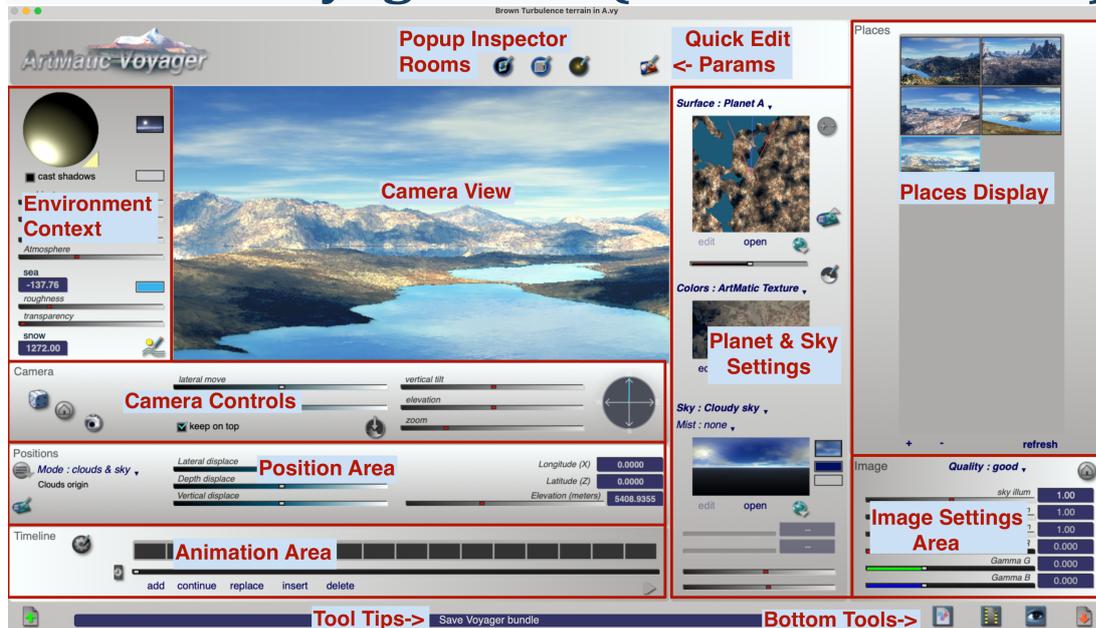


ArtMatic Voyager CTX (Creative ToolboX)

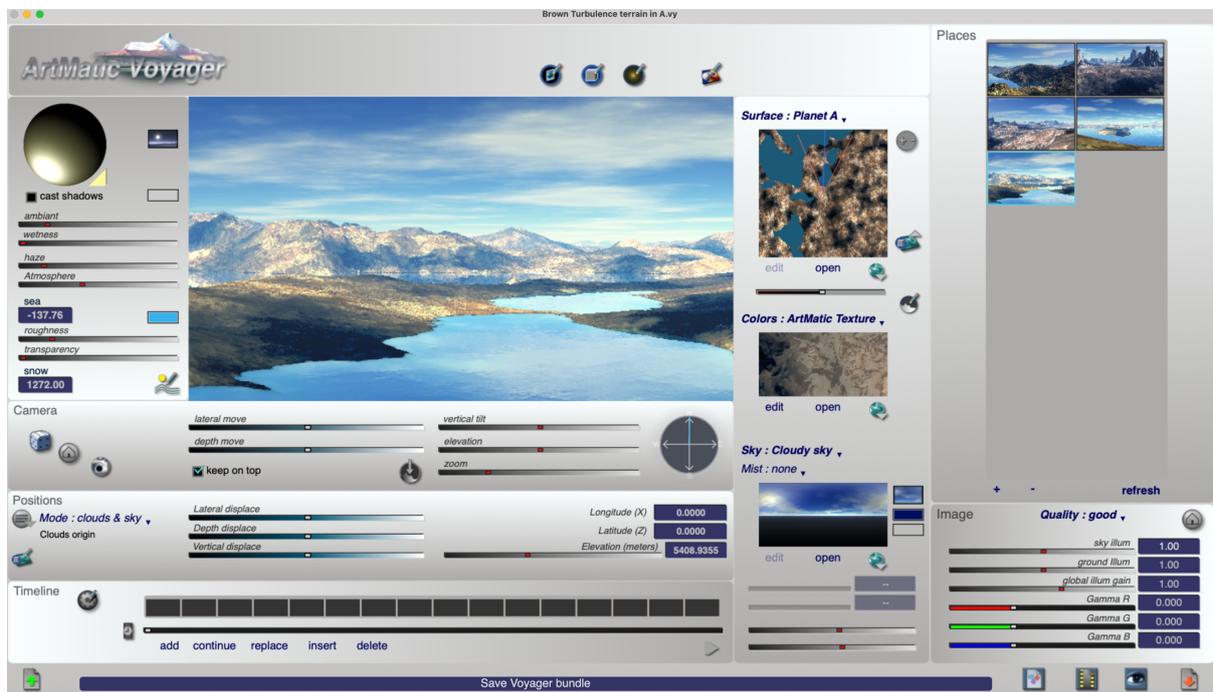


Extended User Guide

2	Overview	39	Planet Building
4	Top Menu & Preferences	39	Designing Terrain Surfaces
6	Interface Basics & Conventions	42	Designing ArtMatic Color Textures
8	User Interface	46	Combination mode
8	Camera View	46	Combination Mode popup dialog
8	Camera Controls	48	Combination mode popup algorithms
11	Environment Context	51	Underwater Shader
14	Environment Settings inspector	55	Building 3D Objects: DFRM Guide
16	Places Display	55	Introduction & Concepts
16	Planet and Sky Settings Area	57	ArtMatic 3D DF objects
16	Planet Surface section	57	Scaling and Size
19	Planet Textures section	58	Positions
22	Sky Settings section	58	Object color
28	Image Settings Area	59	Previewing in Artmatic
30	Popup Inspector Rooms Area	60	Design Guidelines
30	Sprites Inspector	61	DF Modelling Techniques
31	DF Objects Inspector	62	Working with 2D profiles
31	Lights Inspector	64	Working with 3D Profiles
32	Quick Edit ArtMatic parameters	71	Shading DF Objects: shading modes
32	Position Area	80	Ambient Occlusion
34	Animation Area	81	Using extra outputs
36	Animation parameters inspector	82	Performance Tips and Troubleshooting
36	Bottom Tools Area	83	Using DF Vegetation Components

Overview

by Eric Wenger (+ updates)



ArtMatic Voyager is a program for synthesizing and exploring stunning, high-resolution virtual landscapes and worlds. The application is a new take on 3D landscape creation that makes use of the ArtMatic graphic synthesis technology to create photo-realistic landscapes of imaginary worlds. The application can be used standalone using the built-in planets and provided ArtMatic systems, or you can use ArtMatic Designer to define completely new worlds of your own, model 3D objects, create custom clouds and so on.

Using ArtMatic Voyager is simple:

- Choose a planet surface, texture, and a sky definition.
- Define the environment: sun color and direction, sea and snow levels, haze/humidity.
- Travel the planet by moving and aiming the camera.
- Save interesting locations as places or keyframes.
- Render still images or movies.

Camera movement and environmental parameters can be animated to create stunning planetary voyages. If ArtMatic trees are used to define the planet's topology or shading/texture, the surface and shading can also be animated so that planets can be explored while they evolve.

ArtMatic Voyager uses a unique approach to creating virtual landscapes. Rather than polygon-based techniques, Voyager uses procedural functions to generate planet-sized terrains that are rendered as three-dimensional landscapes. This technique allows Voyager to easily create huge planets without any database. Furthermore, most of the procedural building blocks are band-limited adaptive fractal functions which allows you to achieve stunning foreground details without wasting processing power for details far in the background.

ArtMatic Voyager can render views as still images or animation. Environmental parameters and the camera position can be keyframed and used to render stunning animation. The surface texture and shading can even be animated when using ArtMatic Designer terrain maps and color textures. Using the built-in planets or the provided ArtMatic planets is simple and fun. Advanced users can define

their own planets and textures using ArtMatic Designer “ either by starting from the provided examples or by creating new systems from scratch.

The ArtMatic Voyager Advantage: Voyager directly renders planets from a mathematical description of the planet surface. Elevations, defined by procedural mathematical functions or ArtMatic systems (which are themselves compact procedural mathematical functions), are evaluated on the fly as the rendering proceeds. By contrast, traditional 3D landscape software stores elevation data in an array or database called an elevation map or height map. The enormous advantage of Voyager's approach is that it allows huge worlds (many times the size of our Earth) with infinite levels of detail to be defined compactly since there is no need for a database. As a result, the memory requirements and storage size of a planet is trivially small, and yet a Voyager file can yield scenes of extraordinary detail.

Beyond planets a Voyager scene can also feature cities, architecture, spaceships, vegetation and even animals. ArtMatic Engine 7 introduced procedural volumetric infinite cities. ArtMatic Engine 8 completed the set of DF primitives used in the 3D objects design. As with terrains, Voyager has a procedural approach to modelling through the technic called Distance Field Ray Marching (DFRM) which offers an incredible flexibility. DFRM technics are covered in detail at [Building Objects : DFRM guide](#).

Planets & Built-in Planets: The size of a Voyager world is theoretically infinite, but practical considerations (primarily related to limitations of values that can be represented by a computer) require that planet size be limited. ArtMatic Voyager planets are planar (flat, non-spherical) terrains 60,000 kilometers by 60,000 kilometers - more than three times the size of the Earth. (The Earth has a surface area roughly 40,000 by 20,000 kilometers). **Surface amplitude** can be controlled, and allows the same elevation description to create subtle plains or extreme mountains and canyons. 3D ray tracing techniques are used to create the lighting effects.

Distance is measured in kilometers. Latitude and longitude are expressed as offsets from the planet center. So, the map extends +/-30,000 kilometers North/South (latitude) and +/-30,000 kilometers East/West (longitude) from the planet center. Voyager provides 5 built-in planets that are big and varied enough to spend years in exploration. They combine a variety of terrain procedures using complex random filters that provide a great deal of topologies. These planets could mostly be created in ArtMatic surface mode but would require very complex ArtMatic structures that would render more slowly than the built-in functions. **Oceans** and **snow level** are defined with environment variables (the sea level and snow level controls) and are not part of the planet topology. Learn more about planets and ArtMatic defined terrains in [Surfaces modes](#).

Context variables: ArtMatic Voyager provides you with a great deal of control over the planet's environment. The light, haze and other environmental parameters can dramatically change the planet's appearance. Most of the environmental controls are found in the lefthand toolset and in the Environment settings dialog. The environmental variables, with the camera and the built-in lights, constitute what we will call the Voyager Context variables. They include:

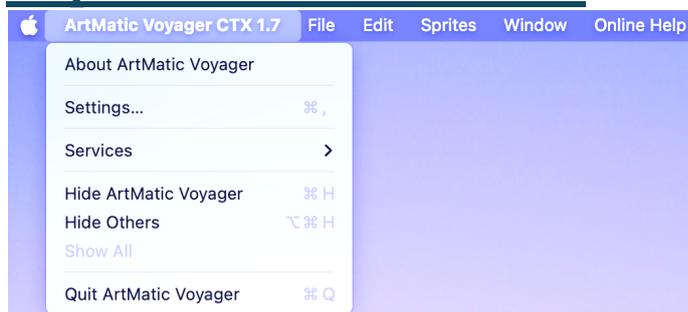
- Camera location and view direction
- The sun's position & sun color
- The Ambient level controls the amount of incidental or ambient light that is available
- The Wetness level controls the degree to which the water and low elevations reflect the sky color
- Haze and the Haze Color picker give you control over environmental haze
- Sea level & sea color sets the altitude and color of the global planet ocean
- Sea Roughness slider influences the wind speed (and thus the motion of clouds when a system is animated)
- The Snow Level sets the altitude at which snow will be automatically added
- Built in lights that provides additional suns or light sources

- Clouds density and position
- Sky Environment rotation or position for backdrops
- Illumination gains and Gamma filter

Context variables can all be animated as each Voyager Keyframe will store a copy of the entire Context. You can disable the animation of a particular variable with the Animation Parameters inspector. *Note that Planet mode and Sky mode and some global settings (Mist, Atmospheric color shift, Underwater color shift and various rendering modes) are NOT part of the context and can't be changed while animating.*

Places: ArtMatic Voyager lets you store favorite places on the planet in the Places display. When saving a place, ArtMatic Voyager stores the current settings defined by the Context variables. Thus Keyframes & Places are very similar in that they both store the complete set of Context variables. Places make sense in relation to a specific planet so obviously when you change the main planet the locations may become problematic (eg. if the camera is below ground). However, as the place saves other variables, it is interesting to keep them as ways to store a particular ambiance and illumination, so they are not cleared when changing the main planet. Just move the camera and resave the problematic places in that case.

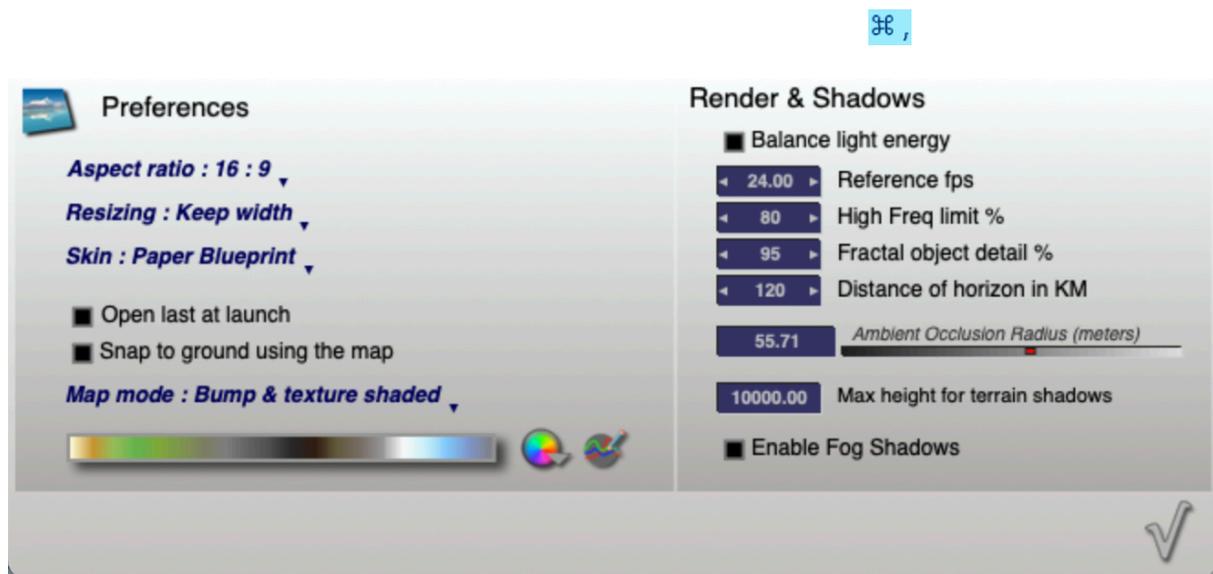
Top Menu & Preferences



When ArtMatic Voyager is the active application, the top menu appears at the top of the Mac screen. This allows selection of a number of actions that can also be done directly with the Voyager UI, and allows access to on-line documentation about ArtMatic voyager.

Click the ‘**Settings...**’ button to open Voyager’s ‘Preferences dialog.

ArtMatic Voyager Preferences dialog: (shortcut: cmd-comma)



Aspect Ratio: Sets the overall aspect ratio of the canvas. Options are...

- *Square*,
- *4 : 3*, (Horizontal 4 by 3 ratio.)
- *16 : 9*,
- *16 : 8* (The HD aspect ratio, most commonly used for animation). This 2:1 ratio is also suited for 360 degree environment map creation.

Resizing pop up

- *Keep width*. When rendering at a different ratio than the overall aspect ratio this choice will keep the x range.
- *Keep height*. When rendering at a different ratio than the overall aspect ratio this choice will keep the y range

Window Skin: Sets the skin of the ui among various styles:

- *Blue Steel*,
- *Metal Noise 45*,
- *Dark Noise*,
- *Shaded Metal*,
- *Shaded Grays*
- *Paper Blueprint*

Open last at launch (checkbox):

When ON this checkbox option will make Designer start with the last opened Voyager file to save time when working on a project. Turn it off if you want to start with a new scene. Note that Voyager will use the last context variables and terrains anyway.

Snap to ground using the map (checkbox):

When ON click on the main ui Map area will land the camera close to ground level automatically.

Map mode: Map mode provides 2 ways of drawing the top view map :

- *Bump & texture shaded*: use the terrain texture with bump shading (default)
- *Elevation color map*: maps the elevation to the color pallet provided below.

•  **choose gradient**

•  **edit gradient**
calls the gradient editor to modify the elevation color palette

Balance light energy (checkbox):

When ON, Voyager will use a new illumination model (developed for CTX 1.5) that normalizes the coefficients of all type of light transport mechanism, so that the sum of all stays below 1. Practically it means that all levels are balanced automatically, so that light output can't be bigger than the inputs (which was a problem before with reflections for example. So high reflections will lower diffuse, specular and ambient internally).

With "Balance light energy" ON, it is easier to obtain a more pleasing and realistic illumination. Examples are provided in *../Voyager Examples/VY15 Examples/Illumination/*

Reference fps: Provides the default fps (frames per seconds) for animation.

High Freq limit: This setting adjusts the high frequency filtering limit. Keep it at 100% to have the maximum details. If too much flickering occurs when animating, you may try to lower it to 70 % or less.

Fractal object detail: This setting can lower the high frequency limit of fractal DF objects. Keep it at 100% to have the maximum details. If the fractal looks too noisy you may try to lower this value.

Distance of horizon (KM): The maximum distance of visible objects and terrain can be adjusted here. Usually at 120 km. Set it to a large value if high mountains suddenly appear in animation when the camera moves.

Ambient Occlusion Radius (meters) slider: Sets the average Ambient occlusion radius for objects.

Max height for terrain shadows:

Enable Fog Shadows (checkbox):

Interface Basics & Conventions

Almost everything that you can see in the user interface is active, including the text, icons and glyphs. Pretty much every graphic item can be clicked or dragged to perform a task. Like all U&I Software applications, most tools are accessible directly from the user interface.

Tool Tips : The Tool Tips area found at the bottom-center of the main window provides useful information about whatever is under the mouse. Move the mouse over any user interface item to display helpful information. Often the tip will include shortcuts keys if any.

Numerical controls and slider: Numeric controls allow you to change values by typing or by clicking and dragging. When typing, complete entry by pressing return or entry key. Selecting another field should also validate the entry. You can change the number with smaller increments when pressing the *option* key while dragging a slider horizontally or a number field vertically.

Shortcuts:

* (times 2) changes the field value to twice its value

/ (divide by 2) changes the field value to half its value

i (invert) changes the field value to 1/value ,

d (degrees) interpret the entry as degrees converted to radians and shall be used at the end of keyboard entry. for example to get exactly Pi type 180 then 'd'. d should validate the entry automatically.

Color Pickers: Color swatches allow you to change items such as the sun and sky color. Click and hold a swatch to pop up the color picker. The cursor becomes an eyedropper that picks up the color beneath it when the mouse is released which makes it easy to grab color from anywhere in the background. Unfortunately Apple in recent OS makes the reading of screen pixel subject to Authorization so you will have to grant Voyager the right to access the screen otherwise the color picker won't work. Keep in mind the picker can read any color anywhere on the screen which is extremely useful as you can pick a color from a picture unrelated to Voyager on the desktop for example.

Inspectors and Dialog: Introduced in Voyager 5, Inspectors allows users to edit deeper settings in a non-modal way, unlike dialog. Thus we keep the name "dialog" for modal windows that prevent main UI to operate, while we use "inspector" for UI that pops up in certain areas but don't prevent controls, menus and main preview to be functional.

The most important UI tools to control objects, sprites, lights, animation, render options and environment controls have been reimplemented as "inspectors".

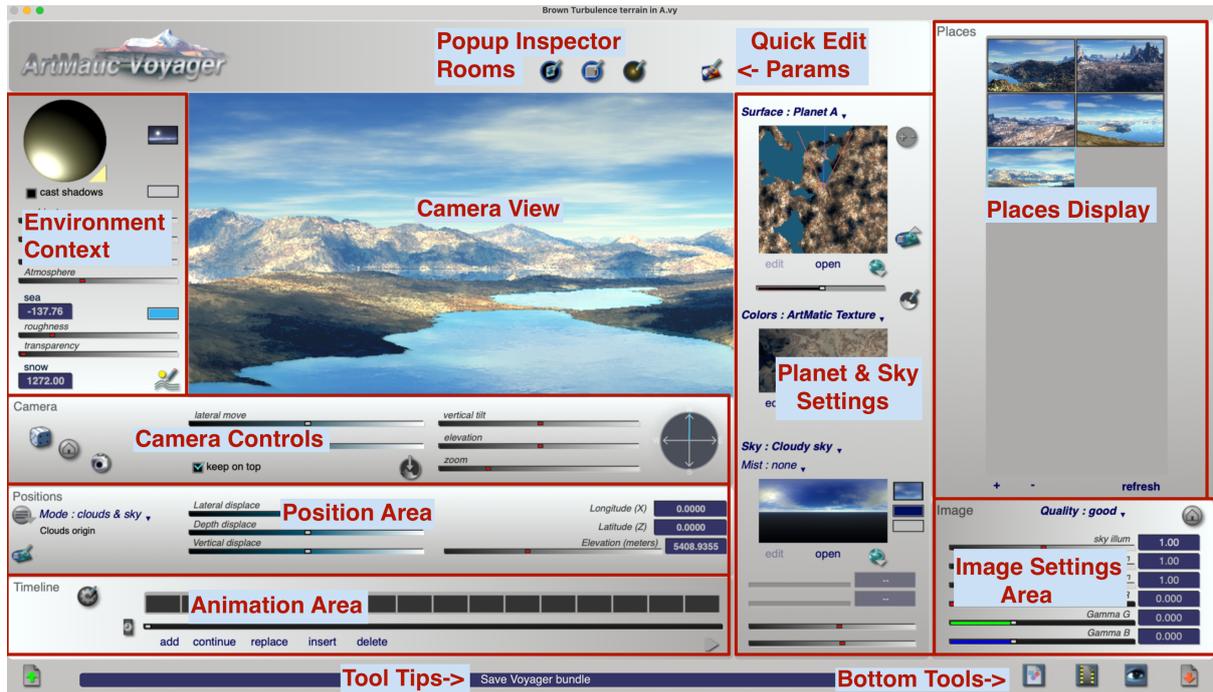
Inspectors will appear on the side without hiding the main view. The main UI that controls the Camera, Environment and Positions is still accessible while using a specific inspector. Modeless "inspectors" are used for Objects, Sprites, Lights, Animation and deeper Environmental options and parameters. They can be opened using the icons on the top right of the UI. The one specific to timeline 'Animation parameters' is located in the timeline area.

Some functionalities that were in the dialogs were moved to the main UI because the non-modality make them usable even when a particular "inspector" is open. For example a new section 'Positions' below the 'Camera' handle the positioning of every element in a VY scene, which meant the position slider became redundant in the dialogs. '**Positions**' can apply to objects, lights, clouds, textures and even the main planet terrain. In general the target will be set automatically when selecting an object or a sprite or a terrain, but you can use the mode pop-up at the left of the section to decide to move the cloud layer, even if you are placing sprites or objects at the same time.

So practically, if you open the object inspector to import various objects you can still move the sun or change the camera view. Keyboard entries can't be assigned to several windows simultaneously. Thus a selection click is needed on Main UI before keys input can be redirected to it when an Inspector is Open. Likewise click on the Inspector area to redirect keyboard entries to the inspector.

Adaptative resolution: Most of all sliders give you real time preview at an adaptive resolution. You can click and drag on any control while seeing in real time the result of camera or object changes. When the scene is slow, this real time preview will be at a very low resolution, but enough to have a useful feedback. As soon as you release the mouse a better rendering is started; a rendering that can be interrupted at any moment to tweak another parameter.

User Interface



Camera View:

This is the main display area. It shows the scene as seen through ArtMatic Voyager's virtual camera. Users can click and drag left/right or up/down to move the camera instead of using the camera controls.

Camera Controls:



This area gathers controls and buttons dedicated to the Voyager Camera. Any scene is viewed through a virtual camera whose position is user-controllable and animatable. The Camera can use Cylindric, Perspective or Spherical projection. The camera's latitude and longitude, elevation above the terrain, the vertical tilt, and the rotation are usually set with the sliders below but you may use the Map to directly move the camera over the top Map view as well as a click and drag on the main Image preview to move the camera view directly. For small adjustments you may also use the arrows keys

Random place (r)



The Random place button picks a location at random within the 60,000 square km of the current planet to put the camera. Direction of the camera is randomized as well. It is a fun way to explore the gigantic worlds provided by Voyager.

Shortcut: 'r' key.

Reset view



The Camera home button will in most cases place the camera at a default world center position. It can be used to find an object or sprite as well: (If a Sprite or Object is set as target (see browse scene elements picker in 'positions') the camera reset view (home) button will move the camera to focus on the selected object).

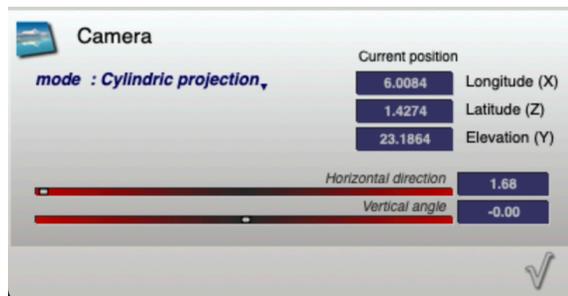
Otherwise the reset view button will return the camera to the origin: Latitude -1 km and Longitude 0 with a north-facing orientation and the default altitude and zoom angle (about 53 degrees). Home is often used after choosing an ArtMatic file as a surface since the most interesting features often occur there near origin.

Camera Settings...



This button invokes the Camera Settings dialog.

The Camera Settings lets you choose the projection mode and set the Camera position in absolute coordinates. It also provides a slider for camera orientation in degree and camera tilt (vertical angle).



- **Cylindric projection :**

This projection mode is the default and allows some very efficient optimisation technics for terrain rendering. It is usually the fastest camera mode when using Height-field based terrains. However, Cylindric projection prevents the camera from pointing straight up or straight down and will make verticals all parallel. Horizontal lines on the ground will be curved.

- **Perspective projection :**

This projection is the usual projection found in 3D applications and divides coordinates by the depth. Horizontal lines on the ground are kept linear, and parallel will converge at horizon. Image distortion will be greater at wide focal angles and 'Spherical projection' could be used instead.

- **Spherical projection :**

Spherical projection uses a spherical lens that can go up to very wide angle focal renderings (up to 360 degree), and is similar to a fish eye lens. At wide angles straight lines will become curves and the horizon will be a curve if camera is pointing up or down. At low focal angles the curving of lines is minimal and the projection gets closer to a standard perspective projection.

This mode allows you to render a full 360-degree panorama which can be used for VR and viewing in apps that allow viewing of full 360 panoramas (such as Facebook 360 mode).

These 360 renders can also be used for environment map texturing and ambient map image within voyager.

When rendering with this mode, make sure to use a 2:1 aspect ratio and to set the Zoom (focal angle) slider all the way to the left (the numerical value is 360).

Keep on top

This checkbox ensures that when you are traveling over the surface of the world that you don't end up inside of a mountain. When this option is on, the camera elevation will rise above the terrain if you move to a place where the terrain is higher than the camera's elevation.

When DF objects, terrains or DF cities are present "Keep on top" also activates collision detection to avoid the camera moving inside features. You may have to turn it off in cases there are structures higher and you still need to place object below, like if you are in a room and don't want objects to be moved to the roof.

Land Camera (z)



The "Land Camera" button puts the camera slightly above surface level, whether the surface is a planet terrain or a 3D volumetric DF object. Shortcut: 'z' key.

