Art Asset Directory Structures, File Formats, & Naming Conventions

Created by Eric M. Scharf

Created On: 04-12-2007 Last Modified: 06-25-2007

ALL Relic Hunters art asset files have a **SOURCE** version and an **EXPORT** version, *except for reference and concept images*. The source file is the ***original*, raw** file to which anyone can refer if the export file has been corrupted or inadvertently over-written in Perforce. The export file is the ***final, GAME READY*** file that is used directly by our engine technology. We use MAX and Maya for creating *prototype* 3d character models. We use 3d Studio MAX for generating and completing *ALL full production* 3d models and associated animations. We use Adobe Photoshop and Painter to generate our texture-maps and user interface components. We use Gamebryo as the underpinnings for our game engine technology.

The following Directory Structure Rules are true:

Each artist associated with Relic Hunters (even in a *temporary* capacity) will have a "working" directory located here: \\Server\projects\RelicHunters\Art\Working.

EricScharf
MikePhillips
ToddKale
WaltCarter

The *working* directory for each artist will include a series of sub-directories accounting for each-and-every art asset category required for the game:

YourName_Characters
YourName_Concepts
YourName_Drones
YourName_Environments
YourName_Gadgets
YourName_Props
YourName_SkyDomes
YourName_SpecialEffects
YourName_Terrain
YourName_Triggers
YourName_UI
YourName_Vegetation
YourName_Vehicles

- Each artist receives the identical working directory structure so that the Art team can quickly identify the origin of *any* art asset. *Accountability is our friend.*

- ***If an artist has generated a specific art asset, such as one of our lead characters, Basil, then, all art assets associated with Basil would need to be deposited into a newly-created-by-the-artist Basil directory that sits within YourName_Characters.***

- ***PLEASE NOTE:*** **Specific art assets have been designated to contain multiple categories of art**. A completed environment, for example, may contain buildings, props, vegetation, and static vehicles, **however, the individual art components of that environment all *originate* from their "home" directories.**
    - An environment art asset, for example, is created and deposited into **YourName_Environments**, and a super computer art asset is created and deposited into the **YourName_Props**.
        - The super computer art asset is, then, merged into and properly positioned within the environment art asset, as a static prop. Anyone can still access these *individual* components without having to sift through a *hybrid* environment file.
        - On the other hand, once an artist has merged the prop object into the larger environment file, then, that artist is responsible for copying all of the prop object-associated texture-map .DDS files into the environment directory as well.
        - Again, no Relic Hunters Art team member should ever have to search blindly for any art files.

***As is standard "source control" protocol*, approved and completed art assets, SOURCE and GAME READY, will be duplicated from \\Server\projects\RelicHunters\Art\Working and deposited into the associated art directories within Perforce: \\Clients\RelicHunters\Art.**

Within this Perforce directory, an artist will find a similar set of directories designed to contain each major art asset category available in the game:

- Characters
- ConceptArt
    - Characters
    - Drones
    - Environments
    - Gadgets
    - Props
    - Puzzles
    - Storyboards
    - Structures (This art asset type is a hybrid; too large to be a prop but too small to be an environment)
    - Vehicles
- Drones
- Environments
- Gadgets
- Props
- **Sandbox** (This directory, during Prototype and Full Production project phases, will be off-limits)
- SkyDomes
- SpecialEffects
- Terrain
- Triggers
- UI
- Vegetation
- Vehicles

Approved and completed GAME READY art assets will *also* be duplicated from \\Topaz\projects\InfinityKids\Art\Working and deposited into the associated *game data* directory within Perforce: //RelicHunters/Game/Data.

- There will be no sub-directories within the Data directory.
- No raw, SOURCE files of any kind are allowed in the Data directory.
    - The game engine will choke if any file type other than GAME READY makes its way into the Data directory.

The following File Format and Naming Convention Rules are true:

01 – *ALL* reference image names will begin with "**RH_"** (RelicHunters_).

02 – *ALL* reference images should be saved in or converted to .JPG file format.

03 – *ALL* concept image names should begin with **"Concept_"**.

04 – *ALL* concept image iterations are referred to as "Stages".

