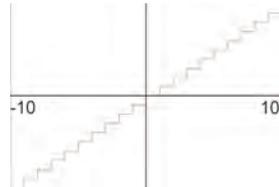
Function: y = rnd x
Description: random number between [-x..x]
Input values: - ∞ to ∞
Output values: 0 to ∞
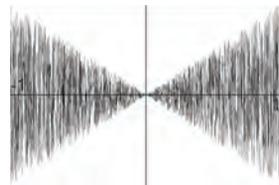
Function: y = rndabs x
Description: x*abs(rnd(1))
Input values: - ∞ to ∞
Output values: 0 to ∞
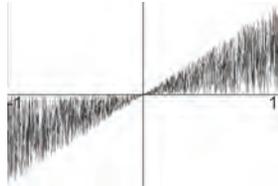
Function: y = phi x
Description: x*1.618034
Input values: - ∞ to ∞
Output values: - ∞ to ∞
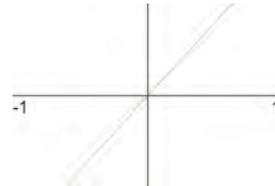
Function: y = abs x
Description: abs(x)
Input values: - ∞ to ∞
Output values: 0 to ∞
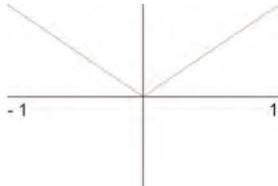
Function: y = rad x
Description: x*3.1415927/180
Input values: - ∞ to ∞
Output values: - ∞ to ∞

5. Reconstruct Nature

To be honest - this task is almost impossible ;-) Nature is unbelievably rich in detail. Everytime you get closer to a real plant you will spot new "features".

Today computers are simply not capable of handling all the necessary data. But on the other hand, the human eye and brain can be tricked. So if you invest some time and take a look at nature, then this will give you great advantages when modeling plants with Xfrog.

One of the most important tasks is to anatomise the real world object and then choose the proper Xfrog components to reassemble the plant.

Deciduous trees and bushes

This is one of the most common tasks you will stumble over. This kind of plant comes with a stem, several branching levels and finally leaves.
To rebuild this structure you could use a:

   Branch object as stem
   Branch objects for branching levels
   Branch object with flat profile, or
   square object to create a leaf

Conifers

The structure of conifers is similar to the one discussed above, but different. In many cases you will have several branches emerging from one "knot". So you could set the "Node Multiplicity" of the Branch object to higher values.

Daisy-esque flowers

Here we could separate into following parts: stalk, leaves, blossom. The blossom could be separated into calyx(leaves), petals, center.

The stalk easily can be simulated with a Branch object. For the leaves we also could use a Branch object, just with a flat profile curve. Calyx: here we have several options, a Branch object could be used, but also a Maya Revolve object.
To create a petal you could make use of the -tada! - Branch object, again with a flat profile. To arrange the petals in a circle, make use of the Hydra or Phyllotaxis object.

The center: the easiest way is to go with a Square or Plane object and a good texture (with alpha map). A second way would be to rebuild one of the "miniparts" of the center and arrange hundreds or thousands of them with the Phyllotaxis.

6. FAQ

This chapter tries to answer some of the most frequently asked questions. If something strange happens, check the following answers.

Q: I put a Branch object on top of another, but nothing seems to happen.
A: Branch objects depend on the Growth parameter. The default Node Growth curve parameter is set to zero at the top of the Branch - this is the reason, why the Head-Branch doesn´t grow. Simply increase the Node Growth at the top of the base Branch object.

Q: I followed above mentioned steps, but still the top Branch object doesn´t appear.
A: Branch objects also depends on Thickness. By default the Thickness is set to 0% at the top of the Branch object, so the base Branch passes a Thickness of 0% to the top Branch object - and this is so slim that you won´t be able to spot the top branch. Simply stop the diet and increase the Thickness of the base Branch object.

Q: I imported a *.xfr file with options set to Xfrog Object Import. The plant hierarchy is visible inside the Visor, but no object shows up in the 3D view.
A: During import, "Evaluation" of the topmost Xfrog object is turned off.
The reason: you can check the hierarchy and consider if the object is okay, or if the object might be "CPU heavy". Simply turn "Evaluation" of topmost Xfrog object on to generate the geometry.

Q: What does the Receptacle option do (Phyllotaxis)
A: Nothing. It´s just there to make you wonder. To be honest it´s a relict that should vanish with the next update.