Lateral move:

This slider lets you move laterally in respect to the camera orientation. The range of the displacement depends on the current MAP view scaling. For small adjustments use *option* key (option will divide the slider range by a factor of 1/50).

The slider is relative to current position meaning it will be set at zero after each use, zero representing current position. You can also use the left/right arrow keys to move laterally.

Depth move:

This slider lets you move forward and back with respect to the camera orientation. Since the motion is relative to where the camera is pointing keep in mind that if pointing up this slider will make the camera go up as well as forward. If you need East/West or North/South displacement without changing altitude use the absolute coordinates numbers in the Camera settings dialog. Use *option* key for small adjustments. Key equivalent: *up and down arrow* key.

Elevation:

This slider controls the camera's elevation. The slider is relative to current altitude position. The range of the displacement depends on the current MAP view scaling. Altitude is displayed in meters. If you need to set camera elevation to a precise altitude use the Camera settings dialog absolute altitude and set the altitude numerically. Key equivalent: *Page Up/Page Down* or *control up/down arrow*.

View Direction Compass:

Click anywhere on the compass (located on the right of the camera area) to rotate the camera. The camera will rotate and point at the position where you clicked. You can also click and drag to rotate the camera. Shortcut: *Control + left/right arrow*.

Vertical Tilt

This slider controls the upward /downward tilt angle of the Camera.

NOTE: When the camera is in Cylindrical mode the virtual lens is cylindrical and does not curve verticals. This allows for very efficient optimizations but causes the y axis to have no curvature while the horizontal axis can curve up to 360 degrees. It also prohibits looking straight downward as there is no downward perspective possible with this projection. The vertical tilt is actually a vertical offset in the image space and does not modify the camera's position or the perspective.

Zoom (Focal angle slider)

Zoom in or out on the landscape without moving the camera. Notice that the Surface Map reflects the zoom setting by showing the angle of view visible through the camera when the angle is below 180 degrees. Zoom changes the effective focal length of the camera's virtual lens. Low zoom values correspond to wide-angle lenses and high values correspond to telephoto. The minimum zoom level provides a full 360 degree view angle and can be used to render full panoramic pictures. The maximum zoom level provides a viewing angle of about 22 degrees. Spherical projection is advised with wide angle focals.

Environment Context



The tools on the left of the Voyager UI are dedicated to Environment control variables that are part of the current Context variables.

Sun direction sphere

The sun direction sphere allows you to set the position of the sun that illuminates the planet. The sun's angle realistically influences the color, the shadows, and reflections. Click on any point to focus the sun's light at that point. The bright spot indicates the sun's position relative to the planet's sky dome which is represented by the sphere. When the illuminated spot is in the center of the globe, it is noon (i.e. the sun is directly overhead). When the illuminated spot is at the edge of the globe, the sun is dipping towards the horizon.

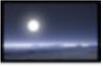
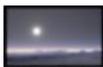
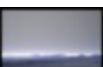
In Voyager it is possible to put the sound below the horizon. In 'no planet' mode, this allows for the sun to light objects from below.

Sun color box

To change the sun color, click and hold the sun color rectangle to pop up a color picker. Sun color affects all colors in the scene and the clouds. The sun provides directional lighting and is the primary source of illumination, whereas the **Ambient slider** provides diffuse non-directional lighting. To decrease the sun's contribution to the lighting, lower the brightness of the sun color. If you set the sun color to black, the ambient light from by the sky and the haze will provide the scene's illumination and shading. Varying sun color along with the haze color and sun position allows ArtMatic Voyager to simulate a wide variety of light conditions.

Sun/Atmosphere mode pop up

Sun/Atmosphere pop up menu is located to the right of the Sun direction sphere. It will set various aspects on how the atmosphere and sun are rendered. The modes are as follow:

- **sharp sun:**
 The default mode, with a sharp disk rendering of the sun.
- **strong halo, fuzzy sun, red shift:**
 Atmospheric halo is stronger in this mode which includes red shift scattering when the sun is low on horizon.
- **extra bright sun, red shift:**
 The sun is rendered as a smaller but brighter point and the mode includes red shift scattering when the sun is low on horizon.
- **soft halo, no sun:**
 A softer halo around an un-rendered sun. This mode is suited for use with ArtMatic 360 or ArtMatic Backdrop sky modes.
- **no halo, no sun:**
 The no halo, no sun option is often desirable when using the ArtMatic 360 or ArtMatic Backdrop sky modes; in particular when the ArtMatic Environment systems already shades the sun with an alternate representation.

For example you may have a sun with rays and a rainbow ring around. The Voyager Skies library provides many examples of custom skies shaders.

- **Automatic with scattering:**
 This sky/sun mode offers a more realistic approximation of atmospheric scattering. It has red shift when sun is low, and the haze color tends to be less pronounced and more realistic because haze is sensitive to scattering direction. The sun is rendered like in 'extra bright sun' mode.
 Note that Atmospheric scattering hue can now be changed to non-oxygen worlds in the "Environment settings" inspector.

Cast shadows (checkbox):

When this option is on, the surface will cast shadows on itself, and any clouds will cast shadows on the ground. This greatly increases the realism of the scene (and also rendering time). This setting is ignored in Draft quality (selecting this while in Draft mode will automatically switch Quality to Good). Computing shadows is computationally intensive and turning this option on can cause rendering to take many times longer than when it is turned off. Terrain self shadows have been improved in voyager CTX 1.5.

The render quality setting affects the accuracy of the shadows. When the setting is *better* or *best* or *sublime*, Voyager uses very refined sampling which can take a very, very long time but will yield the very best results. You may want to start with good quality and increase the quality if the shadows seem wrong or incomplete.

Haze slider & haze color

The haze setting determines the amount of atmospheric haze (caused by moisture and suspended particles in the air). With low values, the haze is only visible at a distance. With high values, visibility is reduced dramatically. The Atmosphere Height preference also affects the density of the haze and determines how high the haze will go.

The haze color can be changed by clicking (and holding) the mouse on the **Haze Color** picker. ArtMatic Voyager models the scattering of light that causes a blue shift of dark colors and red shift of bright colors that increases with distance. Changing the haze color has a dramatic impact on the way that colors are attenuated and shifted in the distance. Setting the haze color to gray enhances the red shift. In Voyager (just as in reality), a low sun will appear more red than it is.

Note that the Atmospheric Color Shift preference (in Environment Settings) also has an influence on how distance influences color.

Ambiant slider

This slider controls the amount of ambient light provided by the diffuse scattering of light from the sky dome. The ambient light is an overall illumination predominantly provided by the sky when in Automatic mode, and is non-directional (unlike the sun's). When ambient light is strong, the colors of the sky will "bleed" noticeably on the surface. Ambient light can be eliminated by setting the slider to its minimum value (0). Without ambient light, areas that are not directly lit by the sun will be very dark.

The Ambient slider affects the level of ambient for all objects and terrains in the scene. Objects and sprites have their individual ambient levee and the maximum is taken between the global and particular ambient level value.

The Environment settings inspector offers five options for the ambient light mode:

- **Automatic:** Automatic is the mode used in ArtMatic Voyager 3 and earlier. The ambient light is a blurred low-res image (128*64) derived from the scene's environment map and is recalculated for each frame. The object or terrain normal set the location in the ambient map.
- **Image:** When you use an image as the light map, it is resized to 128*64 and blurred. The image is stored in the Voyager scene's bundle as AmbientMap.png. The object or terrain normal set the location in the ambient map.
- **Image+Constant:** A constant color derived from various environment color is added to the chosen image.
- **Constant:** Constant mode uses a mix of the sun, haze and sky color to create a uniform ambient light.
- **Constant white:** Introduced in Voyager 1.5 "Constant white" provides an uniform non directional white light. It is often used with 'Balance light energy' activated (set in 'Preferences') so that the role of ambient is stronger and neutral.

Wetness slider

This setting determines how much the sea and snow and areas near the shore reflect the sky color. It also accentuates the water's reflectivity. When the wetness is high, you get fuzzy specular reflections in the areas affected by the setting. It also may change 3D object wetness, as the renderer takes the maximum of the global wetness setting and the object own wetness setting.

Atmosphere height slider

The Atmosphere Height sets the altitude at which atmospheric gases become insignificantly small. It affects the density of the haze and determines how high the haze will go and how much scattering is

going on in the sky shading model. The slider range is between 1 meter and 8000 meters. Note that when no Planet mode is used there is no longer any Atmosphere.

sea (level) & sea color

Sea level determines which elevations are filled with water. It is expressed in meters. Everywhere below sea level will be covered with water. Use the color picker found next to the slider to choose the sea color. In scenes where most of the sea color comes from the reflection of the landscape and sky, use a dark color for the sea, since the sea color is added to the total light coming from the water and can easily get too bright if its color is too light. The sea color is also modulated by the depth of the sea. It is darker where the water is deeper.

Note: the Camera is not allowed to go below the sea level.

roughness slider

This slider controls the surface texture of the sea by controlling the wind speed, which makes the sea rough. The wind speed also affects the speed with which clouds move automatically when time flows. Roughness influences the sea reflectivity and usually the amount of foam as well. Rough seas are less reflective than calm seas. The sea appearance can be modified with the various options available in the Environment settings inspector.

Transparency slider

This slider controls water transparency. The appearance of underwater features is also influenced by the Underwater Color Shift setting in the "Environment settings" inspector .

snow (level)

Snow level determines the elevation above which the planet terrain is covered by snow. The amount of snow is influenced by the steepness of the terrain and the elevation above snow level. The range is from -500 to 10,000 meters. You can generally eliminate snow by setting the snow level to 10,000 meters since it is rare that you will have peaks that high.

Note that snow level has no effect on DF based surfaces or objects.

Environment Settings...

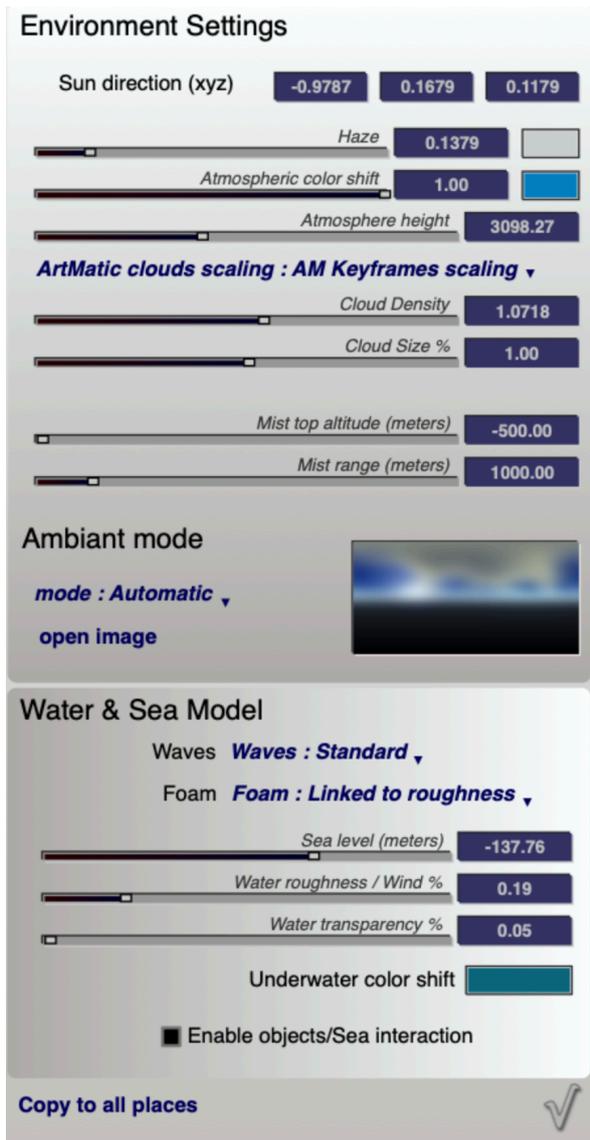


Opens the [*Environment Settings Inspector*](#).

The inspector provides higher precision sliders and numeric fields to set parameters that are often present in the main UI. For example you can set the sun direction more precisely than with the sun ball controller.

It gathers Context variables and global scene parameters that are independent of places and keyframes like the "Atmospheric color shift".

Some rendering modes, like how ambient light is computed, or the behavior of the sea, can be adjusted in there as well.



Sun direction vector: Sets the Sun direction numerically. Note that the final vector is normalized internally.

Haze / Haze color: Color picker to set the haze color.

Atmospheric color: Color picker to set the limit color of atmospheric scattering. It can be more or less saturated and hue shifted. Luminosity has no effect.

Atmospheric color shift: Sets how much hues are shifted by the atmosphere in the distance. The sky color is sensitive to this setting.

Atmosphere height: Sets the atmosphere height in meters.

ArtMatic clouds scaling:

Picker to choose between Absolute and keyframed scaled clouds

Mist top altitude (meters): This global scene parameter sets the altitude at which mist vanishes.

Mist range (meters): This global scene parameter sets the height of mist. Below Mist top altitude, Mist range the mist is very dense. Note that the mist level is further modulated by the haze amount.

Ambiant Mode: open image. Use this button to load your own ambient maps images.

Water & Sea Model:

- . **Waves picker:** Set the type of sea waves.
- . **Foam picker:** Set the sea type of foam. You can disable foam whatsoever using "never".
- . **Sea level (meters):** This context variable sets the sea level
- . **Water roughness / Wind %** This context variable sets the wind speed and hence the sea roughness. It affects the speed of built in clouds as well when animating.
- . **Water transparency %** This context variable sets the sea transparency.
- . **Underwater color shift:** Color picker to set the limit color of underwater absorption.
- . **Enable objects/Sea interaction:** Foam interacts with objects in sea.
- . **Copy to all places:** This button duplicates the environment setting to all recorded Places. It is useful in particular if you need to force Context variables to be identical in every Places.

Places Display



Places provides a simple and easy way to store locations on a planet along with the entire Context variables of Voyager. You may use Places not only to remember and go back to places already visited, but also to store atmospheric and lighting conditions.

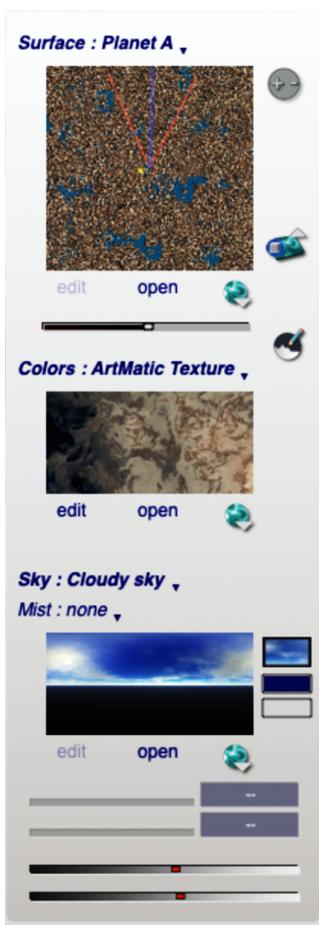
+ **Add place:** Saves the current Context variables into the first available Places slot. Clicking an empty slot has the same effect.

- **Delete place:** Deletes the selected active place. As with keyframe you can use option click to delete a particular slot, selected or not.

Refresh

Not all changes in Voyager will re-render all places preview thumbnails. 'Refresh' is useful when global changes make the preview no longer accurate. All places thumbnails will be re-rendered according to the latest settings.

Planet and Sky Settings Area



This area provides most controls for setting up the current scene: the Surface that defines the planet, the Texture that shades the planet and the Sky settings that handle sky rendering. Various UI elements and sliders may appear depending on the chosen modes for Skies and Textures.

Planet Surface section (top):

Map Overview: The surface map shows the portion of the planet that surrounds the current camera position. The default dimensions of the area covered by the map can be set with the Map Scale (radius) button below. The map reflects the current sea and snow level settings as well as the surface color mode. Click and drag on the Map Scale controllers to zoom in or out on the map.

The red lines indicate the view visible at the current zoom setting. The blue line indicates the current compass heading.

Click on the map to move the camera to clicked location. You can also click AND drag while keeping the mouse down to move the camera on the Map: the main view will render the camera view in fast preview mode interactively.

You may even go outside the Map area while dragging which gives you a much larger virtual region to explore. This offers a very efficient way to position the camera.

Notice that the latitude and longitude absolute camera position are displayed in the Tool Tips region at the bottom of the screen while

dragging and on mouse overs.

You may set the "snap to ground" preference to ON if you want the Camera to automatically stay near the ground when using the Map to position the camera (set in 'Preferences').

Map Scale (radius in km)



Click on the Map Range control and drag left or right to increase or decrease the area visible in the MAP overview. You can zoom in to see fine details of the landscape or zoom far out to get an overview of the planet's terrain. As you drag, the Tool Tips area displays the dimension visible in the map overview. Large ranges (100 km. or more) are useful for seeing large-scale features and to facilitate large location jumps (by clicking on the Map overview). Smaller ranges are useful for fine camera adjustments or to navigate inside a DF volumetric city.

The default range of the map will define the amplitude of the various relative camera sliders as well as the amplitude of motion when you drag the main view.

Surface mode pop up (text):

Use the Surface mode popup menu to choose a planet surface. You can use one of the built-in planet models, an ArtMatic file, or combination mode which allows several ArtMatic systems to be combined with each other and/or a built-in planet. Many examples of various types of terrains are provided in */Voyager Examples/Terrains & Landscapes/* folder.

The available modes are:

- **Planet A:**
Built-in planet A has a lot of small scale variations that result in dramatic changes over short distances. On a large scale, this planet tends to repeat itself and does not have very high mountains nor the large oceans of Planet X or Planet C . It is quite fast to render as its model is the simplest of all.
- **Planet B:**
Built-in Planet B has larger-scaled features than Planet A. Planet B often displays Nevada-like wide valleys with lovely rock clusters and high surrounding mountains. Terraces and steep canyons alternate with rocky shorelines and lake- filled valleys.
- **ArtMatic surface:**
This mode uses an ArtMatic file to define the planet geography with animation disabled. The planet's terrain will be defined entirely by the ArtMatic tree using the current parameter values. Learn more about ArtMatic Surfaces design.
Examples in */Voyager Examples/Terrains & Landscapes/ArtMatic Terrains/*
- **ArtMatic animation:**
This mode uses an ArtMatic file to define the planet geography with animation enabled. The planet's terrain will be defined entirely by the ArtMatic tree using the current parameter stored in the keyframes. The time slider will then animate the terrain provided the keyframes have varying parameters.
- **Planet X:**
Planet X has larger features than planets A and B. It may require traveling 500 km for the landscape to change dramatically. Large oceans and high-altitude plateaus alternate with rough shorelines, gently rolling hills, rocky deserts, lake and river districts, sand dunes, rock fields and large mountain ranges.
- **Planet C:**
Planet C is the most sophisticated planet algorithmically and has the largest scale features of

the built-in planets. More diverse and geologically realistic than the others, Planet C features large oceans, huge beaches, stratified canyons, sand dune deserts, river systems, faults and huge parallel mountains ranges. The highest peak can be over 10 000 meters.

- **Planet D:**
Planet D is a planet with diverse topography that includes beaches, deserts, grasslands, and polar ice caps.
- **Combination:**
Combination Mode allows you to modify the main planet (which can be either a built-in planet or an ArtMatic file) with up to 6 ArtMatic files. The ArtMatic files can be combined in many different ways with the main planet.
See how to build a planet using Combination mode.
Examples in */Voyager Examples/Terrains & Landscapes/Combination scenes/* and */VY5 Combination/*
- **No Planet:**
The 'no planet' mode completely removes the terrain and the atmosphere and is suited for deep space scene, often using a 360 environment map sky mode.

Elevation gain slider

The Elevation gain slider (under the surface map overview) controls the surface amplitude, which acts as a multiplier to the elevations found in the planet's source definition. Setting the amplitude high exaggerates surface features. Setting the value low flattens the features. Default is at 1.

Negative surface amplitudes invert the surface and turn mountains into canyons or oceans.

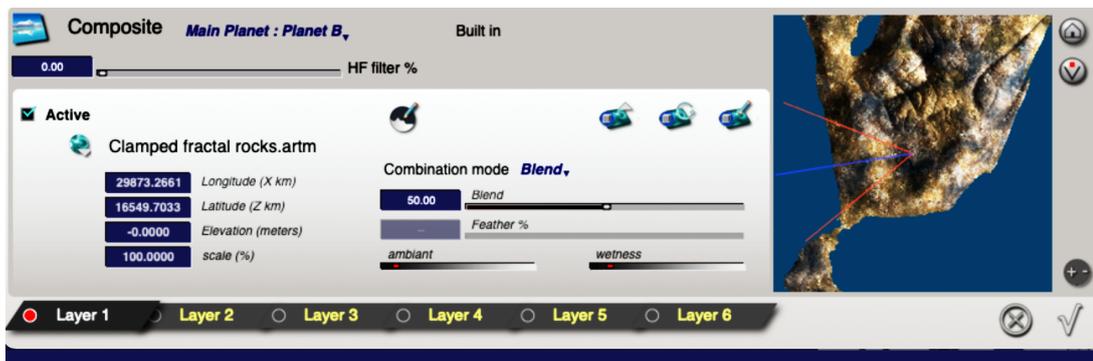
Open ArtMatic terrain:

Select an ArtMatic file to be used for the current terrain. The surface mode will be set to ArtMatic Surface unless already set to ArtMatic Surface or Animation. The ArtMatic system that defines the terrain should be a 2D (2 inputs) tree with 1 output (terrain elevation) or 4 outputs RGBA (RGB color + elevation).

Edit ArtMatic terrain

Opens the current 'ArtMatic Surface' or 'Artmatic Animation' terrain in ArtMatic designer for further editing.

In 'Combination mode', "edit" will invoke the Combination mode dialog.



Browse Terrains library



The Voyager Library provides a collection of preset terrains and planets directly available with the browse pop up. The scaling mode will be set automatically according to which folder is used. 'Absolute' mode is recommended.

You may add your own terrains in the following folders but keep them consistent. A non-texture surface function should be put in Absolute Surfaces, while a fully textured RGBA planet should go in Absolute Colored terrains or Absolute Worlds.

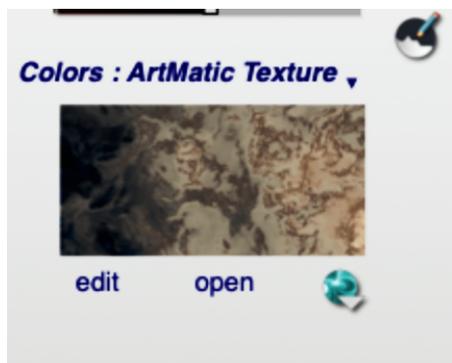
Absolute Bedrocks	>
Absolute Colored Terrains	>
Absolute Surfaces	>
Absolute Worlds	>
Colored Terrains	>

Add 3D ArtMatic DF object...



Usually management of DF objects is done within the Objects inspector. This button was kept for convenience to import a DF object from the main UI.

Planet Textures section:



Colors mode text pop up: The colors mode determines how Voyager texture-shades the terrain. The options are:

- **Default:**
When Default is the color mode, Voyager will use a color texture associated with the surface mode if there is one. If the surface is supplied by a four output (RGB+Alpha) ArtMatic system, the ArtMatic system's RGB output is used. If there is not an associated color texture, Voyager will use a general-purpose naturalistic shading function.

The built-in planets each have their own complex color texture functions associated with them that choose colors based not just on elevation but also on the local geology. For instance, Planet B may have reddish mountains, gray rocky valleys, ochre terraces and canyons and dark rocky formations with patches of yellow sand. On Planet X, you will find

green-ridged hills, reddish terraced canyons and gray rocky shorelines. Planet C has the most elaborate default color texture mode: different materials and geologies each have their own color palette.

- **Altitude gradient:**

This mode uses the selected gradient to map elevation to color. The color on the left is used for low elevations and the color on the right for high elevations. The terrain's slope affects the color slightly.

The gradient's mapping is related to the current sea level. Changing the sea level displaces the altitude gradient up and down. When Altitude Gradient is used, the standard U&I gradient tools are available: a gradient editor, gradient library popup, and customizable gradient display.

- **ArtMatic Texture:**

Uses a static ArtMatic file for color texture mapping. ArtMatic color textures can be either 2D or 3D solid textures that use both ground position and elevation to determine the color. Animation of the texture is disabled in this mode.

- **ArtMatic Animation:**

Uses a static ArtMatic file for color texture mapping with animation enabled. In that case parameter changes stored in the ArtMatic file keyframes will be played back when Voyager time flows. You can also use this mode to control the texture by using the timeline slider in order to find the optimal settings for a scene.

Power tip : If you are unsure which value to use for a particular ArtMatic parameter, store a lower value in keyframe 1 and a higher value in keyframe 2. You can then use Voyager timeline to find the best setting. When there are only two keyframes, the time readout in Voyager will match ArtMatic's. Choosing the same time in ArtMatic will give you the precise value of the parameter.

You can also use keyframes to store a variety of parameter settings and use the timeline to select between a whole series of variations of the same ArtMatic system. The chosen time value is saved with the file so this method works well when working on still images. (Learn more on designing ArtMatic texture systems in ArtMatic Textures).

Open ArtMatic texture: Open a new ArtMatic file for texturing the current planet terrain.

Edit ArtMatic Texture... Opens the current ArtMatic texture (if any) in ArtMatic Designer for further editing.

Browse Textures library



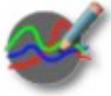
The Voyager Library provides a collections of preset textures directly available with this pop up.

Available folders :

- . Textures/Natural
- . Textures/Color & Bump
- . Textures/Rocks
- . Textures/RGB Alpha
- . Textures/MFD Textures
- . Textures/Multi Channel
- . Textures/Stratified

Textures can have several outputs. (Learn more about texture shading in ‘ArtMatic Textures Xouts and Xouts naming conventions’)

Edit gradient



Available in 'Altitude gradient' mode, this buttons calls the standard U&I Gradient editor.

Select gradient

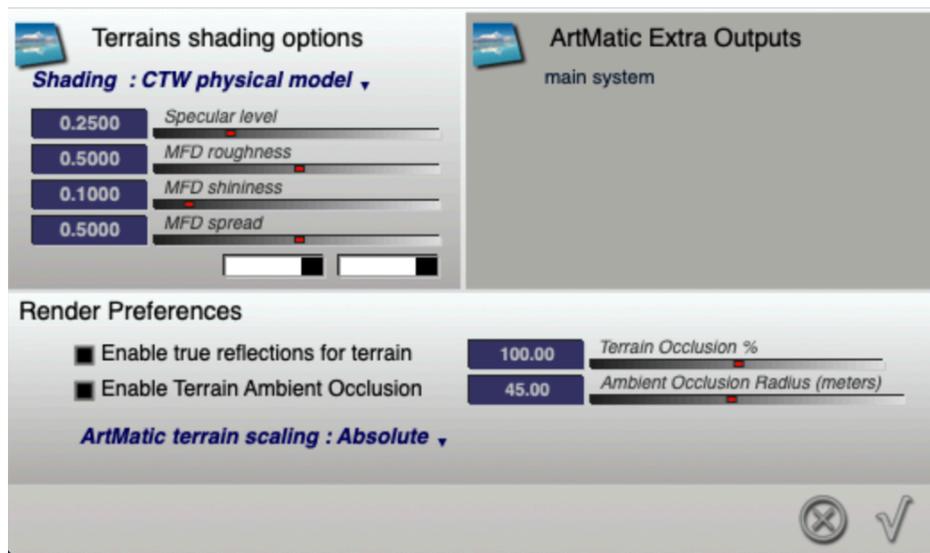


Available in 'Altitude gradient' mode this buttons lets you choose among a list of gradients.