- A properly named set of iterative concept images would be:
    - Concept_Mansion_Stage01.JPG.
    - Concept_Mansion_Stage02.JPG.
    - Concept_Mansion_Stage03.JPG.
- ***A concept stage consisting of multiple images*** (for displaying different profiles of the same character, for example) would be:
    - Concept_Mansion_Stage01_01.JPG.
    - Concept_Mansion_Stage01_02.JPG.
    - Concept_Mansion_Stage01_03.JPG.
- **Any concept image that is a one-off would be displayed as:**
    - **Concept_Mansion.JPG.**

05 – *ALL* concept images should be saved in two file formats (SOURCE and PUBLIC):

- SOURCE is unavailable to anyone outside of the Art team (unless by special request or *an absolute emergency*), will be in .PSD format.
- PUBLIC is available to anyone and everyone outside of the Art team, will be in .JPG format.

06 – SOURCE 3D models, regardless of subject matter, are saved in .MAX file format.

- ***Please NOTE:*** Character 3D models present a special case, as, with rare exception, they will begin as Maya files, and, then, be converted into .MAX files. So, Maya files are PRELIMINARY and .MAX files are SOURCE.

07 – All SOURCE 3D character animations are saved in both .MAX and .BIP file format.

- Saving 3D character animations in .BIP file format allows for quick-and-easy application of one animation, for a particular character, onto another completely different character.

08 – **All SOURCE texture-maps are saved in .PSD file format,** because source texture-maps are meant to be flexible, multi-layered, and uncompressed.

09 – All EXPORT file formats are .NIF (Static-Rigged and Static-Non-Rigged Geometry), .KF (Single Animation) files, and .KFM (List of required Animations).

- Texture-maps are always exported as *separate* .NIF files.
- ***Please NOTE:*** Special Effects and Ambient Environment animations are the only animated art assets which do *not* require .KF or .KFM files.
    - Animations associated with these art asset types can be exported directly as animated .NIF files.
- ***Please NOTE:*** Drone or Vehicle animations can potentially require multiple animations, and, thus, would require .KF *and* .KFM files.

10 – GAME READY art assets are defined as complete, thoroughly tested for errors, and ready to be checked into Perforce for inclusion in the very latest game build.

11 – All GAME READY texture-maps are saved in .DDS file format, which are exported as .NIF files.

- The .DDS file format comes in a variety of flavors:
  - DXT1 (Standard RGB format with NO Alpha Channel)
  - DXT1 (Standard RGB format with 1-Bit Alpha Channel)
  - DXT3 (Standard RGB format with 4-Bit Alpha Channel)
  - DXT5 (Standard RGB format with 8-Bit Alpha Channel)

12 – While all art assets will be shared whenever possible, it will *not* be necessary to create a *shared* naming identifier for any shared files.

13 – All texture-map file names, SOURCE and GAMEREADY, follow this pattern:

- The texture-map file names begin with "TEX_".
  - Followed by the name of the art asset for which the texture-map has been generated.
  - Followed by the type of texture-map. Some art assets require only the standard "diffuse" texture-maps and other art assets may require diffuse, normal, specular, glow, and parallax texture-maps.
    - Diffuse texture-maps are specified by "_DIF".
    - Normal texture-maps are specified by "_NML".
    - Diffuse Decal texture-maps are specified by "_DCD".
    - Normal Decal texture-maps are specified by "_DCN".
    - Specular texture-maps are specified by "_SPC".
    - Glow texture-maps are specified by "_GLW".
    - Parallax texture-maps are specified by "_PLX".
  - Followed by the appropriate file extension.
    - SOURCE texture-maps are specified by ".PSD".
    - GAME READY texture-maps are specified by ".DDS".

<u>Texture-Map File Format Example for Characters –</u>

Art Format: TEXTURE_CHARACTER_GENDER_NAME_VERSION##_DIF.PSD
Translation: TEXTURE_CHARACTER_MALE_BASIL_VERSION00_DIFFUSE.PSD
Source: TEX_C_M_Basil_VER00_DIF.PSD
Game Ready: TEX_C_M_Basil_VER00_DIF.DDS

<u>Texture-Map File Format Example for Drones –</u>

Art Format: TEXTURE_DRONE_NAME_VERSION##_DIF.PSD
Translation: TEXTURE_DRONE_SCORPIO_VERSION00_DIFFUSE.PSD
Source: TEX_D_Scorpio_VER00_DIF.PSD
Game Ready: TEX_D_Scorpio_VER00_DIF.DDS

