



Environment Production Pipeline

Created by Eric M. Scharf

Created On: 04-03-2007

Last Modified: 06-26-2007

This outline displays the production pathway for generating a *default* environment art asset (3D model, texture-map, UV layout, node attachments, collision volumes, and exportation) for Relic Hunters (also known as Infinity Kids).

1. Geometry Creation

- The environment concept art images – upon which the Relic Hunters characters are based – are being created by Mike Phillips and Todd Kale.
- Environments will be generated in separate sub-components, all the way down to individual wall, window, door, floor and ceiling pieces.

2. Texture-Map Creation

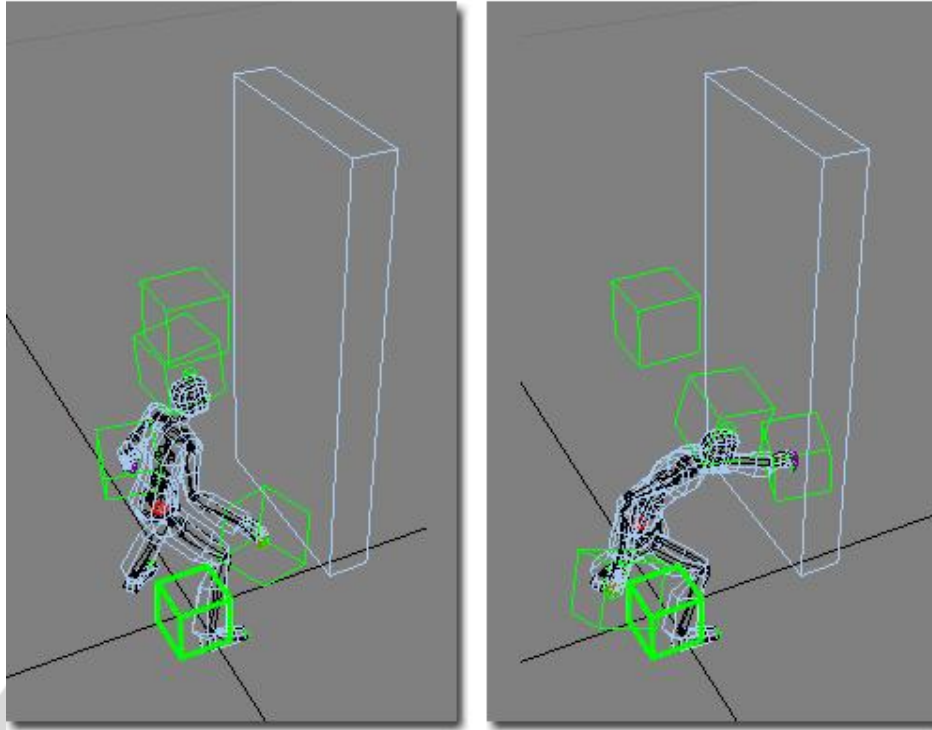
- Each *major* environment component (e.g. a library within a mansion) will receive multiple 1024x1024 SOURCE texture-maps, in .PSD format, and these files will be converted into 1024x1024 GAME READY texture-maps (whether diffuse, normal, parallax, specular, or glow), in .DDS format.
 - The resolution of the GAME READY environment texture-maps may be reduced from 1024x1024 to a smaller size *if final camera zoom extents and frame rate dictate such a change.*
- Each environment prop object (e.g. chair, plant, or television) will receive a single 1024x1024 SOURCE texture-map, in .PSD format, and that file will be converted into a 512x512 GAME READY texture-map (whether diffuse, normal, parallax, specular, or glow), in .DDS format.
 - The resolution of the GAME READY prop object texture-maps for may be reduced from 512x512 to a smaller size *if final camera zoom extents and frame rate dictate such a change.*
- We will be using the NVIDIA Photoshop plug-in to generate our .DDS-formatted texture-maps. The plug-in, as well as some useful .DDS output profiles, can be located on Topaz: [\\Server\projects\RelicHunters\Art\Photoshop_DDS_Plugins](http://Server/projects/RelicHunters/Art/Photoshop_DDS_Plugins)
 - This plug-in can *also* be used to generate normal maps.
 - While the normal map results with this plug-in have been favorable, any of the additional normal map generators, listed below, will more than suffice.
- We will be using any of the following ATI normal map generation utilities for normal map creation, and they may be downloaded from: <http://mirror.ati.com/developer/index.html>.
 - [NormalMapper](#) generates normal (bump) maps from a high and low-resolution model, for PC and the APPLE platform.
 - [NormalMapperUI](#) - 3DS Max plug-in (version 1.22).
 - [NormalMapperUI](#) - Stand alone (version 1.21).
 - These executables can also be found on Topaz here: [\\Server\projects\RelicHunters\Art\ATI_NormalMapCreation_Utilities](http://Server/projects/RelicHunters/Art/ATI_NormalMapCreation_Utilities)
- We will be using the standard Gamebryo material shader for all of our environment components, whether entire locations or individual props.