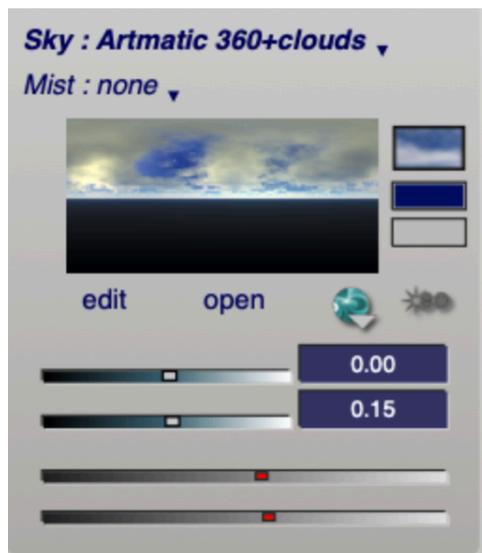
Terrain shader Settings... (t)



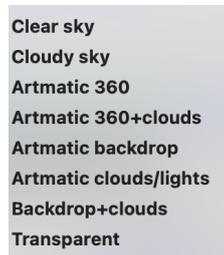
Invokes the Terrain shader settings dialog.



Sky Settings section:



Sky mode text pop up: The sky mode sets the various options to render and shade the sky.



Mist text popup:



To the right of the sky display are popup boxes for 'Sun & Clouds Render modes', 'Sky Color' picker, and 'Cloud & Ambient Fog Color' picker.

Below the sky display there may be one or more sliders visible. Their function is determined by the sky mode. The sky appearance is also influenced by the Sky Illumination Gain setting found in the Image section. This setting can have a dramatic impact on the sky appearance

The reference cloud elevation is set in the 'Position area' when mode is 'clouds & Sky'. It is the reference altitude at which layer clouds or volumetric clouds starts.

Sky Color picker permits setting the sky background color.

Cloud & fog ambient color: controls the amount of light emitted within the cloud by scattering, which is added to reflections from incoming lights. So when the "Cloud & fog color" is white or very bright you will have more total light. Set it to black to see the response that only depends from sun(s) light.

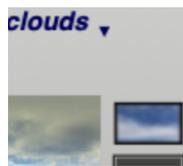
Alternatively you can set the sun to black and play with the Cloud & fog color to see how ambient light is transmitted through the cloud. The role of "Cloud & fog ambient color" is more important. You can shade a cloud only through scattering (light coming from every direction) using a bright "Cloud & fog color". In that case the thickness of the cloud will determine how much light is blocked by the cloud particles. The Sky Illumination Gain control affects only the directional light reflected from the sun(s).

So in general darkening the "Cloud & fog color" and changing the Sky Illumination Gain is the way to adjust. When the two are high the cloud may emit too much light. In extreme case you may need to use the global illumination gain & Gammas to further equilibrate the image.

Cloud Density and Cloud Size Sliders: To control clouds in various forms, Size and Density sliders become available when needed.

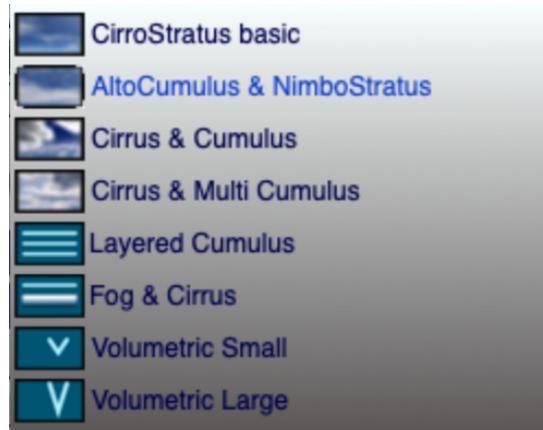
Sky angle and/or offset sliders: to change portion of sky viewed, when needed.

Sky & clouds render modes:



when a mode with built in clouds is selected – ‘cloudy sky’, ‘Artmatic 360+clouds’, ‘Artmatic clouds/lights’, or ‘Backdrop+clouds’ – this box appears.

Clicking on it will popup a **cloud types menu**.



Built in Clouds types:

- **CirroStratus basic:** A single layer of multi-fractal based stratus clouds.
- **AltoCumulus & NimboStratus:** A high layer of meckered alto cumulus and a lower layer of fuzzy nimbostratus
- **Cirrus & Cumulus:** One low layer of cumulus and a higher layer of cirrus fibratus.
- **Cirrus & Multi Cumulus:** Two layers of cumulus and a higher layer of cirrus.
- **Layered Cumulus:** Several layers of cumulus and a higher layer of cirrus.
- **Fog & Cirrus:** Several low layers of fuzzy clouds and a higher layer of cirrus.
- **Volumetric Small,**
- **Volumetric Large:** Volumetric clouds provide enhanced realism but require vastly more computation than the non-volumetric cloud modes. So, they take longer to compute than non- volumetric clouds. Realism is maintained even when the camera passes through a layer of volumetric clouds. Clouds are represented internally as full 3D density fields.

Volumetric Small and Volumetric Large differ by the altitude ranges within which the clouds are rendered.

Small computes clouds in a range of 4,000 meters above the elevation set by the Cloud Layer Height slider. A very high-layer of built-in (non-volumetric) cirrostratus clouds is also added.

Large computes clouds in a range that extends 10,000 meters above the Cloud Layer Height setting.

Volumetric Small clouds compute faster than Volumetric large since the cloud layer's is less than half as tall. Due to the nature of the volumetric rendering, *draft quality* mode gives a poor approximation of the clouds. You should use good or better mode or getting an idea of what the clouds will look like.

If you set the Cloud Layer Height to 0, you may end up inside the clouds with an obstructed view. If this is the case (and you want the clouds to start at elevation 0), adjust the cloud size and density sliders until you are able to see, or simply move the entire cloud layer using the position sliders.

Sky Modes (text popup):

- **Clear sky:**
This mode is useful both when you want a clear sky or when you want to speed up screen redraws. The sky background color will be the most determinant parameter of the sky rendering, outside the haze and Atmospheric height. In "no Planet" mode you can use "Clear sky" and the sky background color to render a specific object over a simple discreet colored background.
- **Cloudy sky:**
When Cloudy Sky is selected, the Cloud types popup menu becomes available and lets you pick the type of built-in clouds provided by Voyager. Two sliders will be visible to influence the clouds: Cloud Density and Cloud Layer Scale. They operate both on layer based clouds and volumetric based clouds. In general layer clouds are much faster to render than volumetric one that require many samples to be rendered and shaded.
- **Artmatic 360:**
This mode uses an ArtMatic file as a 360 degree panoramic backdrop also called an environment map or sky dome. When using a properly-designed ArtMatic system, there will be a seamless sky as you survey in all directions.

In this mode, the *Link Environment to Sun toggle* appears, and the slider controls the horizontal offset of the ArtMatic system. The slider can be used to rotate the sky horizontally up to 360 degrees. When Link Environment to Sun is turned on the ArtMatic 360 environment is linked to the sun position so that it moves with the sun when the sun position changes.

When using ArtMatic 360 mode, it is imperative to use a system designed for use as a 360 degree sky environment map. The Voyager Library provides an extensive collections of 360 environment and the simpler is to use the browse pop up to select one.

- **Artmatic 360+clouds:**
This mode combines Voyager's built-in clouds with an ArtMatic 360 sky (see details above). The sliders control Cloud Layer Height and Cloud Density. It is possible to rotate the background by switching to ArtMatic 360 mode, using the slider to rotate the backdrop, then switching back to ArtMatic 360+Clouds.
- **Artmatic backdrop:**
The chosen ArtMatic file is used as a 2D backdrop. The backdrop does not follow the camera motion and it is useful only for creating still images or movies where the camera does not rotate. Any ArtMatic file can be used.

The sliders provide a vertical and horizontal 2D offset control for the backdrop. Note that this offset is part of Context variables and can be animated.

A second RGB output component in the Artmatic backdrop tree will be shaded as an additive overlay by compositing the output at the final stage of rendering. It provides a mechanism to add image-space after-effects like lens flares, rain, graphics overlay etc. Only one extra output is supported in this case and the global inputs A3 and A4 provides 2D view or Sun position in camera image space. Voyager fills A3 & A4 global input to inform ArtMatic about camera projection center (or sun position when "link to sun" is on). If you add in the ArtMatic system these master input to the XY coordinates the AM image will move along the Voyager camera movement. This will make the backdrop follow the camera like with 360 mode. When *link to sun* is active the AM system will be centered around the Voyager sun position and can be used for lens flare effects or custom sun shaders.

Examples : ../Voyager Examples/Shading & Rendering/Backdrops & image effects

- **Artmatic clouds/lights:**

This mode use an ArtMatic system to define the mathematics of clouds or volumetric lights that participate in the sky rendering. With this mode you can define yourself the density function that will create clouds using the hundreds of mathematical and procedural noise functions available in ArtMatic Engine.

Clouds in Voyager are made using density fields that describe the density of water particles at any point in space. Density fields are, like the distance fields used for 3D DF objects modeling, particular cases of scalar fields - a scalar field is simply a set of one dimensionnal values associated to every point in space, whatever the significance of the value. A density function can be used for cloud layers as well as volumetric clouds. It can also describe light density for special effect shaders.

A special mode "Underwater shader" provides not only density information for underwater volumetric shading, but also detailed information influencing water surface texture, color and reflectivity. In that case the first cloud layer becomes the water level surface and the atmospheric model changes to be more approximative of an underwater situation.

Whatever the sub modes are, the origin of the ArtMatic sky can be set in the Position area in Sky and Clouds mode.

Artmatic clouds/lights modes :

- **Cloud Layer:**



The ArtMatic defined density function is rendered as a single or dual cloud layer. Layers are flat and very fast to render as the cloud density function is computed only once when the ray intersects the plane coordinates. The number of outputs of the Artmatic tree determines how Voyager uses the system to create clouds. When the ArtMatic system has one output, the ArtMatic system defines cloud density and the clouds are rendered using algorithms similar to those used for Voyager's built-in clouds. If the ArtMatic system has two outputs, the outputs are treated as two cloud layers at different altitudes. The second output is the upper layer and appears brighter than the lower layer. If the ArtMatic system has three outputs, they are treated as RGB outputs, and the ArtMatic system will define the entire sky with Voyager performing no cloud shading. If the ArtMatic system has a four-output component at the bottom, it is treated as an RGB+Alpha sky where the alpha output is used as the cloud density function.

- **Multiple Cloud Layer:**



Same mode as above but with multiple layers. In this mode, two clouds layers are generated from the ArtMatic system even if the structure only provides one output component. If there is only one output component, ArtMatic Voyager will use it to create two identical cloud layers 5000 meters apart. If there are two parallel output components, the leftmost will be used for the lower cloud layer and the other for a layer 5000 meters higher. Additionally, there will be a built-in cirrus layer at high altitude. This mode is intended for use with ArtMatic systems that have 2 or 3 inputs and have either scalar (single-output) or RGBA output. The scalar Global Input Note. When Voyager is rendering clouds (but not Backdrops or ArtMatic 360), only global input A2 (the cloud's absolute height value) is defined. The other ArtMatic global inputs (A1, A3, and A4) are not defined during cloud rendering and so should not be used.

- **Volumetric Small:**



Voyager features 3 ArtMatic-based volumetric cloud options : small, large and unbounded. Volumetric Small restricts the volumetric clouds to a region 4000 meters tall by forcing the density function to become negative below reference cloud elevation and above the given height relative to cloud level.

While you can use any combination of functions to create the cloud density function it is recommended that you use 3D systems to have real 3D volumetric density data at the end. Avoid to use overly complex functions in the cloud definitions as the rendering can become very slow. The ArtMatic system can be scalar (one output) or RGBA. In addition to the Volumetric clouds, ArtMatic Voyager adds a high layer of built-in non-volumetric Cirrostratus clouds.

- **Volumetric Large:**



Volumetric Large clamps the density function to a region approximately 8000 meters above reference cloud elevation, that is twice taller than volumetric small. The bigger the volumetric clouds the more computation will be needed to render them. In this mode you can create high cumulus clouds. In

addition to the Volumetric clouds, ArtMatic Voyager adds a high layer of built-in non-volumetric Cirrostratus clouds.

- **Volumetric unbounded:**



Introduced in Voyager 5 'Volumetric unbounded' doesn't clip the 3D cloud density function upwards and doesn't add any high layer of built in clouds. "Volumetric unbounded" still clips the density below reference cloud

elevation to avoid camera being completely obscured at ground level. The entire sky above cloud elevation is yours and you can build clouds that goes up to the stratosphere, have many volumetric layers, have column of smoke or whatever is possible with a density function.

- **Alpha Layer:**



In this mode the ArtMatic RGB channel is blended with the Voyager scene according the tree output alpha value. No cloud shading takes place and the color of the layer is entirely controlled by the ArtMatic system. Alpha Layer is usually used for special effects or non realistic graphic animation.

- **Additive Layer:**



The ArtMatic output is blended with the Voyager scene in additive mode which makes it appear as a glow that does not cast shadows or provide illumination.

- **Additive Multi Layer:**



Multi layer version of Additive Layer.

- **Volumetric light:**



This option treats the ArtMatic system as a density field which is interpreted as a additive light source. Generally (but not always), you will use the *31 Density Shapes* component as the heart of a volumetric light system.

This mode requires a true 3D ArtMatic system (i.e. one in which the X, Y and Z global inputs are all used). The system is interpreted as a 3D light source. Unlike volumetric clouds small and large, there is no upper lower limit to the altitude range used to interpret the density function. So the ArtMatic system must fully define the behavior of the density field.

The density function may be clamped so that high densities don't produce a global wash out of the rendering in cases where the density adjustment slider is not enough. Volumetric Lights provide an illumination from a point light source that emanates from the sky origin coordinates. To set the position of the light system use the Position area in Sky and Clouds mode.

- **Underwater shader:**



The Underwater Shader mode is designed for creating underwater scenes as well as interesting volumetric light effects that are faster to compute than when using volumetric light mode. The Underwater mode can also be used above-water for special lighting effects and alternate sea shading. ArtMatic Voyager comes with several ArtMatic presets systems that implement underwater shaders that can be used by anyone regardless of their expertise.

Creating new underwater shaders from scratch requires a fairly solid understanding of how ArtMatic and Voyager work. Underwater mode is not restricted to underwater effects. When the camera is above the water level, you can create a host of interesting effects which include atmospheric gases, bubbles, rays and customized Water surface rendering. The water surface of "underwater mode" water is opaque, but the ArtMatic shader can influence the look of water surface in many ways. Underwater mode sets the haze stronger underwater and mixes the haze color with the underwater shift color. The haze is used even in No Planet mode which allows you to have a bottomless ocean. The higher the water level the more the haze color is shifted towards dark blue. The current depth (cloud level minus terrain altitude) below is sent via global input A2). You can use this feature to design a water model where the color/waves/foam change with proximity to the shore.

Underwater ArtMatic systems typically will have several outputs:

- *Output 1 RGBA*: water surface + volumetric light effects + caustics. Surface color can be modulated with depth (global input A2) to achieve pseudo transparency.
- *Output 2 scalar*: True reflection amount. Enable true reflections for terrains must be ON for the water surface to reflect.
- *Output 3 RGB (optional)*: Additional light that affect water surface shading only. Common use is to enhance foam brightness.
- *Control Sliders*: The cloud sliders have the following meanings in underwater mode : ‘Cloud Density’ - controls the intensity of the volumetric light effects. When it is set to 0, the volumetric light effects are turned off, but the water surface is still shaded. ‘Water Surface Height’ - This controls the elevation at which the water surface appears. ‘Cloud Size’ - This slider scales the entire ArtMatic underwater shader tree.

- **Backdrop+clouds:**

Backdrop+clouds works as the "ArtMatic Backdrop" mode but with the addition of built in clouds.

- **Transparent:**

This mode makes the sky transparent and is useful when an object or terrain has to be rendered on a transparent background.

- **Open ArtMatic Clouds or Environment texture:** Use this button to import a new ArtMatic Sky file.

- **Edit ArtMatic Sky...** Opens the current ArtMatic Sky file (if any) in ArtMatic Designer for further editing.
- **Browse Skies library**



The Voyager Library provides an extensive collection of Skies, Clouds and & 360 environments that can be imported directly using the Browse pop up.

Absolute Clouds	>
BackDrops	>
Custom Suns	>
Environments 360	>
MultiLayer Clouds	>
RGB Sky plane	>
Scalar clouds	>
Underwater shaders	>
Volumetric clouds	>
Volumetric lights	>

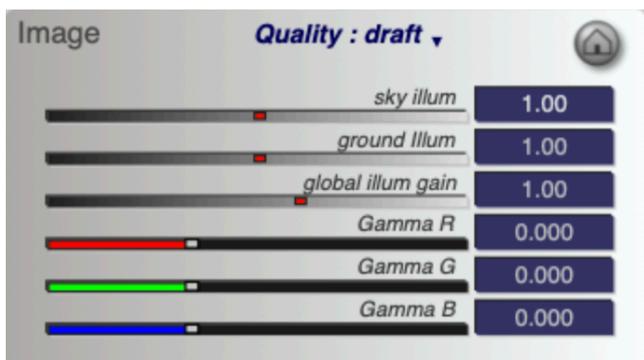
Folders are organized by themes. In general, choosing from these folders will automatically set the sky mode to the proper mode and eventually set cloud type and cloud scaling mode as well. 'Custom Suns' folder holds alternate sun shaders. When choosing from this folder 'link to sun' is activated automatically.

- **skycontrol sliders:** Available in Backdrop and 360 Environment mode these sliders offsets or rotates the sky coordinates.
- **Clouds Density slider:** Controls an offset to the cloud density function. It affects every kind of clouds and can be used to make the sky completely overcast and will growth volumetric clouds as well.
- **Clouds Size slider:** Controls the overall size of the built in clouds or ArtMatic defined clouds/skies.
- **Edit current ArtMatic system**



Available with ArtMatic defined clouds/skies this button will open the ArtMatic tree in ArtMatic Designer for deep editing.

Image Settings Area



The Image Settings area collects all controls that globally affect the rendering: the quality setting, various illumination gains and the gamma sliders.

The Gamma filter is applied at the last stage of the rendering while the illumination gains are taken into account at the shading phase of various elements.

These settings, except the quality setting, are part of the current Context variables and are thus both keyframable and can be stored in Places.

- **QUALITY text pop up:** This popup menu determines render quality (of both the image on the canvas and any pictures or movies rendered to disk). Generally, you will use Draft quality while exploring and switch to a higher quality setting when rendering images and animation to disk. The higher the quality setting, the more computation that Voyager must perform in order to calculate the image and the longer it takes for Voyager to render the image.

In some cases you may encounter strange artifacts at the lower quality settings, due to insufficient sampling steps, in particular for terrains with steep peaks or tiny features (a tiny steep spike can fall in-between samples and be missed). A higher quality setting will increase the amount of samples and will make feature misses less likely. When performing draft renders, you may want to use the lowest quality that provides acceptable results.

For final animation rendering, it is recommended that you use *Better* or *Best* quality, as misses can cause flickering where fine details occur, and make small peaks appear and disappear unnaturally.

NOTE: When Draft quality is selected, Voyager ignores the cast shadows setting.

- **Reset color filter:**



resets color filter parameters to nominal default (Gamma and global illum gain at 1).

- **Sky Illumination (Gain):** Controls illumination gain for sky shaders only. It mostly affects clouds and atmosphere shading. This slider is often used to enhance the clouds contrast or to make them darker if the various illuminations make the clouds too bright.
- **Ground Illumination (Gain):** Controls illumination gain for terrain and objects only.
- **global illumination gain:** global illum gain controls overall illumination gain. It is rarely needed but can be used to compensate for particular lighting conditions. High gamma values will darken the image while adding contrast, so it might useful to compensate with the global illumination gain.
- **Gamma R (Red color filter):** The gamma sliders use an exponential range where 0 means no change is taking place, 1 means power 16, -1 means power(1/16). Gamma values above average (0) will contrast the image. Using gamma can greatly enhance the visual impact of a rendering and may also be used to control the color balance of your image.

Note that Voyager renders colors in 64 bits per component, and the gamma and range adjustment is done before final 8 (or 16) bit quantization, allowing a much better accuracy and quality than doing these adjustments as a post process in photoshop or any graphic application.

TIP: Shift-click the Red gamma slider to move all the sliders together.

- **Gamma G (Green color filter):** Green channel of the Gamma filter.
- **Gamma B (Blue color filter):** Blue channel of the Gamma filter
-

Popup Inspector Rooms Area

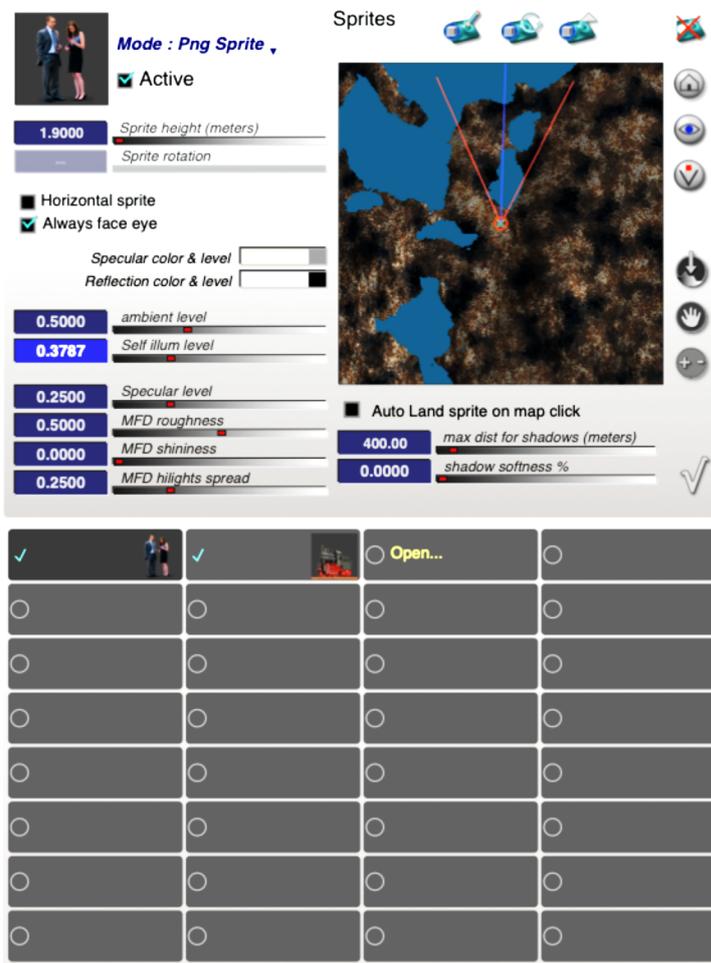
These buttons popup ‘Inspector’ windows allowing import and control of 2D Sprites, 3D Objects, and new lights.

Sprites inspector (x)



Opens the Sprite inspector. Sprites are very fast to render and can provide a lot of cool visual effects for animation when using animated ArtMatic systems (eg. meteorites that cross the sky, various light effects, even reflective water flakes). Static PNG sprites can be used to add people in your architectural or landscapes scenes, a city in the background, or some trees in the foreground etc...

The Sprite inspector gathers all tools to open, scale, orient, activate and shade sprites. You can use the main Sprite menu to copy and paste sprites between different voyager scenes.



Sprites are 2D images rendered inside the Voyager 3D space.

Sprites can be a PNG image, or a 2D Artmatic system (animated or not).

Static PNG sprites will usually include a transparent alpha channel around the subject (eg. Person, tree, etc.) so it blends properly into the scene.

In general, when a 2D Artmatic tree defines the sprite, it should return an RGBA value with an alpha channel that limits the spread of the Artmatic image (unless you need an infinite image plane).

To incorporate an animated movie as a sprite, use '24 RGBa Pict/Movie' component in an Artmatic tree to provide an alpha value.

An Artmatic tree could contain an animated picture atlas. Or a sound sensitive tree could be animated by sound, using 'global matrix: audio input mode'.

Note: 'Sprite Height', 'shadow softness', 'ambient level', and 'self illum level' can be used with *SHIFT* to set the value for all sprites. *SHIFT activate* (or de-activate) will activate/de-activate all sprites.

DF objects inspector (o)



Opens the Objects inspector. The objects inspector gathers all tools to open, scale, activate and shade DF 3D objects.



You can use the main Edit menu to copy, add or paste objects between different voyager scenes.

To position objects, click on the inspector map, and/or use the 'position area' controls.

If object's Artmatic tree is sound sensitive and uses the 'global input matrix: audio input mode', the object may be animated by sound.

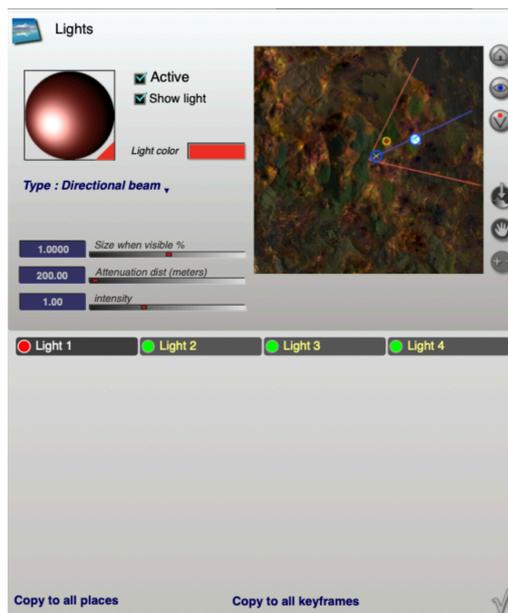
Modelling 3D objects is discussed in detail in "Building 3D Objects: DFRM Guide".

Nb: *SHIFT activate* (or de-activate) will activate/de-activate all objects.

Lights inspector (l)



Opens the Lights inspector. The Lights inspector gathers all tools to setup and shade built-in additional lights.



There can be three **Types** of light: *Point light*, *Additional star*, and *Directional beam*.

Lights are fully animatable and can be treated as additional suns, so Voyager CTX can now have multiple-sun worlds.

Lights positions and settings are context-dependent, so places and keyframes can have different lights.

Quick Edit ArtMatic parameters & shaders:



Opens the ArtMatic params & shaders inspector for Quick Editing within Voyager (a deeper ArtMatic Tree editing requires using ArtMatic Designer).



As ArtMatic trees are used for many things in VY you have (on the top-right of the inspector) a dynamic popup that lets you choose which tree you want to modify.

This inspector gives you a direct access to "Published" parameters from the selected tree and can set the shading options associated with Extra Outputs.

If no particular parameters were published it will present the first 6 parameters found in the tree.

ArtMatic Designer can "publish" up to 6 parameters from the tree, even if they are located deep inside Compiled Trees.

"Published" parameters are accessible for modification directly within this inspector by using the sliders, without having to open ArtMatic Designer. When a particular parameter is keyframed, you may either change the value with the slider, or with the envelop (*keep in mind that to see the result interactively, you need to have the time slider at a point that is close to the time of the changed parameter. Click on the lock icon to bypass locked keyframed values for the parameter*).

You can also **rename** the tree (click on the name and edit) which is often useful after modifying an existing one or to create variations (eg. after modification, or to create a variations without losing the original. The changed name is saved directly inside the Vy bundle. When using hot-linking with Artmatic Designer, it is necessary to re-open Designer after changing the name.

Positions Area



Every Voyager entity now has coordinates that can be moved using the new 'Positions' section in the main UI.

The old "ArtMatic sky coordinate" is thus no longer needed as clouds coordinates are now common for any kind of clouds, and can be directly changed in the 'Position' section when in "clouds and sky" mode. Even the main planet's terrain can be moved. Suppose you have a great sky at some point, but the foreground of the terrain is annoying. You can use the position sliders to offset the planet laterally or in depth; or the absolute coordinates numeric fields to completely change the terrain position. Inversely, you can have a great scene but the clouds cast an unfortunate shadow: Just move the cloud layer until the problem is fixed. Layer and VL clouds do cast a shadow even in fast preview low res mode, so you can interactively move the layer and see the shadows move.