<u>Texture-Map File Format Example for Gadgets –</u>

Art Format: TEXTURE_GADGET_NAME_VERSION##_DIF.PSD
Translation: TEXTURE_GADGET_BOUNCEBOOTS_VERSION00_DIFFUSE.PSD
Source: TEX_G_BounceBoots_VER00_DIF.PSD
Game Ready: TEX_G_BounceBoots_VER00_DIF.DDS

<u>Texture-Map File Format Example for Character Props –</u>

Art Format: TEXTURE_PROP_TYPE_NAME_VERSION##_DIF.PSD
Translation: TEXTURE_PROP_CHARACTER_STATUE_VERSION00_DIFFUSE.PSD
Source: TEX_P_C_Statue_VER00_DIF.PSD
Game Ready: TEX_P_C_Statue_VER00_DIF.DDS

Texture-Map File Format Example for Environment Props –
Art Format: TEXTURE_PROP_TYPE_NAME_VERSION##_DIF.PSD
Translation: TEXTURE_PROP_ENVIRONMENT_STATUE_VERSION00_DIFFUSE.PSD
Source: TEX_P_E_Statue_VER00_DIF.PSD
Game Ready: TEX_P_E_Statue_VER00_DIF.DDS

Texture-Map File Format Example for Sky Domes –
Art Format: TEXTURE_SKYDOME_NAME_VERSION##_DIF.PSD
Translation: TEXTURE_SKYDOME_NOON_VERSION00_DIFFUSE.PSD
Source: TEX_SKY_Noon_VER00_DIF.PSD
Game Ready: TEX_SKY_Noon_VER00_DIF.DDS

Texture-Map File Format Example for Special Effects –
Art Format: TEXTURE_SPECIALEFFECT_NAME_VERSION##_DIF.PSD
Translation: TEXTURE_SPECIALEFFECT_DUST_VERSION00_DIFFUSE.PSD
Source: TEX_SPFX_Dust_VER00_DIF.PSD
Game Ready: TEX_SPFX_Dust_VER00_DIF.DDS

Texture-Map File Format Example for Terrain –
Art Format: TEXTURE_TERRAIN_TYPE_NAME_VERSION##_DIF.PSD
Translation: TEXTURE_TERRAIN_MOUNTAIN_ROAD_VERSION00_DIFFUSE.PSD
Source: TEX_T_Mountain_Road_VER00_DIF.PSD
Game Ready: TEX_T_Mountain_Road_VER00_DIF.DDS

Texture-Map File Format Example for Vegetation –
Art Format: TEXTURE_VEGETATION_TYPE_NAME_VERSION##_DIF.PSD
Translation: TEXTURE_VEGETATION_TREE_PALM_VERSION00_DIFFUSE.PSD
Source: TEX_VEG_Tree_Palm_VER00_DIF.PSD
Game Ready: TEX_VEG_Tree_Palm_VER00_DIF.DDS

Texture-Map File Format Example for Vehicles –
Art Format: TEXTURE_VEHICLE_TYPE_VERSION##_DIF.PSD
Translation: TEXTURE_VEHICLE_PORSCHE_VERSION00_DIFFUSE.PSD
Source: TEX_VEH_Porsche_VER00_DIF.PSD
Game Ready: TEX_VEH_Porsche_VER00_DIF.DDS

14 – All 3D model file names, SOURCE and GAMEREADY, follow this pattern:

- The 3D file names begin with an ***identifier*** for the type of 3D art asset.
  - Identifiers are listed at the bottom of this document, in the identifier glossary (e.g. C = Character, E = Environment, and P = Prop)
  - **An identifier would be followed by the name of the art asset for which the 3D model has been generated.**
    - The name of the art asset varies from 3D model to 3D model and can include details such as gender for characters.
  - Followed by the appropriate file extension.
    - **SOURCE 3D models are specified by:**
      - ".MAX" – 3D Studio MAX file format
      - ".MA" – Maya ASCII file format
      - ".MB" – Maya Binary file format
        - ***Please NOTE:*** Even though Maya file formats will *only* be associated with SOURCE 3D character models, some artists prefer ASCII output to Binary output when working in Maya.
          - **Maya Binary is *not* intended to be read or modified.** It contains some Binary and some text.