- There are texture-maps slots for diffuse, normal, gloss (specular), glow, and parallax, located within the maps panel of the Gamebryo material shader.
 - Select a texture-map for each of those map slots, and your 3D object *should* display with the desired results when exported to the Gamebryo Asset Viewer.
- The Gamebryo shader will also allow you to use the diffuse color in combination with grayscale texture-maps.
- Using the specular color in this fashion, however, in combination with the in-game lighting and shadow systems, will create poor results, according to Programming.
 - A 3D object, under these conditions, will light up like a Christmas tree in the shadows.

3. Node Creation and Functionality

- ***Please NOTE:*** The following node-based instructions may need to be altered due to the requirements of the AGEIA PhysX system, according to Programming, as of 06-25-2007.
- All 3D models within an environment, or *game level*, must be associated with a *placement node*.
- All interactive 3D models must *also* be associated with a *mount node*.
- A *placement node* (“N_”) is used as a “collection area” for one or more 3D objects.
 - If a library, for example, contained hundreds of books, ten bookshelves, six tables, and twenty chairs, *all* of those 3D objects would be linked to one placement node, **N_E_Library** (Node_Environment_Library).
- A *mount node* (“N_M_”) would be used as a “mount point” from where the player character could approach and interact with an object-of-interest.
 - If the twelfth chair out of twenty chairs within that library, for example, was *interactive*, then, *that* chair would be linked to *one* mount node, **N_M_P_E_Chair12** (Node_Mount_Prop_Environment_Chair12).
 - This mount node would, in turn, be linked to the placement node, **N_E_Library**, which is at the top of the hierarchy for the library environment.
- **While some interactive 3D objects only require one mount node, others may require up to four different mount nodes**, because the player character can approach that 3D object from four different directions.
 - ***Please NOTE:*** When generating-and-placing mount node(s) for an interactive 3D object, you *must* place the node(s) in the *near-exact* location (on the floor surface upon which the player character stands) where the associated character animations begin. SEE figure below.
- **The Y-axis of a mount node must *always* point in the direction of the 3D object for which it has been created.**
 - The mount node must always be facing the direction the player character must go in order to interact with the 3D object.

An interactive 3D object must be linked to a mount node, which must be built into the .MAX and .NIF files for *that* 3D object. That mount node is linked to the placement node. The interactive 3D object, in this example, is called P_E_Kitchen_Door_Left_VER00. The mount node is called N_M_P_E_Kitchen_Door_Left. The placement node is called N_E_Kitchen (not seen in this figure).



- If you are required to hide a specific geometry component before exportation, such as the Nav Mesh, then, you are *also* required to hide the corresponding node (where the geometry is called E_NavMesh and the node is called N_E_NavMesh).