Warning : Terrain and Texture position are global to the scene (only one value for all places and keyframes). Changing the position of the planet origin will make all saved Places and Keyframes invalid. Sprites and DF Objects position are also global to the scene and changing their position will affect all Places and Keyframes.

*Only **Lights** and **Clouds** position are part of the voyager Context variables that can be keyframed and stored in Places.*

"Mode" text pop up menu on the left sets the target of the position controls. Usually it is set automatically when editing a particular type of object but at times it may be necessary to set it manually. Options are:

- **terrains:** Sets the target to be the current terrain (either a built-in surface, an ArtMatic defined planet or a Combination mode terrain made of various sources).
- **terrains texture:** Sets the target to be the current terrain texture. Note that changing it will invalidate all saved Places and Keyframes.
- **objects:** Sets the target to be the current DF object. If multiple objects are present in the scene, you can select the target object in the Object inspector.
- **sprites:** Sets the target to be the current Sprite. If multiple Sprites are present, you can select the target in the Sprite inspector.
- **lights:** Sets the target to be the current Light. If multiple Lights are active, you can select the target in the Lights inspector. Light position is part of Context variables and can be keyframed.
- **clouds & sky:** Sets the target to be the sky element reference position. In general, this will be the clouds origin coordinate, but can be the position origin for volumetric lights or water surface level & in underwater mode. Clouds & sky coordinates origin is part of Context variables and can be keyframed.

Browse scene elements:



Objects, terrains and sprites used in the scene can be selected (and made target) directly using this pop up picker. Note that if a Sprite or Object is set as target the Camera home button will move the camera to focus on the selected object.

Edit current Artmatic system:



Opens the selected Artmatic system in Designer, ready for editing, if Artmatic Designer is already open.

Sliders:

- **Lateral displace:** The slider displaces the target laterally in the camera view space. With SHIFT pressed you can force a world space displacement in longitude (x axis only).
- **Depth displace:** the slider displaces the target forward or backward in the camera view direction. With SHIFT pressed you can force a world space displacement in latitude (z axis only).
- **Vertical displace:** The slider displaces the target vertically relative to current vertical position.
- **Elevation (meters):** Sets the absolute elevation coordinate in meters. Quite useful for setting cloud layers or various DF object altitudes. When target mode is 'clouds & Sky' this slider sets the reference cloud elevation, which is the reference altitude at which layered or volumetric clouds starts. In Underwater sky mode this will set the water surface level.

Elevation is just the y of Clouds & sky coordinates origin. As part of Context variables it can be keyframed.

Longitude (km):

Sets the absolute longitude coordinate in Km. Since the Voyager worlds are immense, setting absolute coordinates is rarely used, but it might be very handy to center or align various objects to a specific point in space. If you need to offset objects East/West independently of camera view this field is the answer, probably with *option key* down to avoid going too fast as the range is huge.

Latitude (km):

Sets the absolute latitude coordinate in Km. You may use this field (likely with *option key* down) to offset objects, clouds or terrains North/South independently of camera view.

Animation area



This area is focused on Animation controls. It provides the Keyframes UI, the main timeline slider and various buttons.

Nice animations can be created with a motionless camera by animating the sun position and colors and rendering with shadows on (be aware that turning shadows on increases render time dramatically). Moving clouds will cast moving shadows on the landscape, and the sun can set, with the shadows getting bigger and the colors reddening.

All the variables listed in 'Context Variables' can be stored in keyframes and thus animated.

- **Time slider:** This is the main slider that controls global Voyager time. The time flows from 0 to the given duration. You can use the time slider even when no keyframes are present, as many

Voyager elements are automatically animated over time. In particular, when any ArtMatic tree is used for texture, terrain, sky clouds, objects, it may have its own keyframes and will respond to time changes (*Keep in mind that the entire ArtMatic animation will always map to the Voyager timeline duration*).

You can click and drag on the time slider to preview the animation in non-real time, or simply click at a particular time to see a frame preview at that moment.

Tip: *The time always flows in Voyager, and the timeline can be used to select a particular position in time. This is useful because some elements (water waves and clouds, for example) move automatically at their own speed regardless of the animation duration, and you can use the timeline to find the perfect moment. Thus, a nice trick to change the appearance of the clouds is to set the global duration to 10 minutes or more with the watch icon and use the timeline to find the best cloud positions. In a 10 minute time span the clouds can change dramatically. You can also use the timeline to pick a particular moment within a sky, texture or surface ArtMatic Animation. This is a powerful way to find interesting settings since many parameters can be animated at once with ArtMatic keyframes.*

- **Keyframes:** Keyframes let you store locations and environmental parameters that can be used to render animation. Keyframes are not required for animation if you are using ArtMatic animation for the surface mode, or if you use preset clouds and water; whose movement is controlled by the 'Sea Roughness' setting.

The ArtMatic system's keyframes will be mapped so that all the keyframes play back over the course of the animation. When selecting a keyframe by clicking on it the keyframe Context variables will be copied to the current context, with the exception of parameters that were set specifically to not be animated within the Animation parameters inspector.

- **Add (keyframe):** Add a new keyframe with the current Context variables Shortcut: you can also click on the first blank keyframe to add a new keyframe.
- **Continue (keyframe):** Continue will add a new keyframe without changing the absolute time of existing keyframe, by changing duration accordingly.
- **Replace (keyframe):** Replaces selected keyframe with the Context variables. You may also use command click on a keyframe slot to perform the replacement. .
- **Insert (keyframe):** Calculates a new keyframe that is halfway between the selected keyframe and the keyframe after it.
- **Delete (keyframe):** Delete the selected keyframe. Shortcut: option-click any keyframe to delete it.
- **Duration (MSF)**



The watch icon is used to set the animation duration. Click and drag left or right to change the duration. Duration is displayed in MSF format (minutes, seconds, frames)

- **Play:**



Click on this button to see a realtime preview of the animation. The preview will be a low resolution approximation of the animation and will appear blocky, as even the fastest machines are currently much too slow to calculate a high resolution realtime preview.

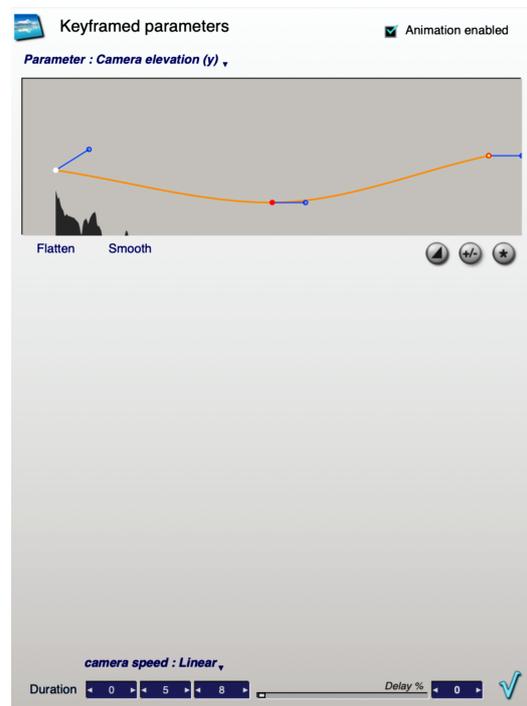
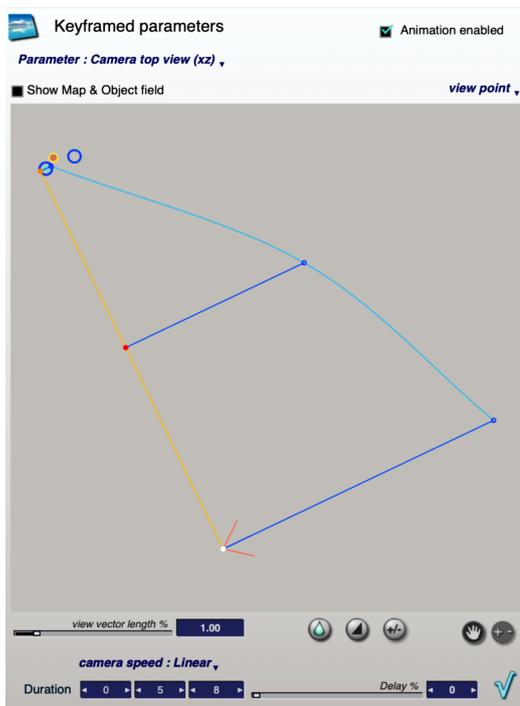
The preview won't show some aspects of the final rendering (shadows and reflections for instance). So, it is often useful to do several small renderings at low frame rates to fine-tune the camera motion and animation parameters before the final rendering that can take days or weeks.

Shortcuts: Press the *spacebar* on your keyboard to start and stop the animation preview.

- **Animation parameters inspector (a):**



Opens the Animation inspector window, if two or more keyframes are saved. Multiple *parameters* can be viewed.



Bottom Tools Area

Open ArtMatic Voyager bundle:



This command is the same as *File->Open* and will prompt you to locate a Voyager bundle.

Save Voyager bundle :



This command is the same as *File->Save* and will either directly save the current scene if the file already exists or prompt you to set a name to save the Voyager bundle

Render pict to screen:



Render the current scene fullscreen with anti-aliasing. As screens are quite large now days, this may take a long time, depending on complexity of the scene; so a smaller test rendering using the 'Render Pict to File' is often better.

Render pict to file:



Click this tool to render the scene as an image file. The **Render Picture dialog** opens to give you control over the image settings.



Size: You can choose dimensions up to 16000 by 8000 pixels. Renderings now always use dithering. Dithering introduces a tiny amount of color (RGB) noise to the image, which greatly enhances color accuracy and avoids color-banding.

NOTE: Voyager does not set a DPI in the rendered pictures which will probably be at 72 by default.

Render Mode: Single view, stereoscopic views.

Render: Current View, All places, All Keyframes.

Render All Places option will render a picture for each place stored in the file. Keep in mind that this can take a long time. To abort a render in progress, press the escape (*ESC*) key.

Anti-Aliasing:

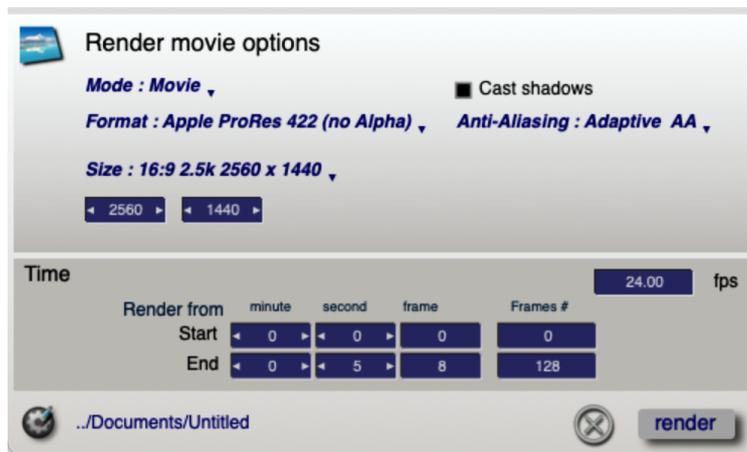
- **Standard 2*2:** Samples per pixel is set at 4. Default and Fastest AA mode but often insufficient for DF cities and textured DF objects.
- **Stronger 3*3:** 9 Samples per pixel.
- **Double 4*4:** 16 Samples per pixel. Recommended to avoid flickering, in particular with DF cities and textured DF objects.
- **Adaptative AA:**
The Adaptive anti-aliasing oversampling is done only when fast changes

in depth and color occur, which makes it faster than a brute N*N oversampling. For terrains, it works only when the camera is set to *perspective* and *spherical*, due to specific optimisation in the cylindric case. Its speed is quite variable depending on the scene, but it is often better and faster than the 4*4 oversampling.

Render Animation:



Click the Render Animation tool renders an animation. The file does not need keyframes to be animated. The **Render Movie dialog** opens:



Movies will take a long time to render. To stop a render, press the *escape key*. The duration of the complete animation is set with the Duration Tool found next to the timeline in the main window.

Mode popup (movie or picture sequence)- The options are:

- QuickTime Movie,
- List of Pictures (png),
- List of Pictures (Tiff).

The "List" options render the movie frames as sequentially numbered image files (in either PICT or TIFF format). Such sequences are recognized by most movie editing programs. Pict/Tiff sequences are a good idea when performing a long render since nothing will be lost if the computer shuts down unexpectedly. (movies will be unplayable if the render is interrupted).

Format popup: options include 'H264 (no alpha)', which is faster, and several 'Apple ProRes' versions, which may be slower but better quality.

Size Preset popup - The preset popup provides a list of common frame size/frame rate combinations. Choosing a preset cause the format and fps fields to be filled in with the appropriate values.

Planet Building

(Based on 'Building Planets' by Edward Spiegel and Eric Wenger)

A huge variety of landscapes can be found by exploring the built-in planets, changing elevation gain of the Surface, sea and snow levels, changing terrain textures from the preset library, etc. However, ArtMatic Voyager was designed to be even more extendable, by using ArtMatic Designer to create new terrain surfaces and terrain colorings.

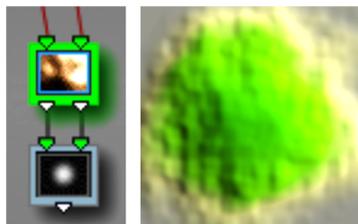
In Designer you can set up tree functions for creating planet-sized terrains or infinite ground textures.

Designing Terrain Surfaces

When the ArtMatic **Surface mode** is set to *Artmatic Surface* or *Artmatic Animation*, the chosen Artmatic system is treated as an elevation map by feeding the longitude and latitude of each point to the Artmatic system and treating the output as elevation.

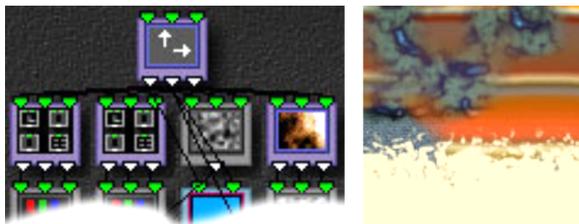
ArtMatic systems that are used with ArtMatic Voyager can receive either **2D input** or **3D** input from Voyager:

- **2D input trees** shall be used for terrains elevation functions. 2D input trees receive ground coordinates (x,z) from ArtMatic Voyager via global Artmatic inputs X and Y, and outputs the terrain elevation. Elevation-only trees provide no color information and the color texture, if needed, shall be provided by another tree.



A 2D-input ArtMatic Structure tree used as an elevation map. Global inputs X and Y represent latitude and longitude (ground co-ordinates).

- **3D input systems** receive both ground coordinates and elevation information from Voyager (for example from built-in Planets). In this case, the (x,y,z) are passed directly into (X Y Z) ArtMatic inputs, and the coordinates are used within the tree to compute color and eventually another elevation that can be used for bump mapping or terrain filtering.



A 3D-input ArtMatic structure tree and the color texture that it creates. In 3D-input trees, Global Input Y represents elevation. In these trees, the ground co-ordinates are passed via Global Inputs X and Z.

3D input systems are typically used for the color-texturing of terrains. Voyager (as with most 3D computer graphics applications) uses x and z to refer to the ground co-ordinates, and y to refer to the elevation (height). When the tree returns both a color as well as an elevation, the **color texture mode** must be set to *Altitude Default* in order for the tree specific colors to be used.

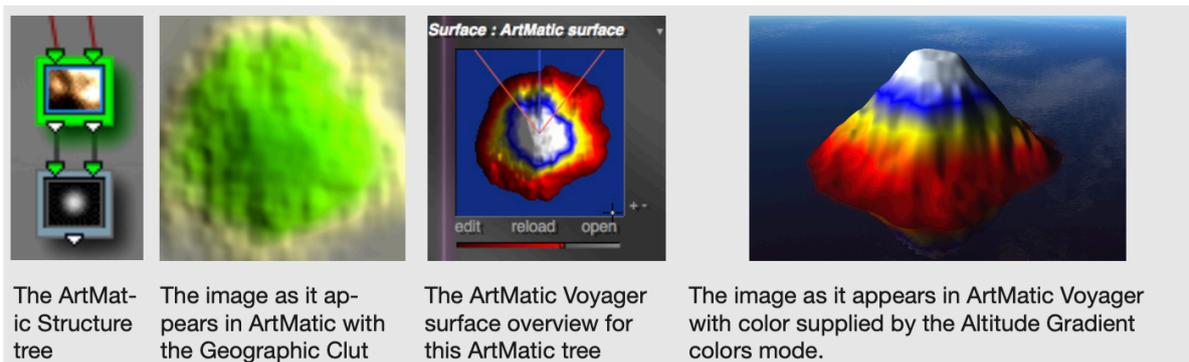
Output:

There are two types of trees that can be used as ArtMatic Voyager surfaces:

- single output trees that provide elevation information only,
- RGB+Alpha (four output) trees that provide Voyager with both the color texture and the elevation map.

SINGLE OUTPUT: ELEVATION-ONLY ARTMATIC SYSTEMS

Elevation-only maps provide only the elevation information to Voyager and no color information. Elevation-only maps have two inputs (global X and Y), which are the Voyager ground coordinates (latitude and longitude), and a single output that Voyager interprets as elevation. The tree shown below is extremely simple. With more complex trees, one can create fantastically intricate planet surfaces. (Explore the provided library to get a sense of the variety possible).



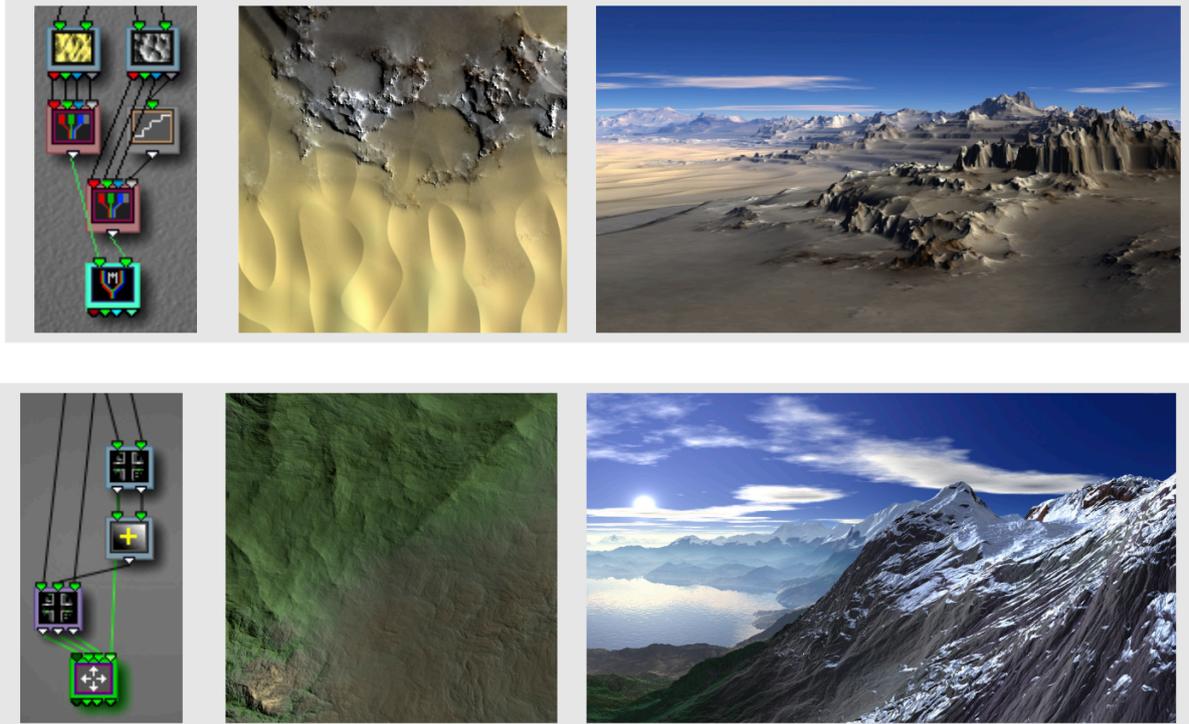
RGB+ALPHA TREES: (COLOR+ELEVATION)

Because it is often useful to design the elevation map and a related color-texture, Voyager makes it possible to use the same Artmatic system to provide both elevation information and color information. When the **Surface Mode** is *ArtMatic Surface*, *ArtMatic Animation* or *Combination*, and the **Colors Mode** is *Default*, Voyager will use the color information embedded in the ArtMatic system, **if** the Artmatic system has four (or more) outputs.

The first three outputs are interpreted as RGB color, and the fourth as the elevation map.

RGB+Alpha trees may be either 2D-input trees or 3D-input trees. If there is a three-input component in the top row of the tree, the tree is treated as a 3D-input tree (where X and Z are the ground coordinates, and Y is the elevation).

ArtMatic trees with integrated elevation and color information (RGB+Alpha Trees)



When an RGB+Alpha tree is used, the **Color** mode must be set to *Default* in Voyager for the embedded color texture to be used.

Extra Output trees:

As with Artmatic color textures, the Artmatic system can provide additional extra outputs that Voyager can map to properties such as wetness or bump-shading.

Surface Features:

ArtMatic Engine provides hundreds of band-limited fractal functions that are especially designed for terrains creation: from the basic *Multi-Perlin Noise* to advanced multi-fractal noises terrains like in *21 Fractal Terrains #*.

Some surfaces functions will have also color texture information imbedded like the with *24 Ridged Noise* tile.

In 'Absolute' mode you will typically use the component's *Frequency Options*: 'Voyager KM' or 'Voyager DF mode' to define the scale of the surface. For entire planets the lowest frequencies will often be 10 to 100 kilometers wide. With Voyager DF mode the amplitude is automatically connected to frequency, to keep a realistic aspect ratio between width and height.

To create a varied planet topology, it is often useful to mix several surfaces with different features. All mathematical and logical operators can be used to merge various terrains. An easy way to create variety is to random-mix several terrains using the scalar *31 Random Mix* or vector *34 Packed Random Mix* component.

The Voyager Library provides many examples of terrain functions that can be browsed directly from the main UI using the *Browse Terrains Library* pop up.

Rendering passes: How Voyager evaluates RGB+Alpha trees

ArtMatic Voyager must evaluate the trees at various stage of the rendering:

- A first pass to find the intersection with the terrain elevation, either from a built-in planet, an ArtMatic terrain, or a combination of both.
- An intermediary pass to compute the slope and normal.
- A final pass (shading phase) to compute the color texture used in the surface shading. The texture can be defined by the same tree that sets the elevation when the output is RGBA, but can be also provided by another tree.

It is important to realize that different information is available during the various passes. During the first pass (terrain & object intersection), the global inputs that provide Voyager elevation, slope and normal information are not yet defined. So, this information should not be used in the part of the tree that defines the elevation.

Because the color texture computation is unnecessary at the terrain intersection phase, it is more efficient to separate it from the terrain topology: Put all tiles used to compute the color texture in a **compiled tree** and set the compiled tree to be *Evaluate only for colors*. Any tile that has the *Evaluate only for colors* option set is ignored during the first pass. This will greatly speedup color terrain rendering. (See ArtMatic *Shading Options* popup menu).

Additional information can be passed to an ArtMatic tree via the other global inputs and several ArtMatic components.

- **slope** information via *global input A1*
- **absolute elevation** information via *global input A2*
- various **VY vectors** like eye position, sun light direction, terrain and object normal are available with various ArtMatic components.

ArtMatic Keyframes & Default View

When an Artmatic system is used as a Voyager element, the parameters stored in the first keyframe are used by ArtMatic Voyager, if there are any keyframes. If an ArtMatic system has no keyframes, the default view will be used; however, it is strongly recommended that you define at least one keyframe. If there are additional keyframes, they will be used to animate the element when the Voyager scene is animated. ArtMatic surfaces and color textures will be animated only if *ArtMatic Animation* is the selected mode.

Designing ArtMatic Color Textures

ArtMatic trees that are used for color texture can be RGB (3-output) or RGBA (4-output) trees. Voyager evaluates the tree at the shading phase and passes (x,y,z) coordinates to the ArtMatic tree along with additional information like slope, normal, light vector etc.

By carefully designing the ArtMatic tree function, one can create very complex color textures that may respond differently to various information. Often, one will base an ArtMatic color map on the system used to define the surface by using the passed elevation in order to maintain coherence.

There are two types of color textures: 2D color textures and 3D solid textures. The type of texture is determined by which global inputs are accessed in the first row of the tree:

- **2 Global Inputs: 2-Dimensional Mapping**

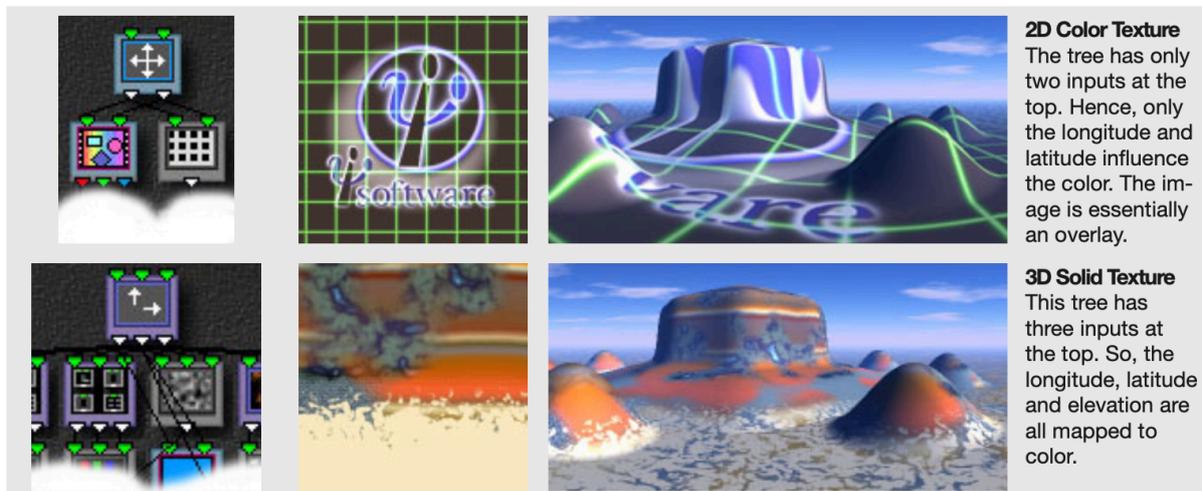
When the ArtMatic tree has **two-inputs at the top**, the longitude and latitude of every point on the map is sent to ArtMatic, and the color output is laid upon the surface. The elevation/surface contours do not influence the color selected (since only longitude and latitude are sent to the

ArtMatic system). Imagine an ArtMatic image large enough to cover the map. This image is essentially overlaid upon the planet surface.

- **3 Global Inputs: 3-Dimensional Mapping**

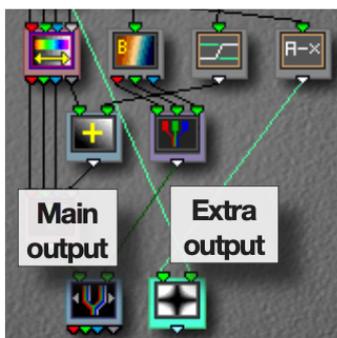
When an ArtMatic tree has **three inputs at the top** of the system, Voyager treats the system as defining a three-dimensional color texture space. Consider that each point on the planet surface has a 3D coordinate: longitude (X), elevation (Y) and latitude (Z). Voyager feeds those coordinates (in that order) into the ArtMatic system and uses the color returned. If that explanation seems abstract, imagine that the ArtMatic system defines a block of intricately colored marble as large as the planet and as high as the highest elevation. Now, imagine chiselling away marble from the block of marble until the contours match the planet surface. When you look at the ArtMatic system in ArtMatic, you are looking at the block of marble from the front.

RGBA Trees: When the texture provides an **Alpha channel**, the alpha controls the blending with the built in Planet default colors. For example, a Rock texture may be set up to return an alpha linked to slope so it will appear only on steep mountain slopes.