7. MEL Reference

Here you can find a list of all MEL commands supplied by Xfrog 5.

aboutXF:

Simple command, that shows Xfrog-About-Box.
Syntax: aboutXF;
Return: none

XFCreateCurveEmulation:

Creates curveEmulation-node (Xfrog CurveParam). These nodes are used as input from Xfrog objects. For UI-use they are generated automatically, but they can be generated separately using this command.
Syntax: XFCreateCurveEmulation;
Return: Name of curveEmulations transform-node

XFCreateCurveNurbs;

Creates CurveNurbs-node (Xfrog Curvature) and needed structure around. In this case simply a spline for visualzation-purposes.
Syntax: XFCreateCurveNurbs;
Return: Name of CurveNurbs' transform-node

XFCreateBranch:

Creates Branch-Node and relative transform-nodes-structure to keep added child/shape/head-primitives organized, a standard path-, a standard shape-spline and a polyShape for mesh-output.
Syntax: XFCreateBranch;
Return: Name of Branchs transform-node

XFCreateTropism:

Creates Tropism-node.
Syntax: XFCreateTropism;
Return: Name of Tropisms transform-node

XFCreateVariation:

Creates Variation-Node and structure around. In this case transform-nodes-structure to keep added child-primitives organized.
Syntax: XFCreateVariation;
Return: Name of Variations transform-node

XFCreatePhyllotaxis:

Creates Phyllotaxis-Node and structure around. In this case transform-nodes-structure to keep added child-primitives organized and a standard path-spline (half

circle).

Syntax: XFCreatePhyllotaxis;

Return: Name of Phyllotaxis' transform-node

XFCreateHydra:

Creates Hydra-Node and structure around. In this case transform-nodes-structure to keep added child-primitives organized.
Syntax: XFCreateHydra;
Return: Name of Hydras transform-node

XFConnectCurveAttr:

Connects a curveEmulation (CurveParam) –node to a defined CurveParamIn-plug of an Xfrog object.
Syntax: XFConnectCurveAttr curveEmulationName Xfrog objectName Xfrog objectParamInPlugName;
Return: Name of curveEmulations transform-node after connection

It is allowed to use the names of either the transform-node or the shape-node of the curveEmulation. The name of the shape-node must be given for the Xfrog object. The name must be unique. If not, the full dagpath must be given as name. It is also allowed to, having selected nothing, select first the curveEmulation-node, then add-select the Xfrog object-node and then execute the command using the syntax:

Syntax: XFConnectCurveAttr Xfrog objectParamInPlug-Name;

Return: Name of curveEmulations transform-node after connection

The standard parameters valid for the given plug are set to the curveEmulation-node, which can be further used to edit the CurveParam.

XFConnectChild:

Connects an Xfrog object or primitive geometry-object as child, shape or head to an Xfrog object.
Syntax: XFConnectChild childName parentName childType;
Return: Name of child's transform-node after connection

The parentName can be the transform- or shape-node of a Branch, Phyllotaxis, Hydra or Variation. The childName can be the transform-node or the shape-node of Branch, Phyllotaxis, Hydra, Variation or any object or group of objects only consisting of transform-, mesh-, nurbs- and subdiv-nodes. If the parentObjects multiplication-type is set to instance or copy, additional-

ly any object or group of objects not containing any Xfrog objects can be used as primitive geometry. (This way, it is possible to add PaintFX-branches and –leafs to trees).

Explanation:
If the given child is an Xfrog object, it is placed under the parent Xfrog object and its inParameters-plug is connected to the parent-out…Parameters-plug defined by childType. For primitives applies the following: If the parentObjects multiplication-type is set to instance or copy the primitives are childed somehow and copied or instanced by the needed child-count and the top-transform-nodes of the created instances/copies are connected to the parents out…T/R/S-plugs defined by childType. If the multiplication-type is set to mesh otherwise, all mesh-, subdiv- and nurbs-nodes in the given primitive object or group of primitives are childed to parent somehow and connected to the associated inMesh-, inNurbs- and inSubdiv-plugs and for each an also childed polyShape-node is created and connected to the concurrent out…-plugs.