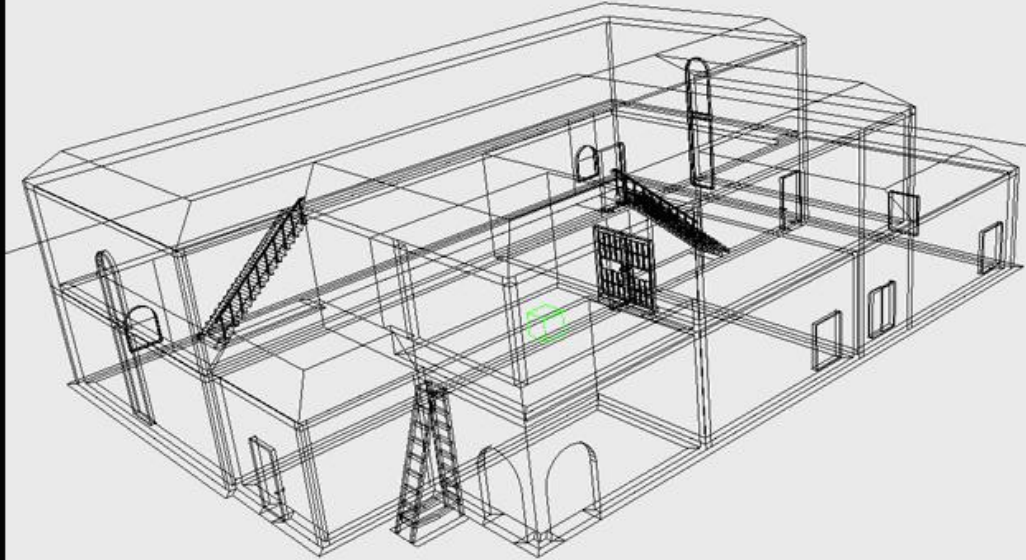
4. Creating Collision Volumes and Navigation Meshes

- Each interactive 3D object in an environment scene requires a collision volume or ABV (Alternate Bounding Volume).
- A standard environment ABV, for the Relic Hunters prototype, will involve the use of either a 3D Studio MAX “primitive” **bounding box** (for simple 3D objects) or a stripped-down version of specific environment geometry (for more complex 3D objects).
- A single 3D object ABV is to be named NDLCD BN.
- When an environment scene contains *multiple* interactive 3D objects, the associated ABVs must be named NDLCD BN00, NDLCD BN01, NDLCD BN02, etc.
- Whereas a 3D object needs to have NiOptimizeKeep applied (under User Defined Properties), the ABV for *that* 3D object will contain *no* user defined properties what-so-ever.
- Whether viewing a 3D object within 3D Studio MAX or the Gamebryo Asset Viewer, one should always expect to see the ABV linked *to* the 3D object, not the other way around.
- **Any and all walk-able surfaces will receive custom-built Navigation Meshes (or Nav Meshes)**, based loosely upon the actually aesthetically-correct geometry from the environment scene.
- Programming was originally going to allow for delivery of one large-and-welded Nav Mesh for an entire environment, but, as of 06-24-2007, if a large environment includes several different spaces or rooms, then, you must divide up

the single Nav Mesh into a series of smaller Nav Meshes (one for each unique space).

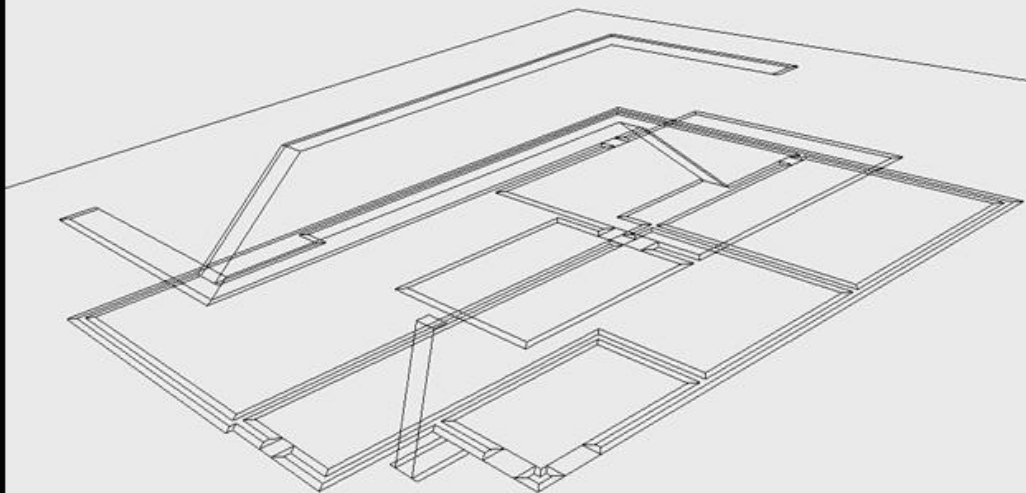
- **Each and every Nav Mesh in an environment will *also* require a *Soft Nav Mesh*, unless otherwise specified by Design.**

This is the original test environment upon which the Infinity Kids Sandbox utility was created and refined.