Note: In a 2D tree Voyager passes Latitude and Longitude to Global Inputs X and Y, so the view in designer is seen from above. In a 3D tree, X and Y are treated as Longitude and Elevation (Z is latitude), so the view of the texture in Designer is a front view, as if you were standing in front of a sheer cliff colored with the texture.

Extra outputs (X-outs)



When an ArtMatic file has one or more extra outputs, the output can be mapped to various shading properties such as wetness, self-illumination or reflectivity. Using extra outputs to modulate texture shading can greatly enhance realism and visual complexity.

The 'Extra' tile is usually a single-output component at the right of the main output of the tree. There can be more than one extra output, to map multiple extra shading properties.

Use Voyager's *Quick Edit Artmatic params & shaders* dialog to set what the extra output is mapped to.

The extra output(s) can be mapped to:

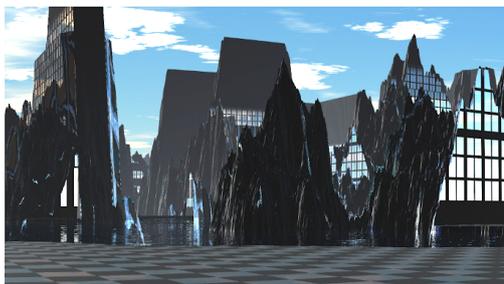
- **nothing:** Use it to turn off a particular shading option.
- **Self illumination:** Unlike "Ambient" which sets the amount of diffuse reflection coming from the environment, "Self illumination" adds its own light to the scene giving the impression of a luminous object. The self illumination light color is the X-out color or white if X-out is scalar.
- **Wetness level:** Controls the specular amount of light coming from the environment. This light can be filtered by the X-out color if any.
- **Ambient:** Controls the amount of diffuse light coming from the environment. This option is 'scalar only' as the ambient color comes from the environment or an image map.
- **Reflection level:** Mirror reflection of light from the environment. The reflected light can be filtered by the X-out color.
Note that for terrain true mirror reflections, 'enable true reflections for terrain' has to be enabled in the 'Terrain Shader Settings dialog'.
- **Bump Map:** This require a 3D X output providing the derivative vector and will map the vector to normal perturbations.
- **Brightness Gain:** The X-out scales the diffuse & ambient light. If the X-out is RGBA then A controls how much illumination is scaled by the X-out color. this mode can be used to simulate shadows or light variations, as well as coloring the main light in variable ways.



Above: Extra output mapped to wetness to provide glistening. **Below:** Extra output mapped to Reflection Level to create the window reflections.



Above: Extra output mapped to Ambient to create the lava's glow. **Below:** The extra output is mapped to Normal (Bump Map) to created the embossed look.



X-outs naming conventions for auto-mapping:

The following letters put at the end of an ArtMatic texture file-name will cause Voyager to set the proper shading option default when opening/importing a new ArtMatic file. The letters can be

combined in any order up to 3 letters ("ri", "wir", "wbi" etc) when multiple X-outs are used, but a space needs to be present before the first x-out letter, to not get confused with letters in the name itself.

- 'i': sets corresponding output to "Self illumination color & level",
- 'r': sets corresponding output to "Reflection color & level",
- 'w': sets corresponding output to "Wetness level / Specular color";
- 'b': sets corresponding output to "Bump Map ",

So for example, "myfilename ri" will have "Reflection color & level" set for X-out 1 and "Self illumination" set for X-out 2. "myfilename r" "myfilename wi" "myfilename lri" "myfilename rib" are all valid auto-mapping hint names .

ArtMatic Scaling Mode

When an ArtMatic system is used as a terrain or texture in Voyager its **scale** can be set in various ways in the 'Terrain Shader settings' dialog.

- **Absolute:** the scaling is independent of keyframes/AM view zoom level. In the default zoom level (=1) the ArtMatic canvas represents an area roughly 122 meters by 122 meters in Voyager. Absolute scaling is generally recommended.

Absolute mode ignores the ArtMatic zoom level and canvas origin, and 3D color textures maintain the correct aspect ratio. Interchangeability of ArtMatic terrains, and color textures is maximized when this mode is used.

- **Absolute + Keyframe offset:** This mode uses the same distance mapping as Absolute, but respects the ArtMatic canvas view (or "camera") origin. **Note: This mode is deprecated and may disappear in the future. Don't use it for new terrains design.** The Main Voyager CTX UI offers position controls for terrains and that should be the way to offset terrain origin.
- **Relative to Keyframe Scale:** the camera zoom level used by the first keyframe determines the scaling of ground co-ordinates from ArtMatic to Voyager. The first keyframe represents an area roughly 1 km by 1 km. The vertical scaling is unaffected by the scaling mode. The relative mode is provided primarily for compatibility with files created in older versions.

To determine the area represented by a keyframe at an arbitrary zoom level, the formula is..:
Distance = Camera Zoom Level * (1200/√2).

A close approximation is: Distance (in kilometers) = Zoom Level * 0.122 In relative mode the aspect ratio of the color texture varies with the ArtMatic system's camera zoom level and is subject to orientation. **Note: This mode is deprecated and may disappear in the future. Don't use it for new terrains design.**

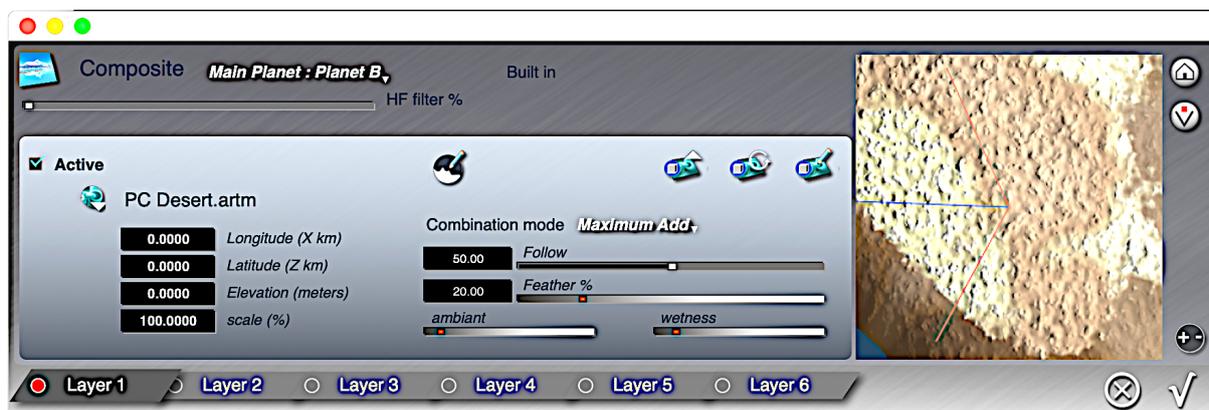
The Combination mode and Combination dialog

Combination mode makes the creation of more complex planet terrains easier by combining a built-in planet or an ArtMatic surface with up to 6 additional ArtMatic surfaces.

The additional surfaces can supply a wide variety of features: from beach pebbles, boulder rocks, rivers, volcanos, to architectural structures, custom oceans and more. A layer can also be used as a filter to modulate elevation and add texture details. Since Voyager CTX, each layer can have its own [X-outs](#) settings, as well as particular wetness and ambient level.

Be sure to take a look at the example files to see the many different features that combination mode can give to Voyager worlds. Examples can be found in *Voyager examples/Terrains & Landscapes/VY5 Combination/* and in *Voyager Scenes/Combination scenes/*

Combination Mode popup dialog: With 'Surface mode: Combination' selected, click *edit* to open this dialog.



The dialog has all the controls needed to import, edit and manage terrains layers.

Click on a **layer tab** to access each layer's settings. Active layers are indicated with a red dot. To create a new layer, select an unused slot and open an Artmatic file that will define the particular layer features.

The Browse Library button gives you a direct access to a number of useful presets to enhance your scene. Position the layer using the Map or numeric entries, choose the combination mode and eventually change the layer scale. Results should be seen interactively in 3D in the main preview area as well as in a 2D top view in the Map.

Main Planet (text popup)

Choose the surface that will be the main surface. The additional layers are combined with the main planet. The main planet can be either a built-in planet or an ArtMatic surface/animation.

Main planet edit/open/reload



When the main planet is an ArtMatic surface/animation, these buttons let you edit the surface in ArtMatic Designer (if it has been installed), open a new ArtMatic file to use, or reload the ArtMatic file. Use the reload button, if you edit the ArtMatic file while the Voyager scene is open.

HF Filter % slider (High Frequency Filter):

This slider reduces the amount of detail in the built-in planet. In many cases, this makes the ArtMatic contribution more noticeable where fine surface detail is concerned. This filter only affects the base planet and not the ArtMatic contribution (as opposed to the High Frequency limit set in the Preferences dialog which affects everything).

Active checkbox: Use this checkbox to toggle the layer's active status.

Browse library: Browse library pop up gives you a direct access to ArtMatic systems designed for combination mode and stored in the *Voyager Library/Combination/* folder. It contains *Absolute Rocks*, *Surface details*, *Surface textures*, *Alternate seas*, and *Filters* folders.



Layer edit/open/reload:

These buttons let you open a new ArtMatic file to use, edit the layer system in ArtMatic Designer (if it has been installed), or reload the ArtMatic file.

<p>Longitude (X): Latitude (Z): Elevation (Y):</p>

These fields provide offsets to control the relative position and elevation of the ArtMatic terrain layer and the main planet. The offsets are added to the ArtMatic file's normal home position. You may also use the Map view to set layer origin.

Global Scale % This is a scaling factor that is applied to the ArtMatic file, before it is combined with the main planet. It represents the relative scaling of the Voyager world to the ArtMatic world. Hence, values over 100% reduce the size of the ArtMatic features, while values less than 100% increase the size.

Wetness and Ambient sliders: Each layer of combination mode may have its own levels of wetness and ambient and will use layer x-outs independently of main terrain texture and shading option. To set up how ArtMatic file extra outputs are assigned in the shading use the **Quick Edit ArtMatic params & shaders** inspector. Select the ArtMatic file that corresponds to the given layer and set up the options as usual.

Terrain shader Settings...



You can call this dialog to adjust terrain rendering & shading options.

Surface Overview Map.

This map provides a top view of the surface with camera orientation lines that display the camera's orientation and field of view (as in the main window's surface overview).

Click on the surface overview to center the current layer's ArtMatic system at the clicked location - the click sets the Longitude and Latitude offsets to the clicked location. This feature is useful if you want to position an ArtMatic structure at the current location rather than at the center of the Voyager world. Option-Click to center Layer 2.

Centre to world origin:



Reset the offsets so that the Voyager world and the ArtMatic world are both centered at 0,0.

Centre to Current View:

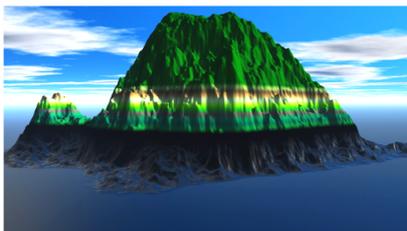


Set the longitude and latitude offsets to the center of the current view.

Combination slider(s):

These sliders' meaning depends on the active combination mode algorithm (see below) and usually control the amount of influence that the ArtMatic layer has on the main planet.

Combination Mode (text popup algorithms):



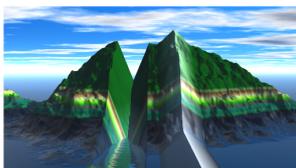
Main Planet: ArtMatic Surface: *Mound*



Layer 1: ArtMatic Surface: *Lines*

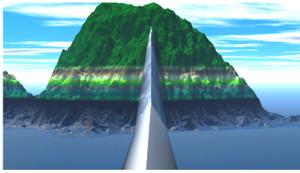
These two surfaces are used for many of the combination mode illustrations that follow. The Lines system features two lines: one (on the right) that has surface elevations above 0 and one (on the left) whose surface elevations are below zero.

- **Blend:**



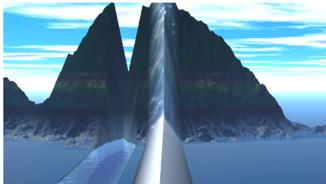
Blend the main planet with the ArtMatic surface by performing a weighted averaging of the surface elevations. The slider controls the weighting of the blend. When the slider is 0, the Layer has no influence. When the slider is set to maximum, the Main Planet has no influence. Blend is the preferred mode to apply filters.

- **Maximum Add:**



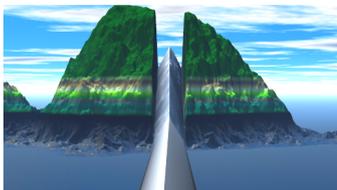
Compare the main planet with the ArtMatic surface and add the ArtMatic surface where it is higher than the main planet's surface. The slider bends the ArtMatic surface to follow the main planet's surface before the comparison is made. The additional "Feather" slider controls the smooth blending of the edges.

- **Add:**



Adds elevation of layer to the main planet. The slider adjusts the relative weighting of the main planet's surface and the ArtMatic surface. Unlike Add Alpha and Maximum Add, Add combines both surfaces even where the ArtMatic surface has values less than 0. Where the ArtMatic surface is less than 0, it will carve out indentations from the main planet's surface.

- **Add Alpha**



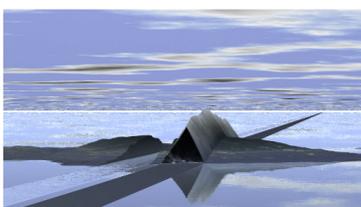
The elevation of the ArtMatic system (when > 0) controls the blending of the main planet with ArtMatic surface. The slider controls the degree to which the ArtMatic surface is deformed to follow the main planet surface before the blending is done. When the slider is set to the maximum value, the ArtMatic surface will follow the contour of the planet surface. When the slider is set to 0, the ArtMatic surface is left un- changed. Set the slider to the maximum to add details such as rocks to the main planet's surface. Where ArtMatic output values are greater than 1 the surface is taken entirely from the ArtMatic surface. Values below result in a weighted blending of the ArtMatic surface and the main planet.

- **Alternate sea:**



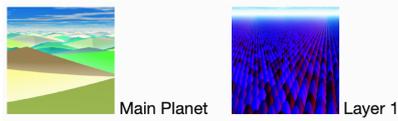
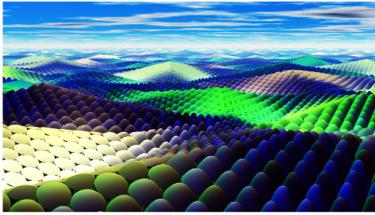
Replace the built-in sea with an ArtMatic sea. The ArtMatic sea should be an RGB+Alpha tree. The tree's RGB output defines the color texture, and the alpha channel defines the sea height (making waves possible). See the example file *Voyager Scenes/Combination scenes/Bad Sea.vy* The Bad Sea example makes use of an extra output mapped to Ambient to give the sea foam an added luminescence. Voyager's native sea level may need to be lowered.

- **Minimum:**



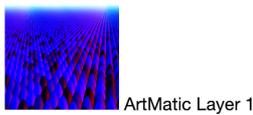
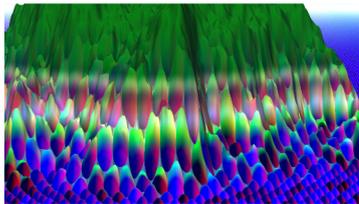
Minimum is taking the lower elevation of the Main planet and the layer's terrain. The slider performs a weighted averaging of the unaltered Main Planet with the calculated minimum. The additional "Feather" slider allows a smooth blending of the edges when feather is above zero.

- **Random Blend:**



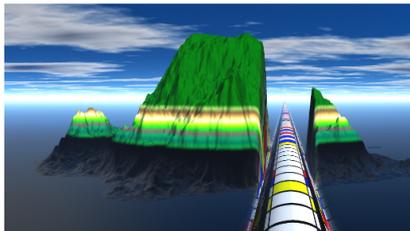
Randomly blend the main planet and the ArtMatic layer with the slider controlling the frequency of a low frequency blending noise. This mode is handy to randomly replace planet zones with features of the layer's terrain

- **Blend on lows:**



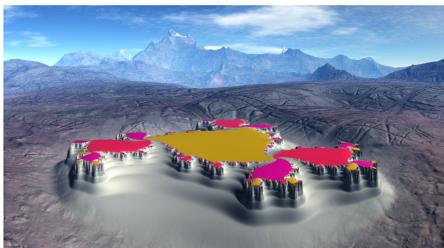
Blend in the layer where the main planet's elevation is low. The slider controls the maximum elevation at which blending occurs.

- **Exp neg Alpha:**



This mode blends the ArtMatic layer with the planet in such a way that the ArtMatic surface cuts through the main planet. Geographic Clut view RGB View Where the ArtMatic elevation (alpha) is greater than 0, the surface is provided completely by the ArtMatic surface. The Feather slider influences how alpha values less than 0 influence the mix of the ArtMatic system and the main surface. When feather is at its maximum, negative values have very limited influence on the main planet. The ArtMatic surface's influence drops off exponentially as the value decreases below zero. When feather is at the minimum value, the influence of negative values drops off very slowly so that the ArtMatic surface has quite a bit of influence even where the alpha value is very negative. Generally, the feather slider will be set close to the maximum value.

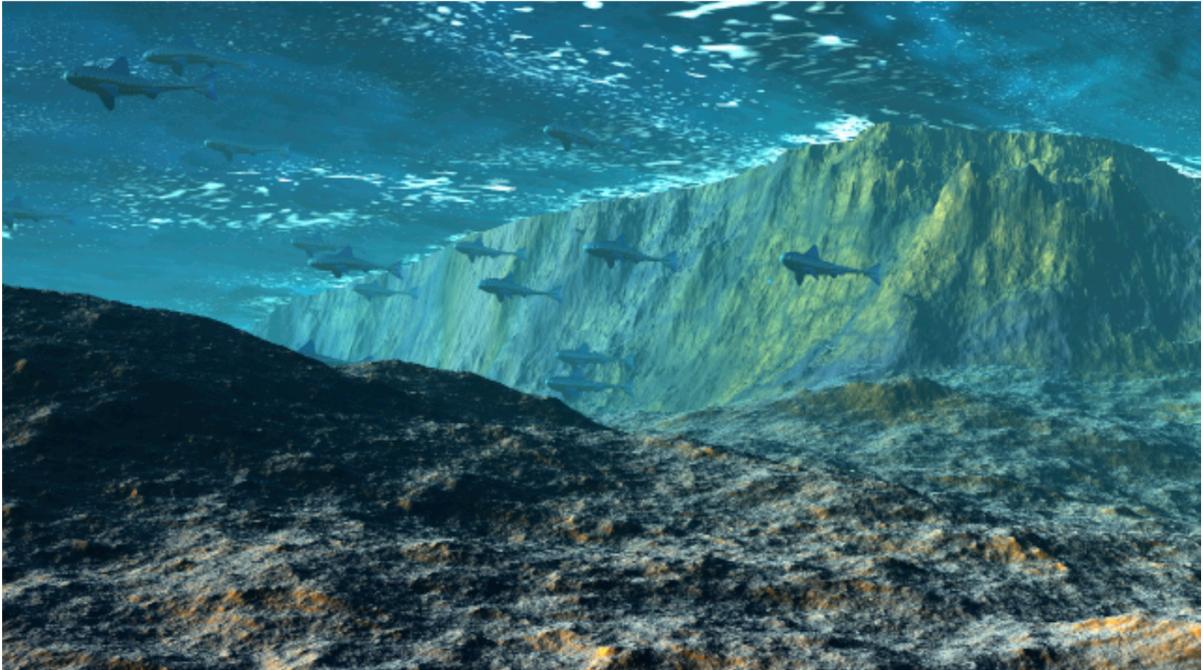
- **Disk Blend:** This mode (introduced in Voyager CTX) mixes the auxiliary ArtMatic terrain inside a blending disk, thus limiting the placement of it at a specific place. The main slider controls how much planet terrain is added to the auxiliary terrain. The additional slider sets the radius of the Disk in percent from 100 meters to 10 km. Elevation & textures are smoothly blended when moving away from the disk. To position the Disk, click (and drag eventually) on the Overview Map. The center of the disk will be the ArtMatic system origin.



Example : Voyager Examples/Terrains & Landscapes/VY5 Combination/JuliaSet inSitu.vy

Underwater Shader (New Sky Mode in Vy3)

ArtMatic clouds/lights mode has new *Underwater Shader* mode for creating underwater Scenes, as well as interesting volumetric light effects that are faster to compute than when using volumetric light mode.

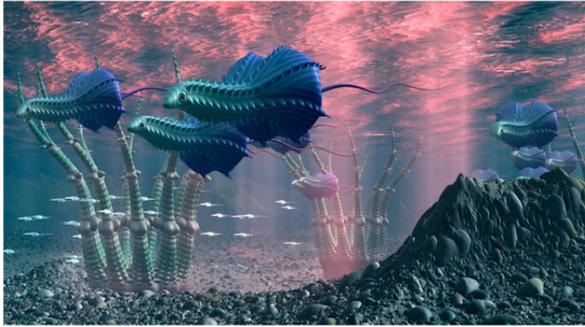


Underwater scene using the Underwater clouds/light mode



Underwater mode can also be used for above-water special lighting effects.

ArtMatic Voyager comes with several ArtMatic systems that implement underwater shaders that can be used by anyone regardless of their expertise. **Creating** new underwater shaders from scratch requires a fairly solid understanding of how ArtMatic and ArtMatic Voyager work.



Other Applications

Underwater mode is not restricted to underwater effects. When the camera is above the water level, you can create a host of interesting effects which include:

- Creating your own bodies of water with different coloring from the main ArtMatic water. You can create great effects (or realism) by modulating the color and waves with ‘depth’ (*global input A2*). The water surface of “underwater mode” water is opaque.
- Above-water atmospheric lighting effects. Underwater mode lights are faster to compute than the volumetric lights mode. So, this mode can be useful for creating interesting volumetric lights without using too much computation.
- Special volumetric effects like smoking water or sea spray/mist.
- Many cool effects that have not even occurred to us!

Haze. Underwater mode sets the haze stronger underwater, and mixes the *haze color* with the *underwater shift color*. The haze is used even in No Planet mode, which allows you to have a bottomless ocean. The higher the water level the more the haze color is shifted towards dark blue.

The current depth (*cloud level minus terrain altitude*) below the water surface is sent via *global input A2*. You can use this feature to design a water model where the color/waves/foam change with proximity to the shore.

Water Surface. Even though *Underwater Shader* mode was created for implementing underwater scenes, it is also great for providing water surfaces seen from above the water, and reacts well to a variety of lighting conditions.



an underwater mode sea seen from above



under different lighting conditions



the same scene viewed from underwater and a bit more offshore

Control Sliders. The cloud sliders have the following meanings in underwater mode.

- *Cloud Density* - controls the intensity of the volumetric light effects. When it is set to 0, the volumetric light effects are turned off, but the water surface is still shaded.
- *Water Surface Height* - This controls the elevation at which the water surface appears.
- *Cloud Size* - This slider scales the Artmatic system just as it does in the other ArtMatic cloud modes.

Special Effects - Underwater mode is useful for interesting above-water lightings effects.



Example: Underwater Effects.vy



Ex.: Underwater Experiment.vy. Camera above.



Example: Underwater Experiment.vy as seen from below the water. True reflections is turned on in the Terrain Shader settings which allows complex reflections at the water surface due to mirroring of the terrain.

Tech Talk

The ArtMatic system used to provide the underwater shading must be a 3D system (making use of the global X, Y and Z inputs). Elevation equals 0 represents the water surface. Where the elevation is less than 0, ArtMatic Voyager will calculate underwater lighting effects. Where the elevation is greater than 0, Voyager renders above-surface volumetric lights. Much looser sampling is used for calculating the lights than for the true volumetric light mode in order to keep rendering times reasonable.

The water surface is opaque, and the alpha channel defines the wave “bumps”. Extra ArtMatic outputs (which are hard-wired as noted below) can be used for special effects such as self-illumination & wetness modulation or reflections. This optimization makes underwayter mode useful for above-water lighting effects, since it is faster to render than the standard volumetric light mode.

Extra outputs. Underwater shaders make use of extra ArtMatic outputs to control aspects of the underwater experience. The extra output mapping is hardcoded, in order to avoid conflicts with the extra output mapping done through the Terrain Shader Settings dialog.

- Output 1 (RGBA): water surface + volumetric light effects + caustics
- Output 2 (scalar): reflection amount. This output is only active when *Enable true reflections for terrain* is turned on in the Terrain Shader Settings.
- Output 3 (RGB): additional light. This effects the water surface only. A common use is to enhance foam brightness.



Example: "Underwater Reflections C". Notice that the surface reflects the medusae.

Building 3D Objects: DFRM Guide

Introduction

ArtMatic Voyager uses a unique approach to 3D objects modelling & rendering called Distance Field Ray Marching, DFRM for short. This document covers the details you need to know to create or modify 3D objects represented by Distance Estimate Fields (DF for short) and gives many practical guidelines. The technical information may help you to understand the reasoning behind the guidelines and develop your own techniques.

DFRM Concepts

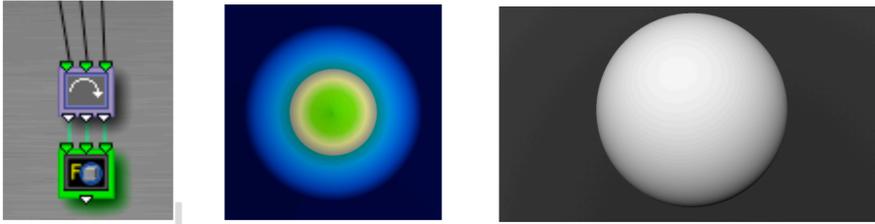
ArtMatic Voyager uses a technique called ray marching to render images (http://en.wikipedia.org/wiki/Volume_ray_casting). Ray marching essentially calculates the intersection of the possible light rays between the observer and the scene by sampling along the ray. It is a slow process as the object or terrain must be sampled many times to know where the ray intersects the object. Ray Marching is needed when the mathematics that describe the object are too complex to find intersections analytically; typically, when the object is an entire procedural planet like in Voyager's case.

By using Distance Estimate fields, Voyager can find the intersection of the ray with the surface of a 3D object much more quickly than using the brute force technique of ray marching. This is because the DE field itself gives some information about the distance to the surface, which allows the sampling to be much more efficient by converging quickly to the surface.

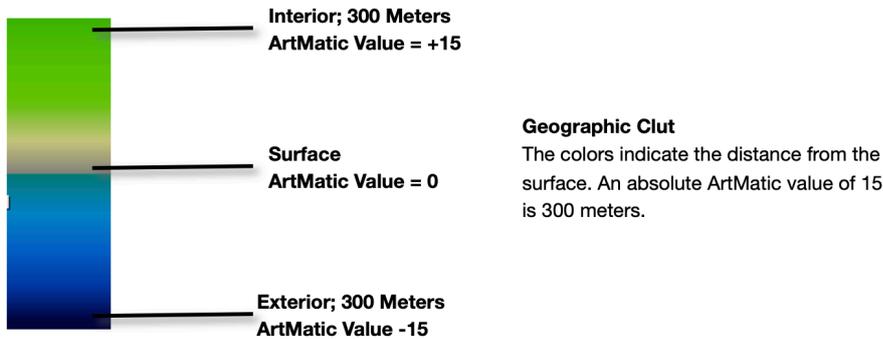
A distance field is just a scalar field where the value of the field gives a good (or exact) approximation of the distance to the surface. The distance field does not have to be mathematically exact to allow proper convergence, but the more exact the distance estimate is, the faster the convergence will be. If the distance estimate deviates too much from the true distance the ray will miss the object (overshoot) if it overestimates that distance. Underestimation won't impair the ability to converge to the correct solution, but the convergence will be slower.

A distance field function takes a space or plane coordinates and calculates an estimate of the distance from that point to the object surface. The object surface is those places where the distance is 0, that is the 'zero crossing' of the field. A value greater than 0 indicates a point inside the object with the field value indicating the distance from the surface. A value less than zero indicates a point outside of the object.