The childType defines the way the objects are connected. For Hydra and Phyllotaxis only 0 for child is allowed. For Branch any number between 0 and 2 is allowed, 0 for child, 1 for shape and 2 for head. For Variation any number between 0 and number of Variations childs –1 allowed, defining the childNumber to be used for the child. If the children were of primitive type, they are made invisible to be just used as geometry-source. It is also allowed to, having nothing selected, select first the childNode, the add-select the parentNode and then execute the command using the syntax:

Syntax: XFConnectChild childType;
Return: Name of child's transform-node after connection

XFConnectCurve:

Connects Tropisms to splines with extended functionality (CurveNurbs, CurveNurbs or native Spline influenced by Tropism) or standard nurbs-splines or these to Xfrog objects.
To understand the behaviour of this command it is needed to know how the curve-information-flow works in the dependency graph.
If Tropisms are used, the left one is always required and an arbitrary number of the right one can be used.

curve-spline connected via

worldSpace[0]-plug

Xfrog object in-connected via inPathCurvesData-plug (if first element is a curve-spline and no Tropism is used via inPathCurves[0]-plug

Tropism in-connected via inPathCurvesData-plug and out-connected via outPathCurvesData-plug

Tropism in-connected via inPathCurves[0]- or inPath-CurvesData-plug and out-connected via outPathCur-vesData-plug, or

CurveNurbs connected via outPathCurvesData-plug

To connect curves (with or without Tropisms attached) to an Xfrog object the following syntax can be used:
Syntax: XFConnectCurve curveName objectName con-nectionType;
Return: Name of the curve's transform-node after con-nection

The curveName can be the name of the transform- or the shape-node of a nurbs-spline or a CurveNurbs. The objectName can be the transform-or the shape-node of a Branch or a Phyllotaxis. The connectionType can be of value 0 for Branches and Phyllotaxis' to connect the given spline as path-spline and of value 1 for Bran-ches to connect the given spline as shape-spline. It is also allowed to, having nothing selected, first select the curve, the add-select the Xfrog object and to use the following syntax then:

Syntax: XFConnectCurve connectionType;
Return: Name of the curve's transform-node after con-nection

To connect a free Tropism to curves (with or without Tropisms and Object attached) the following syntax can be used.
Syntax: XFConnectCurve curveName_or_boundTropis-mName freeTropismName 0;
Return: Name of the object in information-flow, the freeTropism was inserted after

The Tropism to be connected needs to be free and can be given by its shape- or transform-node. The object to connect the free tropism to can be either a curve or a Tropism directly or via other Tropisms connected to a curve. In the information-flow seen above, the free Tropism is attached or inserted after the curve or Tropism given by name. This can be given by trans-form- or shape-node. The returned object-name is the name of the object the freeTropism was inserted after.

To receive the eventually new Dag-name of the former free- and now boundTropism follow the connection of the outPathCurvesData-plug of the returned object-name. It is also allowed to, having nothing selected, first select the curve or boundTropism, then add-select the freeTropism and to use the following syntax for connection then:

Syntax: XFConnectCurve 0;
Return: Name of the object in information-flow, the freeTropism was inserted after

XFFreeObject:

This command is used to group the selected object under world and to cleanly remove any connections to Xfrog objects. For example a boundTropism is grou-ped under world, its connections to binding-partners are broken and the object in the dg-flow before and the object in the dg-flow after are connected. To give another example, a primitive connected to an Xfrog object is disconnected and grouped under world. Its connection to the Xfrog object is broken and the poly-Shape connected to concurrent plug for visualization is deleted.
Syntax: XFFreeObject objectName;
Return: Name of the object grouped under world

If the given object is an Xfrog object (which may have other Xfrog objects or primitives childed to it), it can be given by transform- or shape-node, otherwise the top-level transform-node of an object or group contai-ning no Xfrog objects is expected. It is also allowed to, having nothing selected, select the object to free and use the following syntax for execution then:
Syntax: XFFreeObject;
Return: Name of the object grouped under world

XFUpdateObject:

This command is used to clean up Xfrog objects having any of the update…-plug values set to true. All Xfrog objects that are children of the given Xfrog object are cleaned up too.
Syntax: XFUpdateObject objectName;
Return: none

The Xfrog object can be given by shape- or transform-node. This command is needed, after a change to a Variations number of children using the Variations re-questedChildCount-plug, to keep existing children con-nected to the correct plugs. Also after freeing an Xfrog object formerly childed to another one, the command may be needed to update the changed number of primitive children using copies or instances. All Xfrog-