This environment includes simple and complex geometry, Bounding Boxes, Nav Meshes, and Nodes.

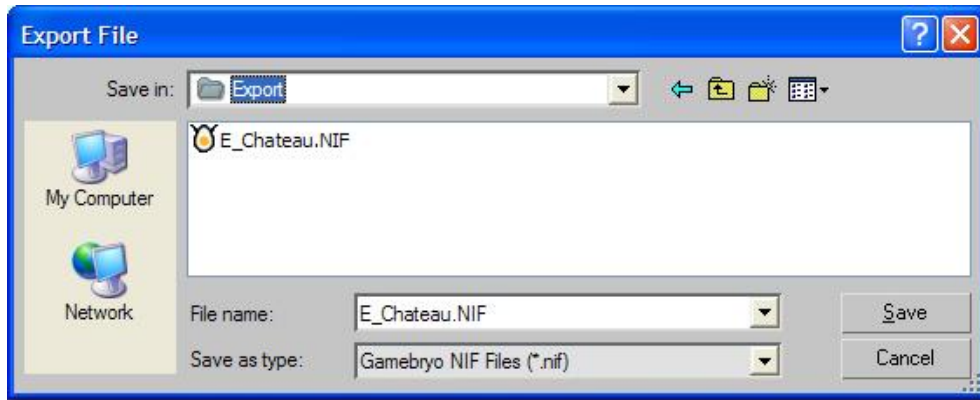
Below is "Nav Mesh" (the walk-able space of the test environment), as well as the "Nav Mesh Soft" (the .5 meter boundary between all wall surfaces and the walk-able space, which allows the character to somewhat-accurately come into contact with wall surfaces without intersecting them).



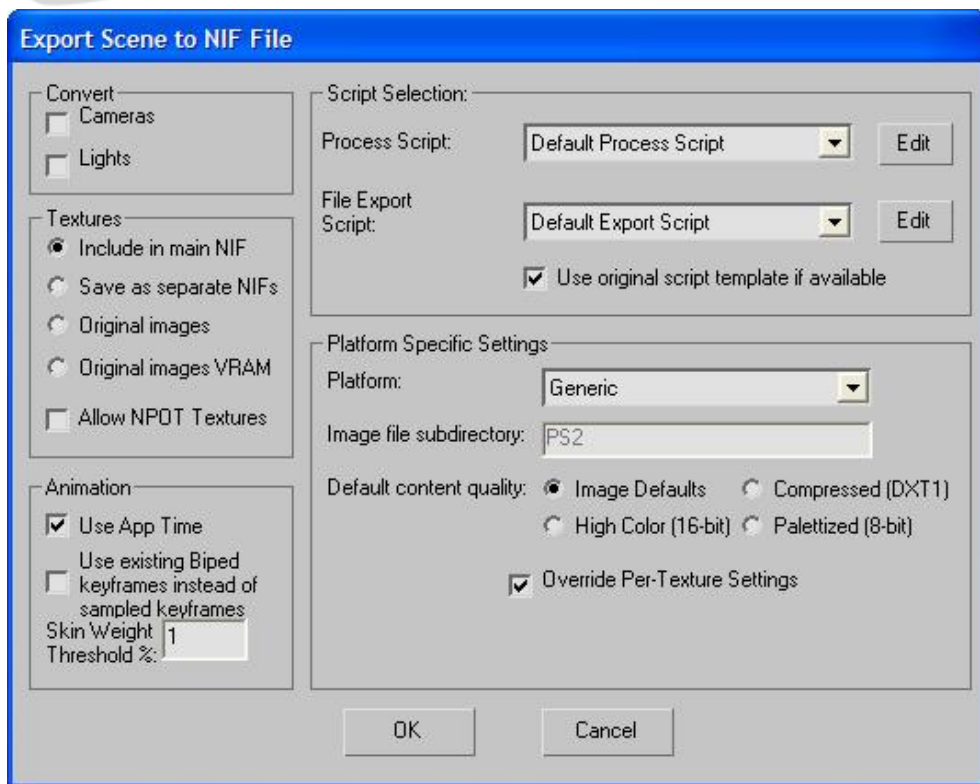
The Nav Mesh and the Soft Nav Mesh, even as separate 3D objects, must always be created so that their collective vertices always match up, as if prepared for welding.