ArtMatic's Geographic Clut color shader is useful for visualizing distance fields as its colors indicate distances. The example below shows a simple tree to create a sphere (using the *3D DF Shapes A#* component), the image as it appears in ArtMatic with the geographic clut color shader, and the sphere as it appears in ArtMatic Voyager. The color at the center of the geographic clut view indicates that the radius is approximately 300 meters.



Below is a key to the colors in the range of about plus/minus 300 meters from the surface:



A distance field can be 2D or even 1D. A 1D distance field is simply x or y or z provided they are unscaled. Thus, you can directly use $-y$, for example to create a DF infinite flat ground where $y=0$ defines a flat ground plane.

The simplest 3D distance field is a sphere. What is remarkable (and unique) is that the sphere equation is its own distance field equation. The field is described by this equation: $R - \sqrt{x^2 + y^2 + z^2}$, (or $R - \text{length}(x,y,z)$; 'length' being the euclidian distance) which comes from the sphere equation: $x^2 + y^2 + z^2 = R$ with R being the radius of the sphere. The minus sign is needed to adjust the field values so that the field is negative outside the sphere and positive inside. With the sphere, the convergence can be done in a single step as the Distance Field $R - \text{length}(x,y,z)$ gives you the exact distance to the solution.

The field exists everywhere in space which makes DF objects non-local unlike with classical polygonal description. One can see a distance field as a special kind of 'scalar field'. Scalar fields are non-directional (unlike vector fields) and non-local. This non-locality (field exist everywhere in space) makes the object information expand way beyond its boundaries. This property is quite interesting, as a simple offset of the field value will expand or shrink the object.

In the documentation we often refer to **DF** for Distance Field, or **DE** Field for Distance Estimate Field. In fact DE or DF are the same but since most of the time the DF is not a true mathematical distance to a complex object (which is often quite difficult to compute) the "Distance Estimate" is more accurate. "DF field" is redundant so when field is written "DE field" is more elegant.

DE fields can be manipulated in many ways impossible (or very difficult) with polygonal descriptions:

- DE fields can be blended or morphed together
- DE fields can be distorted by space distortion functions
- DE fields can be combined using boolean operators
- DE fields can be used as the input coordinates for another DE Field calculation.

DFRM is useful not just because of its computational efficiency, but because very simple operations can be used to create complex and interesting shapes. Animating these transformations can create fascinating object morphs that would be very difficult to create with more traditional 3D tools.

DE fields provide a unified representation for very different types of objects; a tree, a fractal, a building, a sphere. This representation is non-local and independent of a particular topology. That makes morphing and combination of very different types of objects quite easy. Thus, efficient and simpler techniques for 3D modelling are possible with DF objects, and ArtMatic Engine provides hundreds of functions designed for DF modelling.

ArtMatic 3D DF objects:

3D DF objects are created using ArtMatic Designer. Creating and modifying them requires a fairly good understanding of ArtMatic structure trees. Many pre-existing DF primitives are available in ArtMatic Engine to provide you with basic DF building blocks that you can combine using boolean functions. For the most part, you will use built-in ArtMatic components that generate distance fields and combine them (using the guidelines provided later) to create complex objects. Although advanced users can create their own distance fields, it is unlikely that you will ever need to create a distance field from scratch.

The 3->1 *3D DF XYZ shapes #*, *3D DF shapes A #*, *3D DF shapes B #* and 3->4 *3D leaves #* components are the most frequently encountered distance field generators. Many other components are provided for providing useful 3D space manipulations that convert primitive shapes into more complex shapes. You can also create distance fields by combining 2D shapes with other components as discussed elsewhere in the documentation.

ArtMatic trees need certain properties to work as DF objects: Any 2D or 3D scalar function can be interpreted as a distance field, as long as the field value provides a good distance approximation to the function zero crossing (the surface). 3D object trees need to be 3D, which means they use the X,Y and Z global inputs. The field has to be negative outside the object surface and positive inside. Any functions that are clamped to zero (positive only) cannot be used as a DF generating function. A DF object is not necessarily bounded and small. You can have one DF object describing an entire city or a forest. Using Compiled Trees, there is virtually no limits in the number of functions and complexity of geometry you can have in a single DF object instance.

Things You Should Know About Working With Objects:

- **Scaling & Size:**

The scaling of DF object is always absolute, and the overall size can be set within Voyager in the *Object Inspector*. But sometimes it is needed to scale DF elements within the ArtMatic tree when merging various shapes or building fractals. In general most DF building blocks will have a Radius or Scale parameter that sets the element size directly.

Changing object size is generally done by adding an *offset* to the distance field value, rather than by scaling space. If you must scale the space, it is necessary to compensate by inverse scaling the field value to maintain a proper DF estimate. Take the case of a sphere: If x,y and z are scaled by 4, the distance estimate will become four times larger than it should be. Scaling down by 1/4 corrects the error. This is the same as just lowering the radius by subtracting an offset to the field itself.

A couple of special components (**S-space scaling**) keep track of scaling automatically, so that the field can be adjusted at the end of the tree with the proper inverse value. 44 space transforms will give you many operators keeping track of the S-scale to build fascinating DF based volumetric fractals. Rotations and Mirroring functions can be used safely, as they don't change the space scaling and keep the distance field always accurate.

- **Positions**

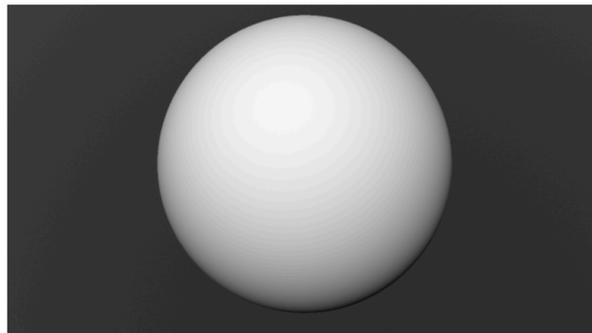
Voyager provides many sliders and ways to position the entire DF object in the scene.

When an ArtMatic tree mixes several DF objects it is often needed to position the objects relatively within the tree. A simple space translation doesn't modify the DE field accuracy and is safe to be used. *1D Offset* component, *3D Offset* component, and any vector offset functions can all be used to move various parts. A translation is a simple addition of a constant value to a space coordinate. If the relative displacement is needed only in one dimension it is efficient to use the *13 Add* vector function.

To Animate the object position you will use the artmatic keyframes with varying offset tile parameters, or more complex motion functions connected to global time input w. It is often easier for most regular motion to use time instead of keyframes to animate object. For example: time can feed the w of a *43 w rotate* component to rotate an object automatically.

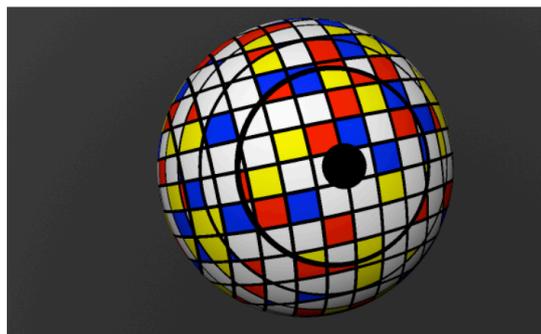
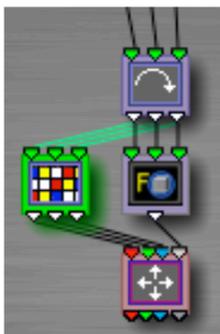
- **Object Color**

ArtMatic Voyager may ignore the color shader used in ArtMatic. The colors displayed in ArtMatic Voyager depend on what type of output the ArtMatic structure provides.



For a tree with scalar output, the output is the distance field value. Voyager will treat it as a white object.

The color can be modified in the Voyager object properties inspector.



For a tree with RGBA output, the alpha channel must be the field value. Voyager will use the RGB output to color the object.

To associate a color with a DE Field you will typically use the RGBA stream format with A holding the Distance estimation data. A constant color tile can provide the RGB data if the object has a single color and no texture.

Advanced users can build a color texture function to feed the RGB associated with the object. As with terrain textures, the object texture is computed after the intersection phase, and you can optimise rendering speed by separating the texture computation from the object Distance field computation. In that case, put all tiles used to compute the color texture in a compiled tree and set the compiled tree to be *Evaluate Only For Colors*: Any tile that has the *Evaluate*

Only For Colors option set won't be computed during the Intersection phase. (See “*Rendering passes; Artmatic Surfaces*”.

The Color texture function will often be a 33 tile that gets its input from incoming space and outputs RGB data.

In more recent versions you will find a lot of DF RGBA functions that output both a DE Field and a color. These can be used to carve a DE Field, and at the same time provide a color to the object. When all RGBA channels are used by the same component, it is no longer possible to use the *Evaluate Only For Colors* option, as it affects the geometry as well.

Specific boolean operators have been added in Engine 8.5 to simplify DF object texturing.

- **Previewing in ArtMatic**

Since ArtMatic Designer has only a 2D view, you will see a *Slice* of the field. 33 space transforms like **3D Spaces #** : *Artmatic top view* are handy to see a top view map of the field, even if the Voyager rendering shows the object standing. You may also use a **3D rotation** tile and set up a bunch of views using ArtMatic keyframes to 'look' at the object slice from different directions while modelling.

When creating DFRM objects in ArtMatic, it is often useful to switch between shaders. The **Geographic Clut** in particular is great to visualize the distance field to make sure that it is correct. The Geographic Clut makes it easy to see if there are anomalies caused by too much scaling or distortion. For all objects, there should be an orderly and reasonable transition as one moves away from the surface or into the object's interior. The negative region outside the object are shaded in blue while interior is shaded with a geographic color ramp according to distance estimate.

To see a preview of a texture function, you can switch the ArtMatic shader to **RGB Density**, to get a slice of the color texture. Regions outside the object are treated as transparent.

But the most efficient way to model in ArtMatic is to have Voyager run in the background and allow ArtMatic to send data to Voyager using the link button. In that case you will see a preview window of the 3D Voyager rendering while working in ArtMatic Designer. Then you can fine tune many parameters while seeing the 3D result interactively.

Note: When using Voyager and Designer concurrently, make sure that you have ArtMatic Designer launched before you click on an "Edit in ArtMatic button". That will ensure the correct version is used.

Design guidelines

- **Don't scale the space :**
If you must, do it uniquely with the 34 [S-Space Scale](#) function, and don't forget to divide the DE Field by the S value at the end. In general, growing DF objects is more efficiently done by adding/subtracting to the field itself. A simple *offset* connected to a DF output will thicken or thin the field.
- **Don't distort space too much**, or compensate by scaling down the DE Field. When using arbitrary noise functions for displacement, make sure not to use amplitudes that are too large. If the amplitude is too large, the displacement may be so large that the DFRM will not converge or will miss certain areas (resulting in artifacts). The solution is to reduce the *amplitude* parameter, or to add a filter to the output to reduce the values.
- **Don't overblend with non DF functions.** Many interesting functions for terrain design are available in ArtMatic engine. These functions, even without being true DF estimates, can still be used to add textures or deformations to DE fields by blending them with the DF function. As with scaling, do it sparingly, and if the convergence is affected reduce the amplitude of the final DF value.
- **Don't use Additions:** Use 21 *Interpolate (rotate interpolation or linear interpolation)* instead of $+ (Ax+By+C)$. Interpolations will make the DF estimate slightly undershoot while addition will make it overshoot ($1+1=2$: so adding 2 DF will scale the estimate by 2). Undershooting is fine: with a distance estimation that is below the real distance (undershoot) you will always converge to the solution, just a bit more slowly. Overshooting is bad: it will produce lots of rendering artifact and in the most catastrophic situation the object will simply disappear.
- **Use logical operators** to combine DE fields. Logical operators like *MIN* (intersection) or *MAX* (union) don't scale or damage the field accuracy, so they are perfect for mixing various DF objects. Many Logical operators components are provided for scalar and RGBA DE fields by ArtMatic Engine:

- 21 S:P Logic & Profiles
- 21 Logic Tools #
- 24 Packed Logic #
- 34 Packed Logic #

Examples of Logical operators are located in *Voyager Examples/Components/Logic tools*

- **Use Compiled Trees (CT) and divide and conquer method :**
For complex modelling, try to break the problem into smaller parts. You can model and test each part separately and make them into a CT, usually of the form 3 in 1 out; with a packed RGBA output so that the object contains its own texture as well as its geometry. When all is set you can copy and paste the CT into the bigger project.

CTs can be nested, so you can always replace one branch with a collection of functions packed into a CT. Think of a CT as a logical group of object.

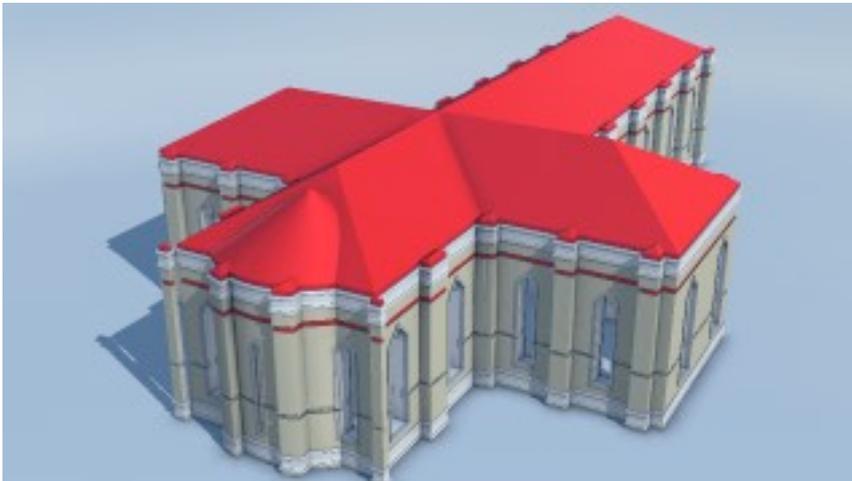
A good practice is to place a 33 translate on top of each CT, so that you can move each part relatively to others. Remember that everything connected to a particular space transform will share its coordinate system and be moved with it. Transforms that are included below will be

relative to the transform above. So, you can design complex articulated motions, keeping track of the flow of transforms and their dependencies.

Imagine, for example, a rotating articulated robot arm placed on a wagon on a rail. The logic of what is needed, and what should be dependant of what, determines the structure and flow of information; thus the structure of the Tree. The wagon and arm move together: its 2 branches connected to the same space. The arm has various parts articulated: put them in a CT with a rotation on top as the entire thing rotates. Each arm part can be other CTs connected in cascade, so that the hand moves correctly when the first part rotates.

It is always good to take some time to clearly understand what is needed and break the problem into smaller chunks. Start with simple 3D objects to get the structure and behavior right. Then you can refine them separately in other trees and replace the draft version with the final model using copy/paste of the CT.

Designing a generic multi CT model is a huge time saver, as the same structure can easily be changed to make many models; as the examples provided in *Voyager Examples: DF Modeling/Architecture Classics: "Path Profiles Church", "Path Profiles TwinTowers" and "Path Profiles Rect4D.vy"* demonstrate.



Example of generic CT designed model tree /Voyager Examples/DF Modeling/Architecture Classics/Path Profiles Church

DF Modelling techniques

Examples of basic modelling technics are located in Voyager Examples/DF Modeling/Basic technics

Many times it is simpler to build a 3D object using 2D DF profile like the ones provided by **21 Profile Shapes #** or **21 DF Curves**.

A 2D DF profile is just a DE Field defined only in 2 dimensions: it will be infinite in the undefined dimension. For example, a 2D DF disc connected to (x,z) will render as an infinite column in Voyager because y is not specified.

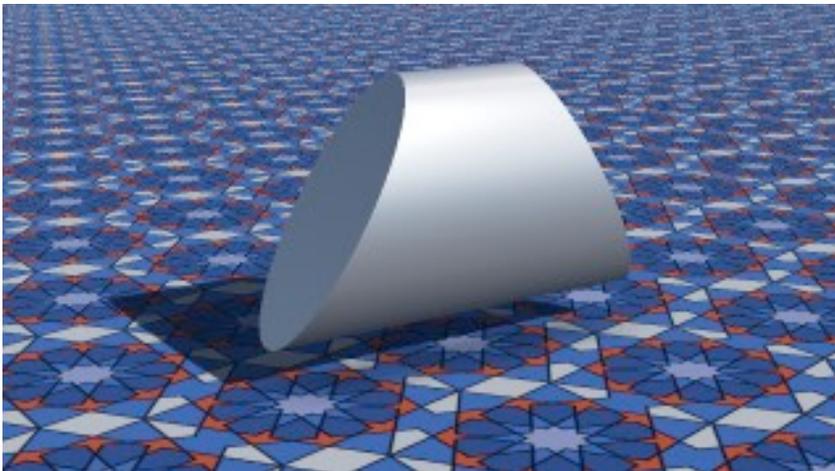
The most useful techniques to work with 2D profiles are:

- **Intersection:**

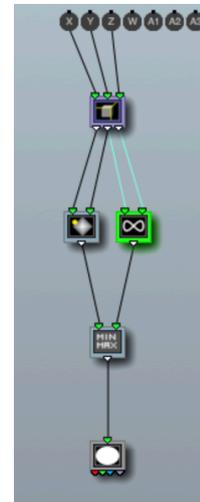
You can intersect two 2D DE fields defined in different planes to create a 3D object. Think of a 2D DE field as a "profile path" where the zero crossing of the field defines the path shape. Used directly in 3D, these profiles will be infinite in the other axis, typically z if the 2D DF component is connected to (x,y) or y if the 2D DF is connected to (x,z).

By intersecting an (x,y) profile with an (x,z) profile you will ensure the object is bounded in all dimensions. The result will be a 3D object that looks like profile A in one direction and profile B in the perpendicular direction.

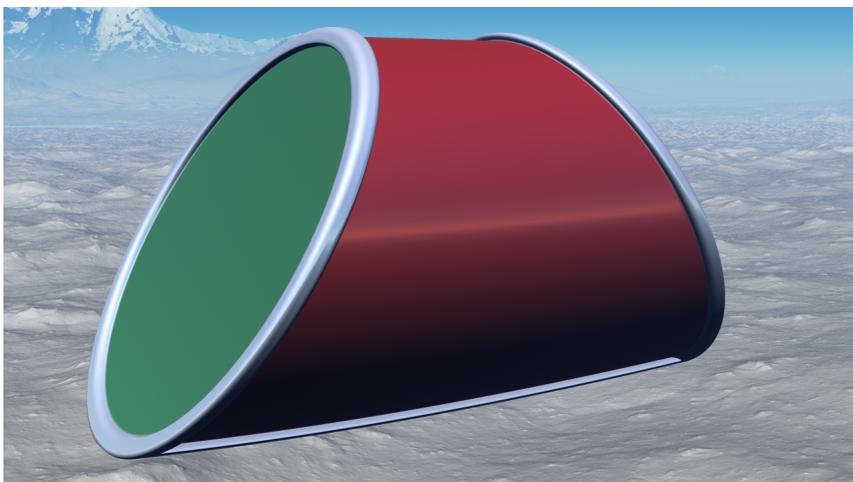
The intersection is typically performed using Boolean (logical) operators like **21 Logic tools #** or **S:P Logic & Profiles #**, but for a basic intersection a simple Minimum function will work (**Min** <-> **Max** set to minimum).



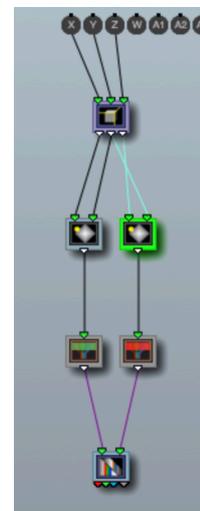
A 2D triangle in (x,y) intersects with a 2D B-Ellipse in (y,z)



The intersection can itself can add details to the geometry using various flavors of the Boolean operator. "Edged intersect" for example will add edges at the intersection.



2D green triangle in (x,y) Edged-intersects with a 2D red disk in (y,z)

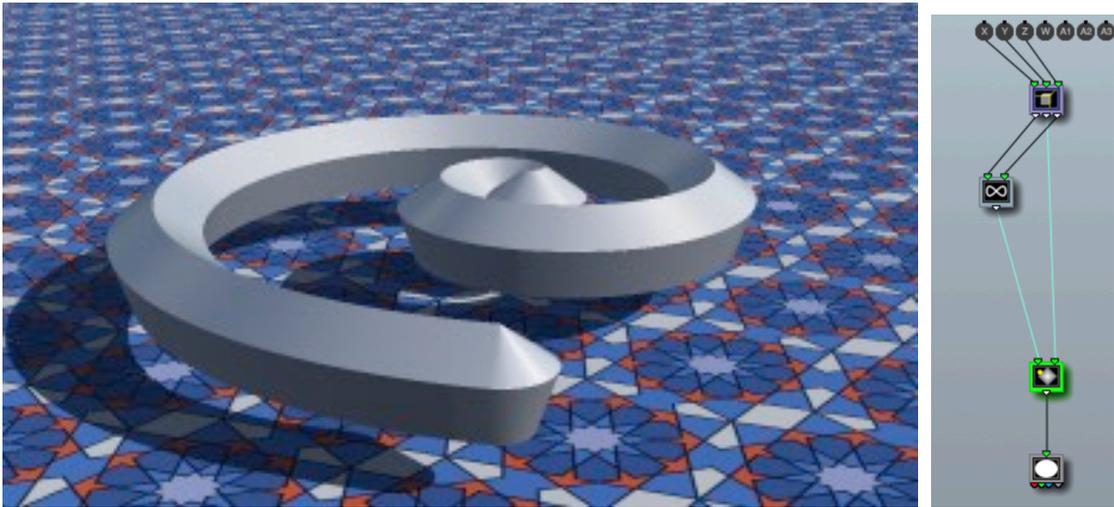


- **Sweeps:**

As DE fields can be used as the input coordinates for another field calculation, it is possible to use a 2D DE Field as an input (x or y or z) for another DF function either 2D or 3D.

This will basically 'sweep' the object B along the profile of object A. For example, to get a torus, sweep a disc in (x,y) along a circular path defined by disc profile in (x,z). Sweeping any profile with a disc will create a revolution (or 'lathed') object, like a glass or a bottle.

Usually, you will directly connect a 2D profile to one coordinate input of another 2D profile.

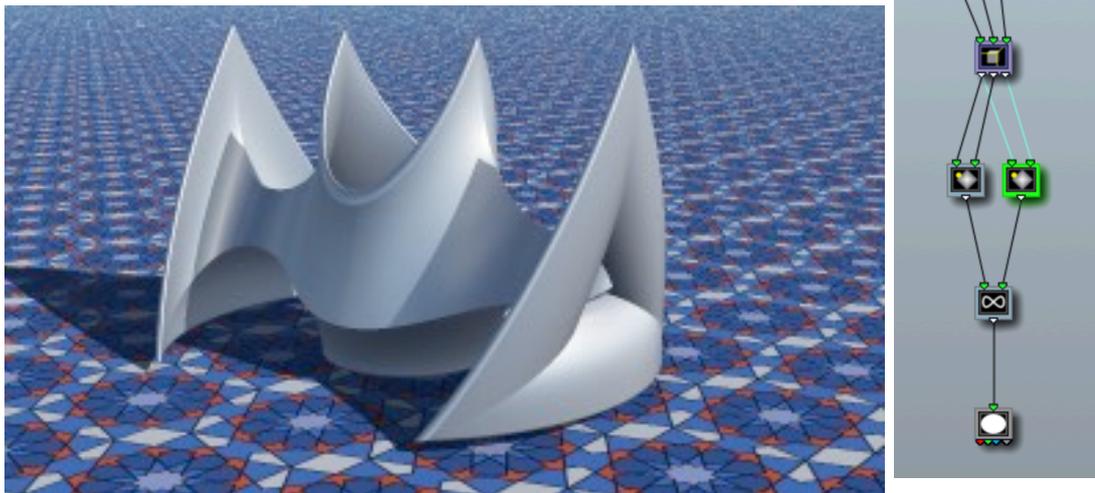


2D pentagon (21 Profile Shapes #) sweeps along an Archimedes spiral path (21 DF Curves #)

- **Cross Sweep:**

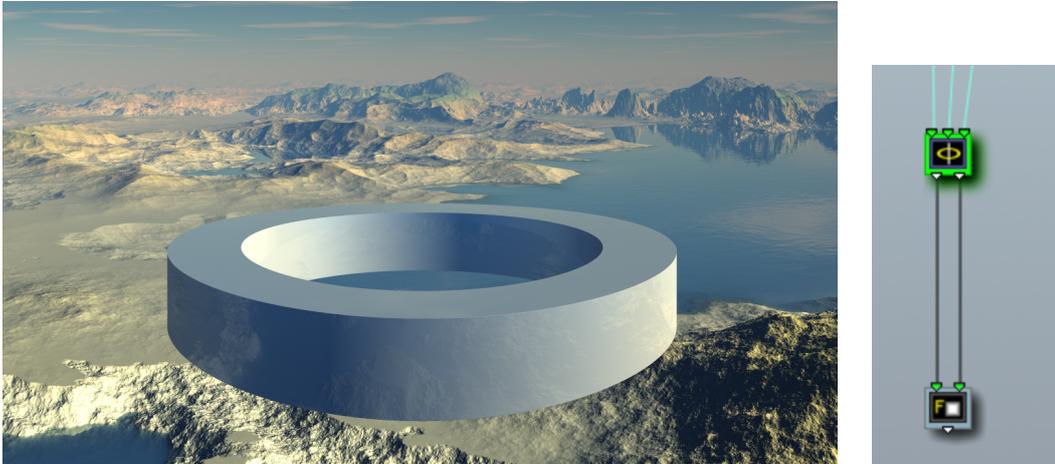
A cross-sweep will be achieved when a 2D profile is connected to two others 2D profiles, one fed in x input, the other in y input.

Quite complex models can be achieved that way.



A 2D Triangle and a 2D disk both feed coordinates of a 21 DF Curves # Convex Tower.

Another way is to use the **32 Revolution & Sweeps #** component that provides many paths for sweeps. When 2D uv coordinates are needed, you may also use the **34 uvid Sweep Volumes #** component that will perform sweeps internally and return uv as well as the DE Field itself.

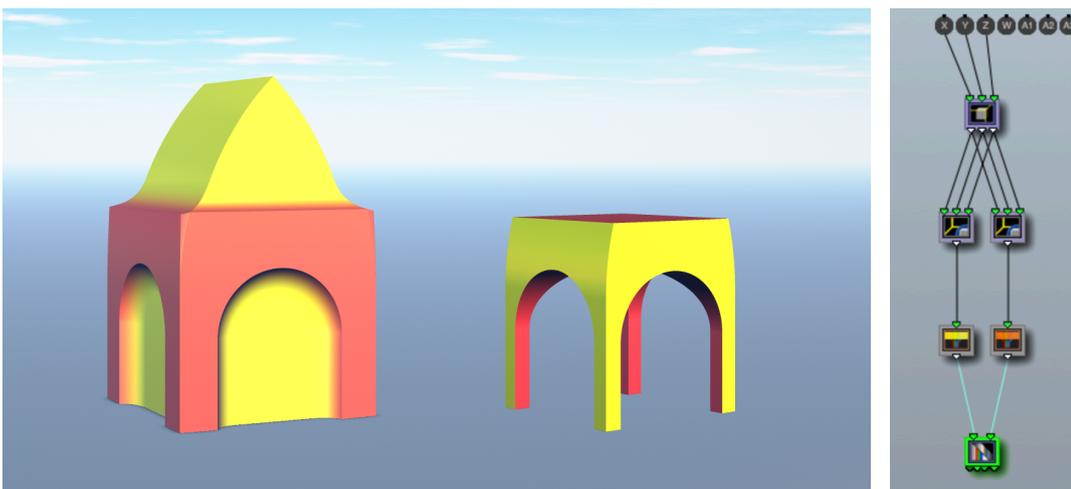


Circle Revolution (Revolution & Sweeps #) of Pyramid Cubic (Elevation shapes #)

Techniques for working with 3D Profiles:

There are many ways to work with 3D DE fields; the techniques below can all be combined to achieve quite complex geometry.