MEL-commands automatically use this command, so it should be not needed directly after child-connection or freeing. Nevertheless it can be useful, if a polySurface-node to visualize multiplied primitive input is missing or the texture of a primitive input was changed and is needed to be updated at the visualization-surface too. This is done too by this command. It is also allowed to, having nothing selected use this command after selection of the object to be updated with the following syntax:

Syntax: XFUpdateObject;
Return: none

XFSelect:

To switch a relatively slowly updating Xfrog object, with in most cases a lot of connected Xfrog- and primitive children to a faster updating simple mesh, this command can be used. It recursively selects all shape-nodes of Xfrog objects starting by a given top-level-node. If these are deleted then, the remaining objects are simply the primitive mesh. It is also possible to select all meshes generated by Xfrog objects below given node. This may be needed for texturing or similar purposes. It means then, that all meshes forming Branches geometry and all copies, instances and multiplication visualizing polySurfaces of used primitives are selected. Further it is possible to decide, if the selection to be made shall be executed rejecting current selection or in addition to the current selection.

Syntax: XFSelect -mesh/-object -add/-replace object-Name;
Return: none

The top-level Xfrog object can be given by shape- or transform-node. It is also allowed to, having nothing selected use this command after selection of the top-level-object with the following syntax:

Syntax: XFSelectObject -mesh/-object -replace;
Return: none

XFInputVisibility:

All primitives used as input by Xfrog objects are hidden by default delivering just the geometry for the multiplied surfaces, copies or instances. For editing purposes it may be needed to show or hide them during the process of editing. To show or hide the input-primitives for a given Xfrog object use the following syntax:

Syntax: XFInputVisibility -show/-hide objectName;
Return: none

The Xfrog object can be given by shape- or transform-node. It is also allowed to, having nothing selected use this command after selection of the Xfrog object with the following syntax:

Syntax: XFInputVisibility -show/-hide;
Return: none
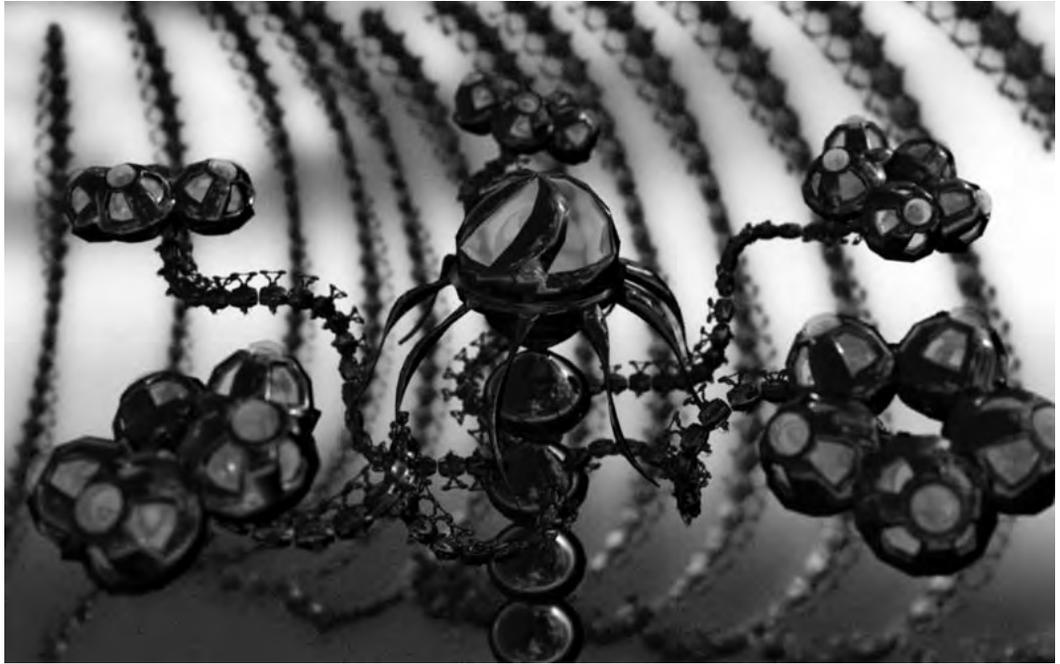
Metalbeast, done with beta of Xfrog 5 for Maya.

Green, plants modeled with Xfrog 3.5 and Xfrog 5   © Jan Walter Schliep