5. Exporting the Environment Model Art Asset to Gamebryo

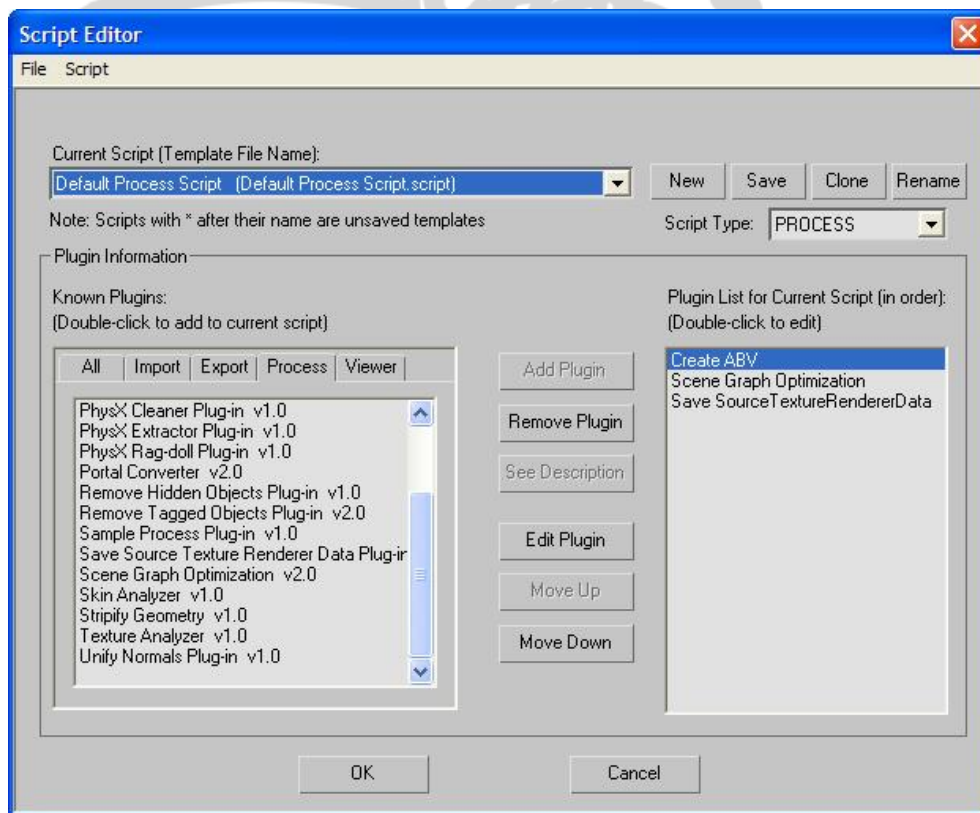
- Once your environment scene or environment prop is ready for exportation (and every element of your art asset has been error-checked), you will launch the Gamebryo **Export File** requester, either through File – Export or through the associated icon located on the Gamebryo toolbar.



- **All 3D objects, regardless of association with animation, are saved as .NIF files.** Upon naming your file and clicking “Save,” you will be met with another requester that allows for several options, only a few of which you will utilize for Relic Hunters:
 - **Include your texture-maps with the .NIF you are saving**, or, save your texture-maps as separate .NIF files along with your 3D object .NIF.
 - Programming, as of 06-24-2007, prefers that each .NIF include the associated texture-maps.