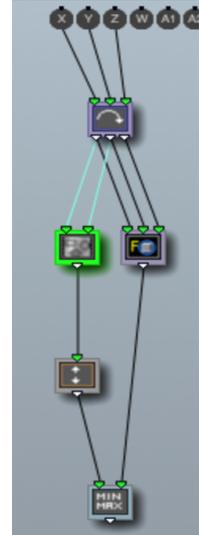
- **Boolean Transformations to create composite objects** (Intersection, Union.. etc)
Most of the time you will build complex 3D objects by mixing various 3D DE fields using Boolean (logical) operators like **21 Logic tools #**, **S:P Logic & Profiles #**, **24 Packed Logic #**, or **21 Min<->Max**. *Many examples of Logical operators usage are located in Voyager Examples/Components/Logic tools*



Union (slightly smoothed) and **Intersection** of red Arch-cut box and yellow Goth Arch (3D DF XYZ shapes #) using **24 Packed Logic #**

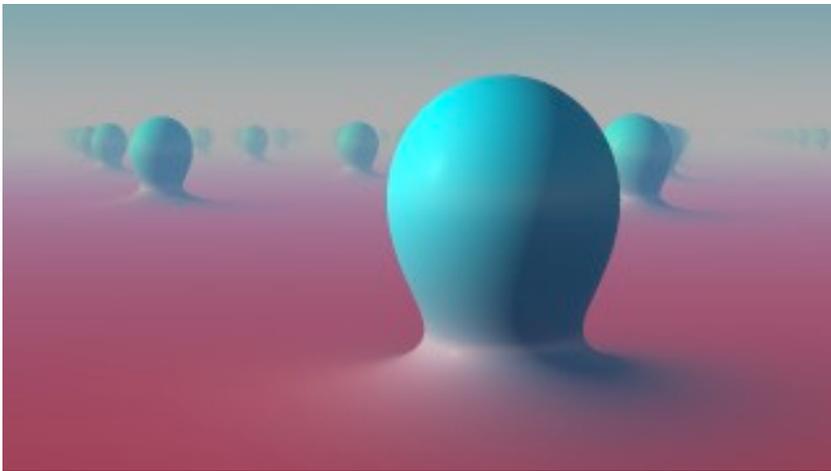


A 2D 21 Blurred Pict component mixed with a Panel xy (3D DF shapes A #) makes a true 3D version of the 2D image

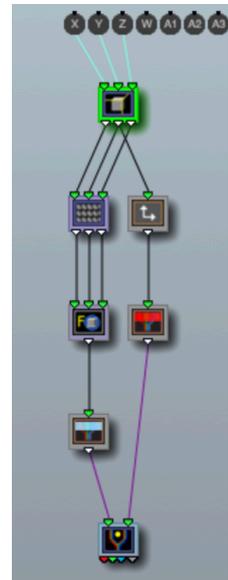


- **Morphing fields**

Use the Morph components to make a morphed union of 2 objects. For Scalar fields you can use the **21 Math Tools # 'morph'** function (*algo: morph log(e^x + e^y)*). To morph 2 colored DF objects you will use the **24 tile Packed Morph**.

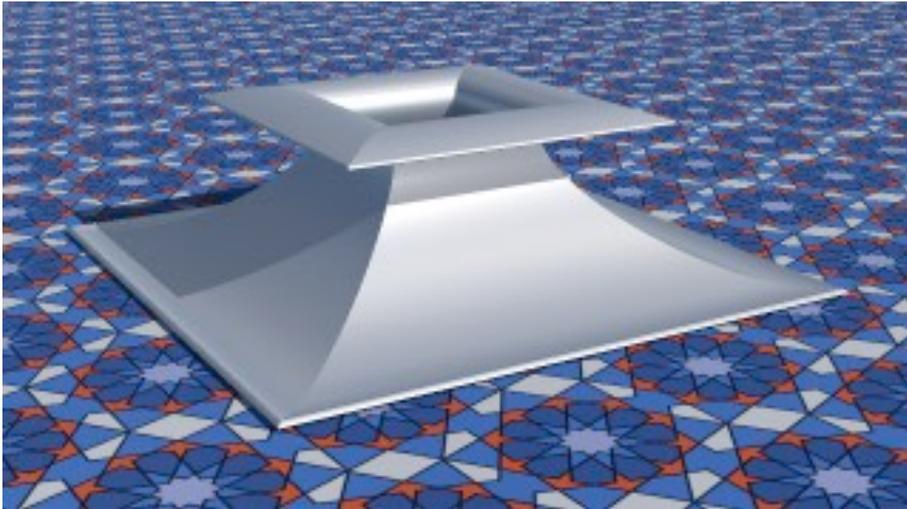


A cyan sphere array morphed with an infinite red DF plane



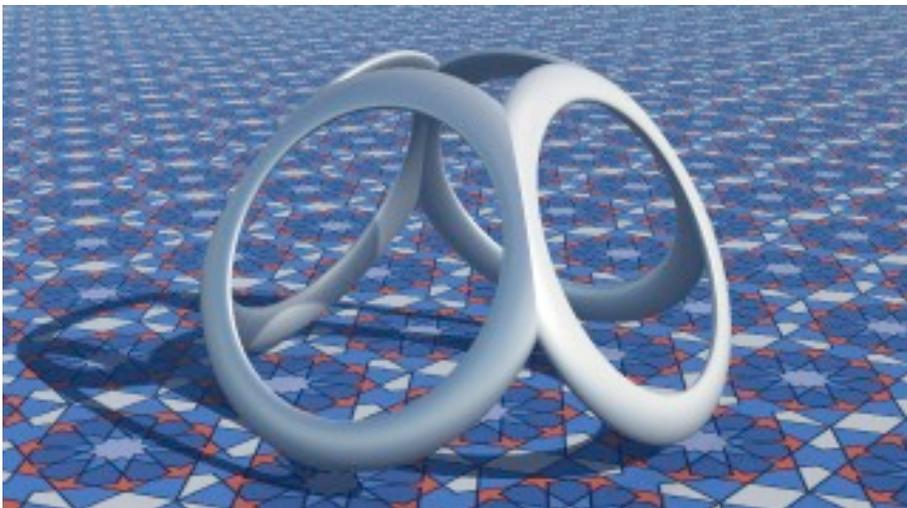
- **Sweeps & cross-sweeps**

Sweeps can also work between 3D and 2D DF objects. You 'sweep' a 2D DF profile along a 3D DE Field by feeding the 3D object field to one of the 2D DF profile coordinates.



A 2D arc curve sweep along an four sided 3D pyramid

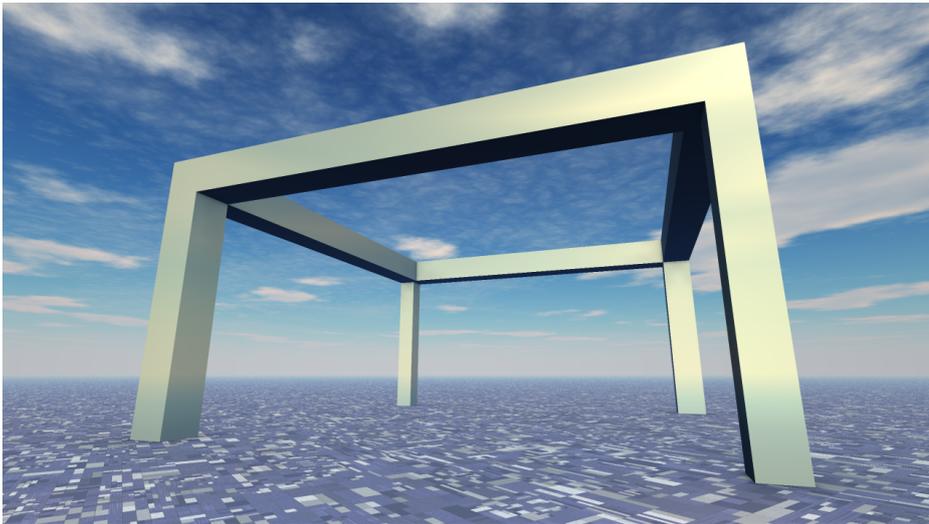
It is also possible to sweep a 2D profile along the intersection of two 3D DF volumes: In that case the 2D profile will trace the shape of the intersection contours.



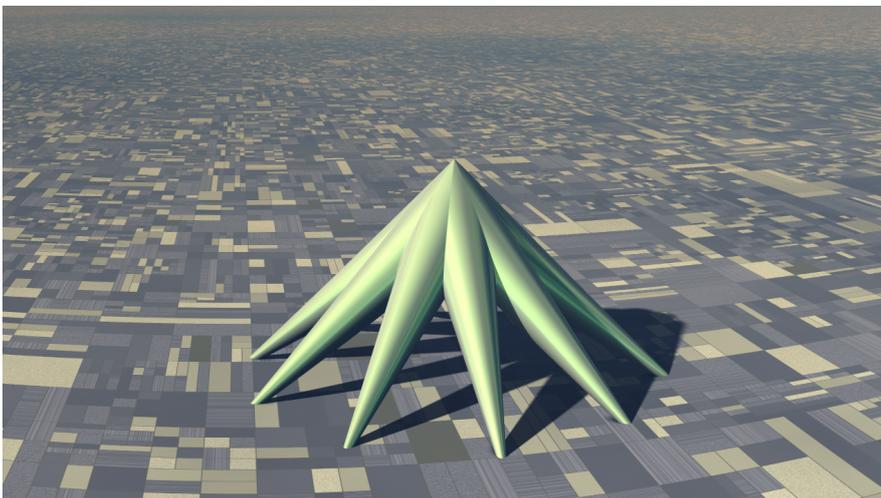
A 2D disc swept along the intersect of a sphere and a four sided pyramid

- **Space Deformation**

A very efficient way to shape the DF object is to add a space-distortion function to modify the incoming space. Mirroring and Rotations are very often used to force the object to be symmetrical, or to have various number of rotational symmetries. Mirroring, rotations and translations are transforms that keeps the length invariant, so they are the best candidates for DF modelling as they won't distort the distance field. **33 3D Plane Mirror**, **33 2D Mirrors & Rotates #**, and **33 3D N-Fold Mirrors & Offset #** provide 3D Mirroring and Rotations functions.



Four *3D Plane Mirrors* act on *Dot, Plane & Line* component: algo 3D DF Square Line



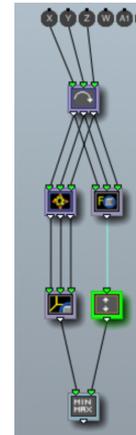
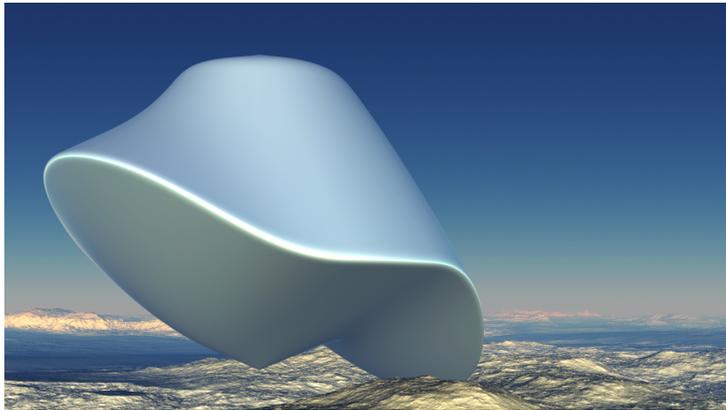
3D Mirrors & Rotates (algo: Mirror N xz & Rotate 30 zy #) acts on a Cone (*3D DF shapes A #*)

A 3D fractal displacement like **33 3D Fractal Displace** will completely change the look of a simple sphere.



DF sphere and ground with 3D space displaced by *3D Fractal displace*

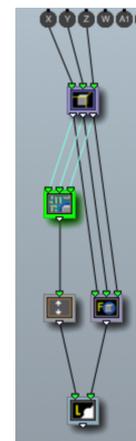
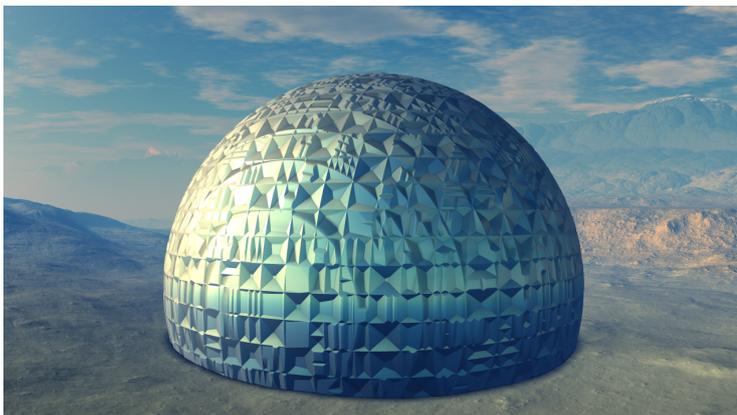
Some displacement functions are specifically designed to deform DE fields, like **33 3D Distorts & Bend #**.



A *33 3D XYZ Shape (smooth slanted cylinder)* deformed by **3D Distorts & Bend**, then mixed with a cone

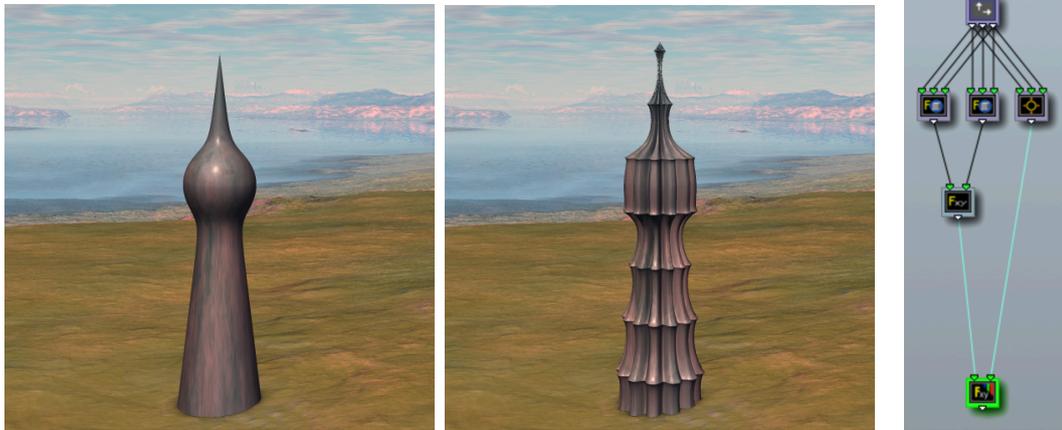
- **Displace the field value**

To add a bump texture or small details you can simply add a little bit of a 3D noise function to the field. A number of *3D DF noise* and *3D DF pattern* functions are available in ArtMatic Engine to add textures at the geometry level; but for small details almost every ArtMatic function can be used to modulate the DE Field.



3D DF Pattern # intersects a sphere

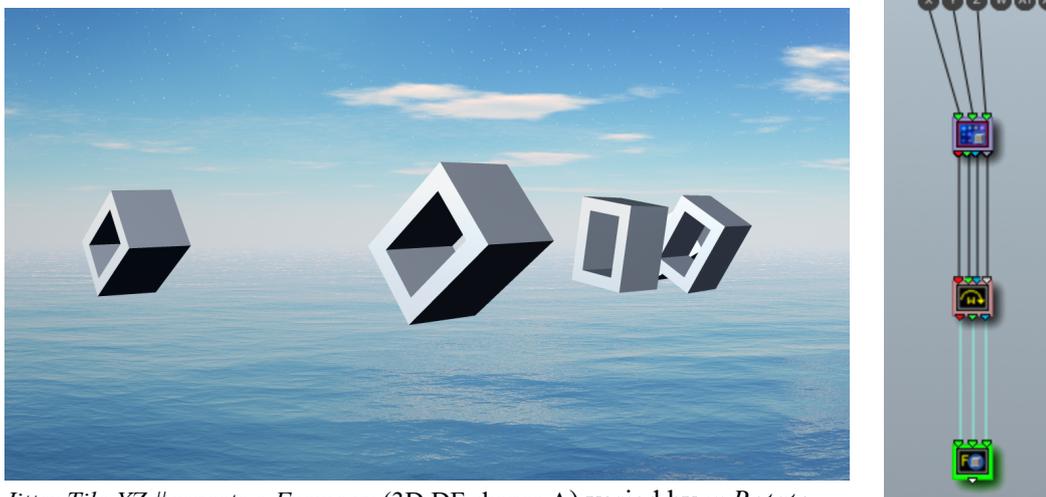
The **31 Radial Displace #** component creates radial displacements that provide interesting surface features to 3D objects, especially vegetation. This component provides quite a few different algorithms for very different effects. The displacements radiate outwards from a vertically oriented cylinder.



Left: Sphere morphed with cone. Middle: 'Radial Displace #' applied using 'S:P Maths # (algo Blend)'

- **Instantiation by space manipulation**

The most efficient way to duplicate and instantiate objects is to manipulate space so that a single object will appear at many places at once. A simple **1D Modulo** function will repeat the object in one axis infinitely, for example. One may use voronoi diagram (2D or 3D) to partition space into many cells, each having its own coordinates. ArtMatic Engine provides many components that create instances by tiling or partitioning the space: eg. **33 3D Repeats & Tile #**, **34 Jitter Spherical #**, **34 Jitter Axial #**, **34 Jitter XZ #**, **34 Motion Cluster #**, **42 3D Motion Path render**.



Jitter Tile XZ # repeats a Frame xy (3D DF shapes A) varied by w Rotate

The difficulty with this technique is that the DE field has to be well-centered and relatively far from the space cell boundaries, where space coordinates will suddenly be discontinuous and jump to totally unrelated values. To maintain an accurate Distance estimate and avoid overshooting one can clamp the DF value when using these, to not be below a fixed value. When the tiling is regular and space symmetrical the problem vanishes as the space is coherent even at cell boundaries.

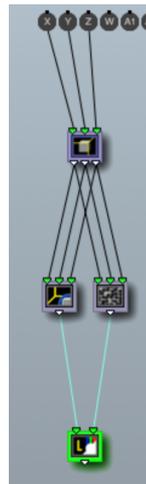
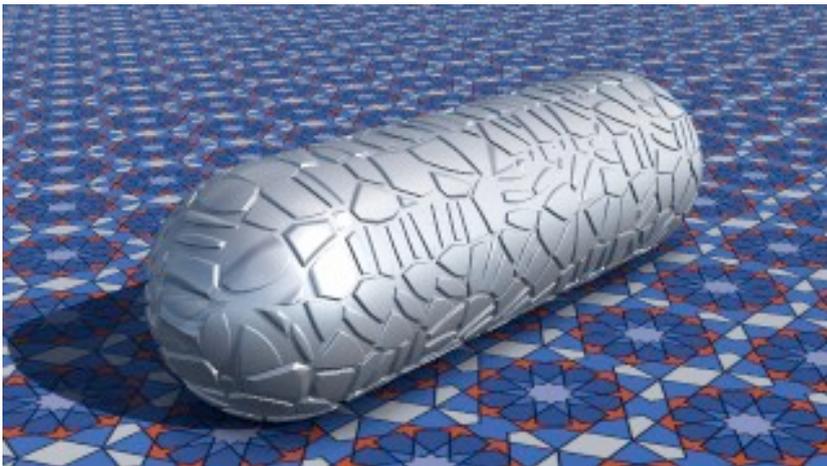
A **24 3D Looper** component (tied to a memory component) may also instantiate objects.

- **Carving**

A 3D pattern or noise component can provide details to any 3D volume, using 21 **S:P Logic & Profiles** displacement functions; like 'Displace', 'Chisel displace' and 'Circle displace'. They will carve the volume along the contours defined by the zero crossings of the 3D volumetric pattern.

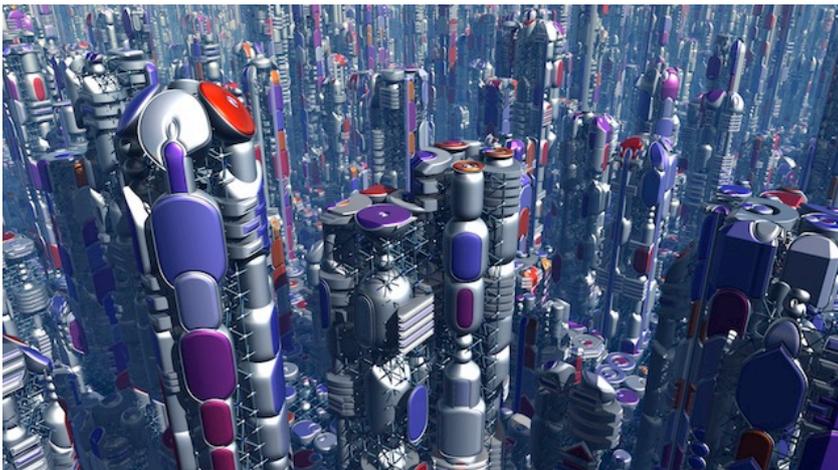
You can also carve an object using intersection or subtraction booleans.

ArtMatic Engine 8.5 provides new functions to carve a DF texture, like 21 **S:P Logic & Profiles #** (*Texture Intersect*) and (*Fuzzy Intersect*). Fuzzy Intersect is also available in 24 Packed Logic #.



An elongated *sphere* (31 3D XYZ shapes) carved with a voronoi split pattern (31 3D Bubble & Skins) using 21 *S:P Logic & Profiles #* (*Displace*)

A simple DF surface carved with a complex RGBA DF pattern can yield incredible results as the carving texture provides both colors and geometry.



Jittered City shapes color-carved with the "conics elements" RGBA df pattern

Shading DF Objects: shading modes

Textured or not, Voyager provides several rendering and shading options for DF objects. In general, you will use the **Volumetric opaque mode** (in Voyager Objects inspector), but alternate modes can provide clouds and gazes, fuzzy objects, lights fields and transparent/translucent objects. Examples are provided in */Voyager Examples/Shading & Rendering/folders*.

Object Shading modes are:

- ***Volumetric opaque:***

This mode shades the 3D object as an opaque solid object. If the ArtMatic system has only a single output value, the output defines the object's shape and the color is white (but the apparent color can be changed using the object's specular/reflection properties).

If the ArtMatic system provides RGBA output, the alpha channel defines the object's shape and the RGB outputs provide the object's colors.

ArtMatic extra outputs (X Outs) are used, if there are any specified in the Voyager settings. Volumetric opaque can be used for a mind- boggling variety of objects and features.

- ***Volumetric light:***

This mode shades the DE Field as a volumetric light density field by accumulating color/opacity values along the ray. It is suited for a wide range of luminous effects: fire, city lights, array of lights etc.

- The *occlusion slider* determines how much light from the background is occluded by the object.
- The *light density parameter* scales the distance field interpreted as density values when inside the object. You often need to adjust it when the light gets too saturated or too bright.

This mode is slower than volumetric opaque as the object (its density field) needs to be scanned inside and out (whereas evaluation of an opaque object stops where the light rays meet the object's exterior).

Volumetric light objects can cast light.

- The *light emission range parameter* controls how far light are cast from the distance field center.
- The light direction is taken from the normal of the DE Field, unless you use the "*Shade as projector*" mode, in which case the object center becomes the source of light. The light cast with DF normal may be physically impossible and won't cast shadows, but nevertheless is quite efficient to render multiple lights or complex light field like cities streets lights.

- Alternatively, you may have an *extra output defining the light direction vector*. In that case you can automate the Xout settings by using the letter b at the end. The 'b' tagged Xout vector will define the light direction vector. In that case the light field will cast shadows.

Examples: Voyager *Examples/Shading & Rendering/DF lights fields*



In the Desert Light Field example, you can see an array of lights casting light on the desert.

- ***Jitter opaque:***

This mode replicates an object throughout an environment, with small variations so that the repetitions are not identical. Voyager essentially breaks up the environment into randomized cells and instantiates one copy of the object into each cell with a 'jittered' (randomized) center and rotation. ArtMatic global A3 is sent a unique randomized value for each cell that can be used to randomize the object properties. You can use this technique to create an entire forest from a single tree.

If you use an ArtMatic system that itself uses a jittering component, make sure that the ArtMatic system's jittering clip radius is smaller than Voyager's jitter cell size, and keep the object small enough so it stays away from cell boundaries. This may require some experimentation to find the correct parameter values. For a greater control you would usually use a jittering tile within the ArtMatic tree.

- ***Volumetric & translucent:***

This shading variant of opaque mode is dedicated for vegetation (leaves and plants) shading. It adds some light going *through* the object, and light *scattered inside* the object surface. The thickness of the object is important, as a thin object (a leaf for example) tends to be more translucent than a trunk (obviously).

The 'light transmission' and 'light transmission range' parameters control how much and how deep light can travel through the medium. 'light transmission range' ranges 0 to 200 meters. Light travelling inside the medium is colored by the reflection color and the object texture color.



Backlighting a tree will still have light passing through the leaves in this mode

- ***Fractal opaque:***

This mode is designed for fractal objects, and objects with very rough surfaces, as it smooths the sub-pixel details that would otherwise make the image noisy; especially when far away.

The global 'high-frequency limit' and the 'fractal object detail %' (in the *Preferences dialog*) can further allow you a fine control over objects' details.

Use this mode for fractal objects with very rough or infinitely thin structures; like the MandelBulb, the MandelBox and similar created with 32 **3D Fractal sets #**.

- ***Transparent (surface):***

This mode is suited for transparent objects like glass or windows. The object surface is treated as transparent, without internal volumetric shading nor refraction calculations. Light color is tinted as it passes through the object much as it would be affected by tinted glass.

The stained-glass windows in the **Opaque + Transparent** mode example below are treated as transparent surfaces. Notice how they project their colors on the ground and walls (provided 'cast shadows' is switched ON in Voyager). Transparent mode does not generate new rays and is faster than Transmissive modes.



Voyager Examples/Shading & Rendering/DF Special Shaders/NF12 stained glass O-Transparent.vy

- ***Opaque + Transparent***

Combines Opaque mode with Volumetric Transparent mode (see above). The transparent volume has to be provided by a second output of the ArtMatic file.

Like the other multi-modes, this mode requires an ArtMatic system that has two sets of outputs: one for an opaque object and one for a transparent object. The second object is interpreted as a transparent and reflective object. It can be colored, but light is not accumulated volumetrically.

True reflections are disabled for the opaque part, so they will only apply to the transparent parts. This mode is particularly useful for creating objects that have windows in architectural design.

- ***Transmissive (surface):***

Introduced in Voyager CTX 1.2, "transmissive" can render refracting material. Transmissive offers several air/medium indexes of refractions, with a single ray or multiple rays.

While a single ray gives physically inexact results for bounded objects, they are fast and can produce nice and less noisy results. A single ray is sufficient for water planes, for example, where there won't be exit from the medium from the other side.

With multiple ray mode it is possible to place the camera inside the object.

The surface mode only handles the rays at the intersection, and doesn't perform volumetric density estimation, unlike the "Transmissive (volumetric)" mode. Specific parameters:

- Surface shade controls how much the surface is shaded in balanced with light coming through the medium with refractions. When Surface Shade is at maximum the object is completely opaque.
- Tint gain controls how much light going through the medium is colored by the object color. Use strong values for stained glass for example.

Transmissive (surface) has the following options:

- Helium (1 Ray) index of refraction 1.025, and very close to Air/Air: 1.
- Jelly (1 Ray), hypothetical medium with index of refraction at 1.125
- Water (1 Ray) index of refraction 1.333
- Glass (1 Ray), index of refraction 1.52
- Helium (MR),
- Jelly (MR),
- Water (MR),
- Glass (MR),
- Diamond (MR), index of refraction 2.417

Multiple rays implementation. Note that ray numbers are limited to 4 in good quality and 6 in higher.