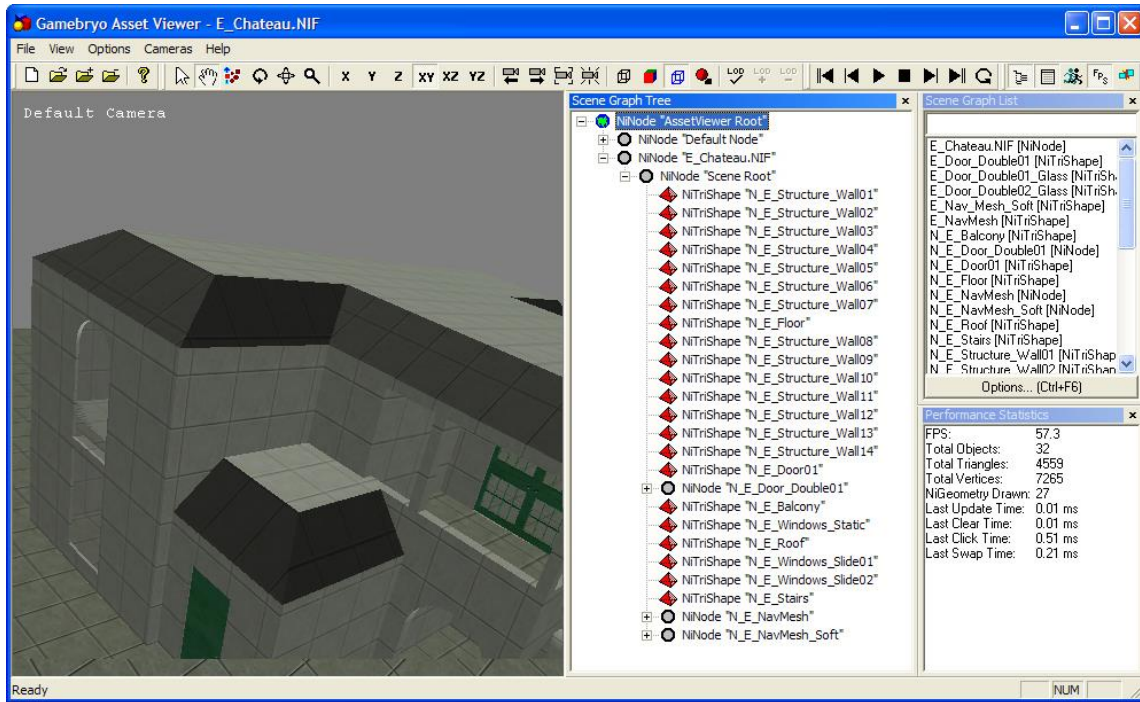


- **Edit the “Default Process Script” (SEE below).**
 - The *only* time you are required to edit the default process script is when you have installed the Gamebryo plug-in for 3D Studio MAX on your computer system for the *first* time, or, if you have had to *re-install* the plug-in.
 - The default process script operates, by default, with a specific list of plug-ins:
 - Create ABV
 - Scene Graphic Optimization
 - Save Source TextureRendererData
 - Stripify Geometry v1.0
 - We are using tri-shapes rather than tri-strips, for Relic Hunters, because tri-shapes allow us to be able to process the geometry in a less complex way, and modern hardware gets no benefit from tri-stripping, according to Programming.
 - You must remove Stripify Geometry v1.0 to allow exportation of tri-shapes.
 - Click on **Stripify Geometry v1.0** in the plug-in list, click **Remove Plugin**, and click **OK**.
 - Clicking **OK** will return you to the **Export Scene to NIF File** requester, where you will click OK, once more, which will officially export your file to your chosen directory.



6. Viewing the Environment Model Art Asset in the Gamebryo Asset Viewer

- Once you have successfully exported your environment model art asset, you can view the resultant file by double-clicking directly on the .NIF file or double-clicking on the Gamebryo Asset Viewer icon, going through File – Open, and loading the .NIF file.



- The Scene Graph Tree (SEE center panel above) allows you to view the hierarchy of all the basic components that make up your .NIF file.
 - Notice how *NITriShape* is displayed rather than *NITriStrip*.

7. Checking the Environment Model Art Asset into Perforce

- Once you have verified that your exported environment art asset is error-free, you need to check your art asset (as well as all associated SOURCE art files) into Perforce.
- **There are three Perforce directory locations, as of 06-25-2007, where you are obligated to check in your environment files.** The prototype Mansion, for example, requires that you deposit files into these locations on your local system:
 - **C:\Clients\Projects\Art\Environments\Mansion\Export**
 - Deposit all .NIF and .DDS files (*even though* the .DDS files are included with the associated .NIF files) here.
 - **C:\Clients\Projects\Art\Environments\Mansion\Source**
 - Deposit all .MAX, .PSD, and .DDS files (as the .MAX file map paths point to the .DDS files, not the .PSD files) here.
 - **C:\Clients\Projects\Testbeds\Sandbox\Data\Objects**
 - Deposit all .NIF files here.
 - Everyone on the Relic Hunters team, until further notice, is running the game from the Sandbox executable, thus, it is extremely important to make sure all of your .NIF files are up-to-date in this directory above all else.