Examples are provided in Shading & Rendering/DF Special Shaders/



Transmissive jellyfish

- **Transmissive (volumetric):**

This mode offers several air/medium indexes of refractions with a single ray or multiple rays. Unlike the Surface version, it can also accumulate opacity along the ray for volumetric density evaluation, with the parameter 'Opacity gain' controlling volumetric density.

The volumetric density shading can be a simple diffuse shader (shaded variants), or just take the object color unshaded (unshaded variants), which corresponds to the previous version's "self-illum mode", when ambient level is above zero. The surface is shaded with the specular color and blended with a volumetric accumulation of the inside of the DE field. Specific parameters: 'Surface shade' and 'Opacity gain'

Transmissive (volumetric) mode has the following options

- Helium (1R, unshaded),
- Jelly (1R, unshaded),
- Water (1R, unshaded),
- Glass (1R, unshaded),

- Helium (1R, shaded),
- Jelly (1R, shaded),
- Water (1R, shaded),
- Glass (1R, shaded),
- Diamond (1R shaded),

- Helium (MR unshaded),
- Jelly (MR unshaded),
- Water (MR unshaded),
- Glass (MR unshaded),
- Diamond (MR unshaded),

- Helium (MR shaded),
- Jelly (MR shaded),
- Water (MR shaded),
- Glass (MR shaded),
- Diamond (MR shaded)



Transmissive self illum jellyfish

- ***Fuzzy loose:***
This is a faster version of "Fuzzy" mode that renders less accurately than "Fuzzy" by sampling the volume much more sparingly. Use this mode if the other is too slow for fast previewing.

- ***Fuzzy***
Only the volumetric interior is rendered and shaded by accumulation and there is no surface shading. Specular is off in this case. "Fuzzy" shading mode can be used for fuzzy objects, and even plants.



Fuzzy Jellyfish



Fuzzy Vegetation (grass and bushes)

- ***Gas & Clouds***

In this mode, Density objects are shaded like clouds. This provides an alternate, more flexible and controllable solution than the volumetric clouds. With 'Gas and Clouds' you can make smoke, steam, fog, clouds, and even vegetation for an impressionistic approximation when seen from afar. Shading parameters are:

- **Opacity gain:** scales the density of the gaz.
- **Self shadow dist:** length of self shadow accumulation ray
- **Self shadow gain:** strength of self shadow
- **Derivative level:** should be zero in most cases, as the derivative mostly captures surface details that are not really there for true gases.
- **Contrast:** global shading contrast.
- **Ambiant level:** amount of light scattered from the environment and passing through the medium.



Examples/Shading & Rendering/DF Gaz &Cloud shader

- ***Opaque + light***

Combines Opaque mode with Volumetric light mode (see above). The volumetric light has to be provided by a second output of the ArtMatic file. The second output must be an RGBA component. 'Opaque + light' is suited to create lamps, illuminated cities, and for special effects like lights rays or reactor exhausts coming out of a space ship.

To obtain a lamp that cast shadows, you will need to add an additional output to set the light direction. Usually, the coordinates of the source is enough to set a point light direction, as the vector is normalized by voyager. For a parallel light use a constant vector.



With volumetric light casting real light you can have "Opaque + light" lamps that can be manipulated as a single object



Utopia city combined with a DF lights field.

- ***Opaque + Transmissive***

Combines the Transmissive (surface) mode with Opaque in the same way as Opaque + Transparent for 2-output DF ArtMatic systems. First output provides the opaque parts, second output the transparent/transmissive parts.

Opaque + Transparent does not perform true refraction, and is faster in any cases. True reflections are disabled for the opaque part. so they will only apply to the transmissive parts. Note that starting ArtMatic CTX 1.2 a new global RGBA shader to visualise 2 RGBA outputs systems where A is the distance field (DF)



Transmissive glass and bottle (examples: 'Still Life Refractions')

Ambient occlusion

Ambient occlusion approximates how much light coming from the environment is blocked by the object, in addition to true shadows. It provides a kind of clarity and realism not possible without it; especially when directional sun light is absent, like in overcast sky situations. When rendering rough textured terrains or fractal objects, ambient occlusion is especially helpful for bringing out the scene details.

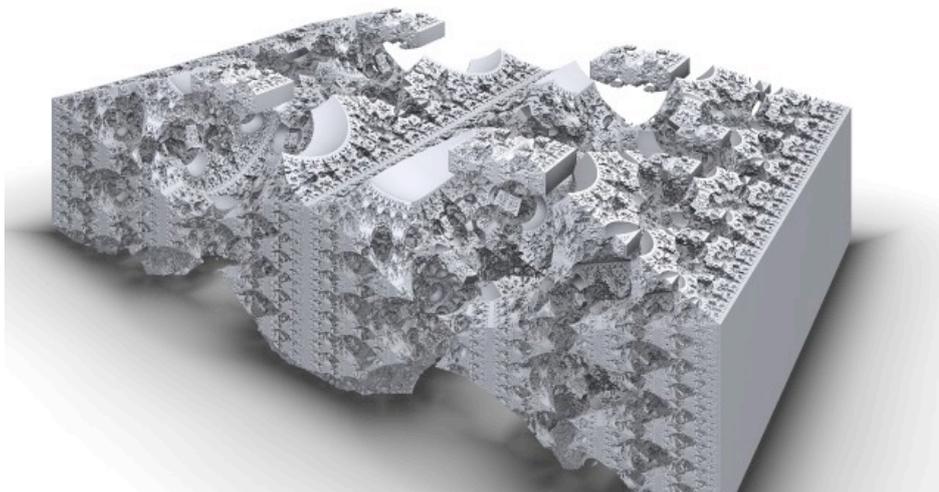
Ambient Occlusion estimates the amount of non-directional ambient light that reaches various areas (as opposed to shadows which are caused by directional light). AO is independent of the main light direction. Concave or hard-to-access areas will be darkened.

Ambient occlusion can be applied to terrains and objects independently. Ambient occlusion affects ambient and diffuse lighting, but not specular and reflective light channels; because it mostly simulates the blocking of light coming from any direction, but not the light hitting the surface from a single direction.

Ambient occlusion can take some time to compute and is set to OFF in draft mode.

DF objects provide several algorithms for Ambient Occlusion. The Low freq AO is the most accurate but also the slowest. There is a global preference for the AO Radius in the main preferences dialog, but each object in a scene can have its own AO amount setting.

- **AO Amount.** When the AO amount is *less than 100%* only convex surfaces are affected. Amounts greater than 100% tend to affect all areas but may leave convex areas intact.
- **AO Radius Preference.** Voyager scenes can have varied needs. Preferences contains a global control for the Ambient Occlusion Radius that allows you to adjust the AO for the scene's context. On a landscape dominated by large-scale features, a size of 50 meters or so will give good results. Changing the radius will influence features of certain size or detail. The same object can look quite different with different settings. So, it is worth some experimentation to find the setting that gives you the results you prefer.
- Ideally, AO should be scale-independent, but this is currently impractical because of the enormous impact it would have on render time; so AO radius is scaled by the DF object scale when below and above 100%. This allows the same scene to have a (say) 40 meter AO radius for terrains and large DF structures, while still having correct AO for a small 20 cm object in the foreground.



Example of a fractal DF object entirely shaded by Ambient occlusion

Using extra outputs

When an ArtMatic DF object has one or more extra outputs, the extra output can be mapped to various shading properties such as wetness, self illumination or reflectivity. Using extra output (or "X-outs" for short) to modulate texture shading can greatly enhance realism and visual complexity.

For example, you may have a model that provides a texture for day light, and a texture for night in an X-out channel. Turning on the channel by selecting "Self illumination" is easy, and you will typically do it for night renderings, without having to change the model itself.

The extra output can be mapped to:

- **'nothing'**: the way to turn off a particular shading option.
- **'Self illumination'**: Unlike 'Ambiant & Wetness' which sets the amount of diffuse reflection coming from the environment, 'Self illumination' adds its own light to the scene giving the impression of a luminous object. The self illumination light color is the X out color, or white if X out is scalar.
- **'Wetness level'**: controls the specular amount of light coming from the environment. This light can be filtered by the X out color, if any.
- **'Ambiant & Wetness'**: control the amount of light coming from the environment, diffuse and specular. This light can be filtered by the X out color if any.
- **'Reflection level'**: Mirror reflection of light from the environment. The reflected light can be filtered by the X out color.
- **'Bump Map'**
- **'Light Color Filter'**
- **'MFD Params & Specularity'**
- **'Light Direction'**

Auto Mapping of X-out option: The following letters put at the end of the name of an ArtMatic 3D DF object multi-output file will cause Voyager to set a proper shading option by default, when opening/importing a new AM system.

The letters can be combined in any order, up to 3 letters (eg. "ri", "wir" "wbi" etc) when multiple X-outs are used; but a *space* needs to be present before the letters to avoid confusion with letters in the name itself.

- **'i'**: sets corresponding output to "Self illumination color & level",
- **'r'**: sets corresponding output to "Reflection color & level",
- **'w'**: sets corresponding output to "Wetness level / Specular color",
- **'b'**: sets corresponding output to "Bump Map",
- **'l'**: (in first position only) will set the first extra output to be assigned to "Volumetric Light" in the mode 'opaque + light',
- **'t'**: (in first position only) will set the first extra output to be assigned to "Transparent" in the mode 'opaque + Transparent'

Performance tips

When working on a slow computer or working with a particularly CPU-intensive DFRM based scene on any computer, you may find times when making adjustments becomes very difficult, because the CPU is bogged down calculating the preview, while you adjust sliders and other user interface controls. Or simply because feedback is too slow to be practical. When this happens, there are a few tricks that you can use to improve the user interface's responsiveness.

Reduce render quality.

The first thing to try is setting the Render Quality setting to *draft* or *good*. In some cases, this provides a dramatic improvement. Work at the reduced quality setting until you actually need the higher quality. "Draft mode" now cuts out any extra rays and should be used systematically when setting up a scene and positioning elements. In rendering, using *Best* or *Sublime* quality is very slow and unnecessary most of the time. There are many cases where a Good or Better quality render is almost indistinguishable from a Best or Sublime render -- and the lower quality render may take 1/10th the time.

Make objects temporarily inactive.

With very CPU-intensive systems - especially with object's that are reflective or transmissive or very slow like fractals - it is often useful to temporarily *make the object inactive* in the ArtMatic Object inspector. While inactive, make whatever adjustments you need (sun position, camera position, etc.), and then make the object active again. You may also temporarily *turn off all reflections* by setting quality to draft mode.

Make texture computation separate

Finding the intersection between the ray and the object is the most CPU intensive task with DF objects, especially if the object field is poorly convergent. Texture computation is not necessary at this stage and should be put in a CT (Compiled tree) set with the *Compute only for colors*. Sometimes the texture algorithm is much more complex than the object volume function and you really don't want it to be computed for every sample along the ray when it is only needed for the object shading. For constant colored objects, or simple and fast colored textures, it may be not worth the effort to do that ,as there is a little overhead when using CT.

Temporarily change the sky mode.

If your scene uses objects and volumetric skies or the volumetric light sky mode, you may want to temporarily set the Sky Mode to Clear Sky or Cloudy Sky. Volumetric clouds and volumetric lights can be very cpu-intensive.

Turn off Cast Shadows and Ambient Occlusion

The Cast Shadows option can dramatically increase the calculation time. In some cases, this option can increase rendering time by as much as a factor of 10. Turn it off until you need it. If you are rendering animation, you should render some test frames to see if the option is worth the added rendering time. Ambient Occlusion can be set to "none" for each object; or is globally bypassed in Draft render mode.

Troubleshooting DF objects

Object invisible

Make sure the object is not below ground, too small, or outside camera range.

Object artefacts

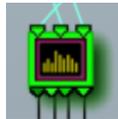
Artefacts in the rendering or shadows usually means the DE Field is inaccurate and convergence poor. Revise the maths of the field or lower the field amplitude to make convergence safer.

Black screen

Camera is probably within the object. To fix this situation, move the camera outside the object's bounds. Otherwise, it may be that the field has no more any "outside"; that is, negative values that define the empty region around the object. Revise the maths of the field to ensure it does have an "outside".

Using DF Vegetation Components (AM 5 New Features)

3>4 Color Grass Field



This component creates an infinite field of grass for use with Voyager DFRM objects. The output is *color* (the left-most 3 outputs) + *field value (object)*. The field value defines the shape of the blades of grass. This component is faster than using a jitter function with a repeated single grass object.

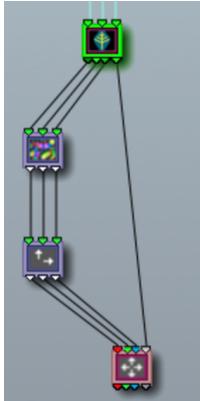
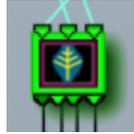
Parameters: A: Modulation Scale, B: Modulation Amount, C: Contrast.

Option: this popup provides several different grass types. Those with "random" in the name have randomized **colors**.



Random Yellow Grass using the follow terrain Option

3>4 3D Leaves #.

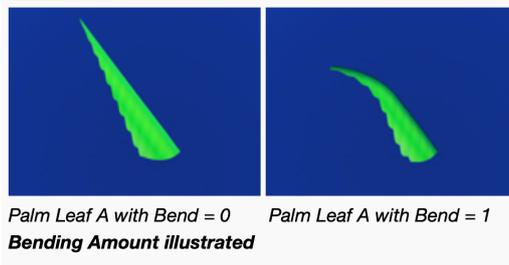


This component creates a variety of leaf-shaped density fields that are especially useful for creating leaves or other DFRM objects for ArtMatic Voyager. A typical structure tree demonstrating this component is shown below.

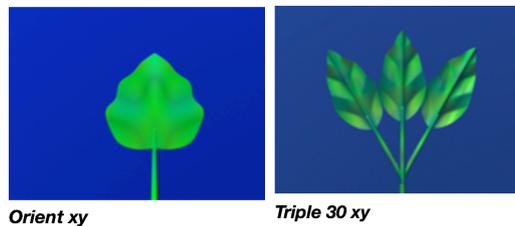
The input is a 3D space and the output is a 3D space warped into a leaf shape (which will be sent to a color texture function) and a density field (which will be used in Voyager to define the leaf shape).

Leaf style is selected from the algorithm popup.

Parameters: **A:** angle, **B:** Bending amount, **C:** Shape



Parameter Options: Orient xy, Orient zy, Symmetric 30 xy, Symmetric 45 xy, Symmetric 60 xy, Triple 30 xy, Triple 45 xy, Fan xy.



EXAMPLES.

There are many examples of this component's use in the Voyager's *Examples/DF Plants and Trees* folder, and more in Voyager 2 examples. In particular see the example '*leaves Demo.vy*' in Voyager 2 examples.

4>3 3D Tree Structures



This component creates a tree-like branching of space using a variety of techniques (symmetries, rotations, translations and repetitions) that are especially useful for creating trees and plants for ArtMatic Voyager. When used inside a recursive CT (compiled tree), this component may be sensitive to the iteration level (depending on which tree algorithm is selected); which results in a more varied structure than if each level were simply a scaled down version of the parent.



[Left] ArtMatic Voyager tree **[Right]** 3D Tree Structure used inside a recursive compiled tree

To create an tree-like object, the component is used in conjunction with a component such as 3->1 *Density Shapes* , since by itself, it is just a space distortion function. Most users will want to use one of our many examples as a starting point for their creations since using the component from scratch requires a solid understanding of ArtMatic structures and recursion. However, it is fairly simple to make modifications to the many example files that are provided with ArtMatic Voyager.

Of course, you are not restricted to creating trees for ArtMatic Voyager.



Use this component to create non-tree-like objects, too. A sphere transformed with a single recursion of 3D Tree Structures

Inputs and outputs. The four inputs are a **3D Space** (the left 3 inputs), and **w**, which acts as a random seed for the random numbers used internally to determine the branch spacing. When the 4th output of *Jitter Tile xz#* is connected to **w**, it will randomize the aspect of repeated trees. Or, you can connect w to global input A3 if *Voyager Jitter Opaque* mode is in use. To control the range of w related modifications, you can use the [Obsolete??1->1 Math component set to the Fract algorithm?] to limit the range from 0 to 1 and add a scale and offset so that you can control the minimum, maximum and amplitude of the variations.

The best way to get started using this component is to start from one of the examples in *Voyager's 2.0 Examples/DFRM Life & Plants/Plants iterative* folder.

Parameters:

- **A: N Rotate** - For most algorithms, this is the number of symmetries of rotation. Some algorithms, such as Alternate 180, only use this value for iteration #3 (in a recursive CT).
- **B: Angle A** - The main branch angle. Low values result in branches pointing upward. 90 degrees will be perpendicular and high values create branches angled down.
- **C: Angle B / Slope %** - The slope or a second angle. The meaning depends on the particular tree algorithm. The slope controls how much the length of the branches diminish going up. At 0, the branches are parallel. For some algorithms (such as cactus) it controls the bending. For some (palm trees) is the sub-branching angle.

Algorithms Illustrated (3D Tree Structures)

The tree algorithms are also referred to as "tree rules" since strictly speaking some of the algorithms are rule sets sensitive to the number of iterations.

0 - Palm Tree - All branches start from the same center. nice for palm-like structures



1 - SR Tree B1 - When iterated (recursed), the first iteration level can have N branches while the additional iteration levels just have N branches created with rotational symmetry off of the parent branch.



tree rule 1 (3 iterations)

2 - SR Tree BN bend up - All levels can have N branches with some randomization

3 - SR Tree BN bend down - All levels can have N branches with some randomization

4 - Random SR Tree BN - All levels can have N branches with some randomization



tree rule 4 (3 iterations)

5 - Random SR Tree B1 - Similar to rule 1 but branch direction and length are randomized.

6 - Cactus - Branches grow alternately left and right. When iterated it produces a structure like the one below:



7 - Alternate 180 - branches grow alternately left and right.

8 - Alternate Ferns,



9 - Rotating Branches - branches grow in alternating fashion with a global rotation controlled by parameter C.



tree rule 9 (3 iterations)

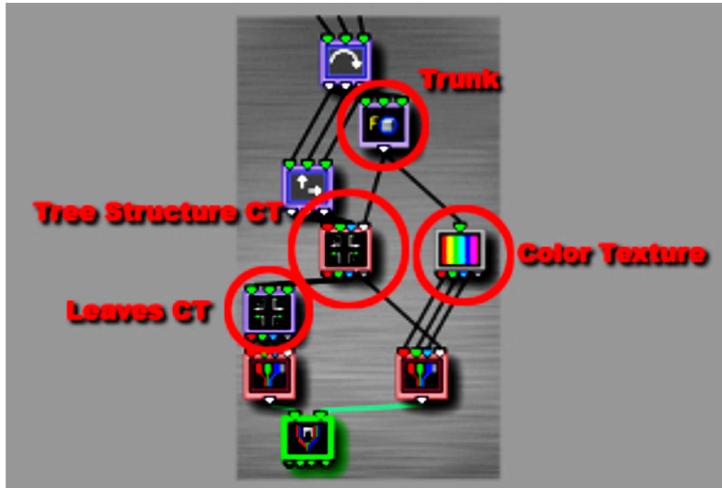
10 - Rotating Branches & SR - Similar to 9 for the first levels but then grows clusters of rotating branches whose number is related to parameter A. Here an example of tree rule 10 with leaves attached with SR 8:



Lesson: Using 3D Tree Structures

Analysis of a basic structure for creating a tree

The generation of a trees/plants is a complex problem. 4>3 *3D Tree Structures* can create a wide variety of plant structures. A typical structure tree for a tree/plant is shown below:



A typical ArtMatic structure tree for trees and vegetation

- There is a section (a single 3->1 *Density Shapes* tile in this structure) responsible for creating the trunk.
- The trunk shape is passed (along with 3D space) to a *recursive compiled tree* that creates the branches (it includes the *3D Tree Structures* component,
- The trunk+branches are fed to a compiled tree that adds the leaves.

The compiled tree that handles the branches is generally a recursive compiled tree. The number of levels of recursion determines the number of layers of branches. Some tree algorithms vary the branching pattern according to the recursion level. More details are found below.

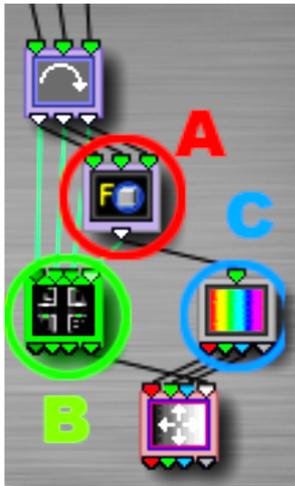
A SIMPLE STRUCTURE: TRUNK + BRANCHES: Below is a very simple structure that provides a trunk and branches and colors them with a simple color texture.



The outer level structure [ArtMatic system: Angles 43 tree iter from the Voyager example Tree rule 1.vy]. The structure creates a trunk with branches and uses a simple color texture.



The same structure and recursion level (3) but a different tree rule (number 4 instead of number 1) creates a very different tree.

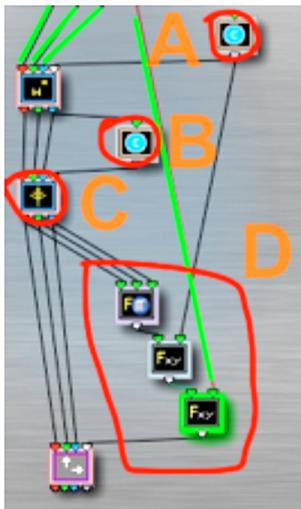


Main Structure: The main (outer-level) structure tree provides the tree trunk (A) and color texture (C). The recursive compiled tree (B) provides all the branches and sub-branches. The 3>3 3D Rotate component at the top is useful for adjusting the orientation.

Helpful hint: 3D Rotate is very helpful when designing any 3D object, as it enables you to rotate the object in space while working on the object, even if you don't plan on having the object rotated in your final scene.

The 3D space (from 3D Rotate) and the tree trunk (from 3->1 Density Shapes) are passed into the CT that adds the branches (described below). The tree trunk (A) output is also sent to the 1->4 Hue Color tile (C) which provides the tree's color texture. Hue Color's RGB output is sent to Alpha Scale and Offset which provides final adjustments of the completed tree.

The 'Branches' CT (Compiled Tree) (B):



In the structure shown above, the compiled tree (CT) labeled B provides the branches. Below is the internal structure of the compiled tree that creates the branches. This structure (with different parameter settings) is used for almost all of the plants and trees that we create. Adjustments of a few parameters can create a virtually endless variety of plants and trees.

[Left] The structure inside the compiled tree that provides the tree branches. The green lines indicate the input received from the parent tree: 3D Space and the tree trunk (a density field)

CT Inputs: The compiled tree receives four input values from the parent structure (which are indicated in green in the diagram): three space co-ordinates and the tree trunk density field .

Quick analysis: the incoming space is scaled (by the W^* component) and sent to the *3D Tree Structures* component, whose output is sent to *3D Density Shapes* (algorithm: 'Cactus Branch') and blended with the tree trunk (the density field received from the input). The output is the modified space (modified by 3D Tree Structures component) and the modified density field (which is the tree object).

Details: The *1>1 Set Constant* tile (label: **A**) is used as a scaling factor, by being multiplied against the incoming space using the *4>3 W Multiply* tile. Its output is also used as the y-input for the *2>1 Math Tools* component (in the group of components labeled **D**) where it has a further impact on the scaling.

The scale space is sent from *W Multiply* to the *3D Tree Structures* component (label **C**) - which is set to *Algorithm 1, N Rotate = 1, Angle A = 5, Angle B/Slope = 40.844*. The *3D Tree Structure* component's **w** input (which acts as a random seed) comes from the *1>1 Set Constant* component labelled **B** in the diagram.

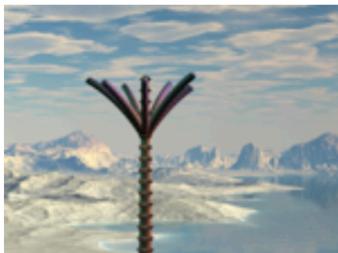
The *3D Tree Structure's* output is a space warped into a tree-like shape. This space is sent to *3->1 3D Density Shapes* tile, which is set to the 'Cactus Branch' algorithm. This provides the actual branch objects (the density field). The *3D Tree Structure* output is also passed to the first three outputs of the final component (whose job is just to pass out the values it receives).

The *3D Density Shape* (which is at the top of the group labelled **D** in the diagram) is sent to *2->1 Math Tools* ('Iteration Scale'). The resulting branches are thus automatically scaled by the number of iterations. Each successive iteration creates smaller branches. The adjusted branches are then added to the density field (the actual tree object) that was passed in to the compiled tree.

The resulting output is a density field (3D object) which is passed to the fourth input of the last component. The last component sends 4 output values out of the tree. The three coordinates of the branch-warped space, and the updated branching object (density field).

Recursion. The recursion parameter of the compiled tree determines the number of branching layers. Setting it to a value larger than 0 causes the CT to be looped, so that its operations are performed several times. This particular type of compiled tree is recursive which means that the compiled tree's output is fed back to the input each time that it is iterated (looped).

Each iteration of the tree creates a new level of branching as illustrated below. Some of this component's algorithms (the so-called *tree rules*) are sensitive to the recursion level and create different branching patterns at different levels of recursion so that different levels of branches are not simply scaled down versions of the previous level.



Recursion = 0



Recursion = 1



Recursion = 3

With this tree rule (algorithm 1), each level of recursion adds another set of branches at the end of the existing branches. Some tree rules add branches elsewhere as shown below.

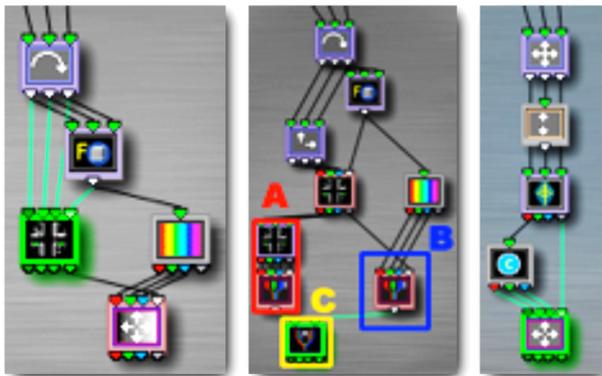


Tree Rule 4 with Recursion = 3

ADDING LEAVES: To add leaves to a tree, we need to add components to:

- add leaves to the trunk and branches
- blend the leaves with the tree branches.

Below is a "trunk+branches" type structure to which components have been added to provide leaves



[Left] Structure that creates just trunk and branches

[Middle] The structure modified with components to add leaves

[Right] The compiled tree that handles the leaves



(the tree with the added leaves)

Analysis:

- a compiled tree (A) was added that takes a 3D "tree" space as input and sends out a 3D color texture and a density field that corresponds to the leaves. Its output is packed with *4->1 Pack* so that it can be sent to a mixer.
- a *2->4 Packed Alpha Max* component (C) was added to mix the tree and branches with the leaves
- a *4->1 Pack* component (B) was added to pack the trunk+branches



A tree created with the structure tree above as it appears in ArtMatic Voyager.

Recursion and 3D Tree Structures:

When used inside a recursive CT, one must set the scaling ratio for each level carefully, to avoid having branches grow outside the plant's boundaries. Our example systems are already set up with the necessary scaling components, and should generally be your starting point.

In general, scaling values of 2 to 5 work best.

Because the structures are recursive, they are quite sensitive to parameter values, and can be computationally intensive.

Note: “option: *allow feedback (outs- -> ins)*” has been added for all recursive compiled trees. By default, the option is on and the CTs are recursive. If the option is turned off, the CT will loop non-recursively (and thus needs a memory component at the end to accumulate the loop results).