- **Maya ASCII *can be edited*,** but it is not in a transparent form. The data is used as input to Maya functions, so geometrical details like vertex and triangle lists are not available.

- **GAME READY 3D models are specified by (Also SEE rule 9):**
  - ".NIF" – Static-Rigged and Static-Non-Rigged Geometry
  - ".KF" – Single Animation
  - ".KFM" – List of Required Animations

<u>3D File (Models and Animations) Format Example for Characters –</u>

Art Format: IDENTIFIER_GENDER_NAME_VERSION##.MAX
IDENTIFIER_GENDER_NAME_VERSION##.MA
IDENTIFIER_GENDER_NAME_VERSION##.MB
IDENTIFIER01_IDENTIFIER02_GENDER_ANIMATION_VERSION##.MAX

Translation: CHARACTER_MALE_BASIL_VERSION00.MAX
CHARACTER_MALE_BASIL_VERSION00.MA
CHARACTER_MALE_BASIL_VERSION00.MB
ANIMATION_CHARACTER_MALE_WALK_VERSION00.MAX

- **For animation files, VERSION will refer to the different types of a particular animation:**
  - Walk_Standard, Walk_Athletic, and Walk_Heavy, from an animation description list, will display as:
    - Walk_VER00, Walk_VER01, and Walk_VER02
- Any animation files that do *not* exist as more than one type will continue to be referred to as VER00.

Source: C_M_Basil_VER00.MAX
C_M_Basil_VER00.MA
C_M_Basil_VER00.MB
A_C_M_Walk_VER00.MAX

Game Ready: C_M_Basil_VER00.NIF
A_C_M_Walk_VER00.KF
A_C_M_Basil_VER00.KFM

<u>3D File Format Example for Environments –</u>

Art Format: IDENTIFIER_NAME_VERSION##.MAX
Translation: ENVIRONMENT_MANSION_VERSION00.MAX
Source: E_Mansion_VER00.MAX
Game Ready: E_Mansion_VER00.NIF

<u>3D File Format Example for Drones –</u>

Art Format: IDENTIFIER_NAME_VERSION##.MAX
IDENTIFIER_NAME_VERSION##.MA
IDENTIFIER_NAME_VERSION##.MB
IDENTIFIER01_IDENTIFIER02_NAME_ANIMATION_VERSION##.MAX

Translation: DRONE_SCORPIO_VERSION00.MAX
DRONE_SCORPIO_VERSION00.MA
DRONE_SCORPIO_VERSION00.MB
ANIMATION_DRONE_SCORPIO_CRAWL_VERSION##.MAX

Source: D_Scorpio_VER00.MAX
D_Scorpio_VER00.MA
D_Scorpio_VER00.MB
A_D_Scorpio_Crawl_VER00.MAX

Game Ready: D_Scorpio_VER00.NIF
A_D_Scorpio_Crawl_VER00.KF
A_D_Scorpio_VER00.KFM

3D File Format Example for Gadgets –

Art Format: IDENTIFIER_NAME_VERSION##.MAX
Translation: GADGET_BOUNCEBOOTS_VERSION00.MAX
Source: G_BounceBoots_VER00.MAX
Game Ready: G_BounceBoots_VER00.NIF

3D File Format Example for Character Props –

Art Format: IDENTIFIER_TYPE_NAME_VERSION##.MAX
Translation: PROP_CHARACTER_BOOMERANG_VERSION00.MAX
Source: P_C_Boomerang_VER00.MAX
Game Ready: P_C_Boomerang_VER00.NIF

3D File Format Example for Environment Props –

Art Format: IDENTIFIER_TYPE_NAME_VERSION##.MAX
Translation: PROP_ENVIRONMENT_STATUE_VERSION00.MAX
Source: P_E_Statue_VER00.MAX
Game Ready: P_E_Statue_VER00.NIF

3D File Format Example for Sky Domes –

Art Format: IDENTIFIER_NAME_VERSION##.MAX
Translation: SKYDOME_NOON_VERSION00.MAX
Source: SKY_Noon_VER00.MAX
Game Ready: SKY_Noon_VER00.NIF

3D File Format Example for Special Effects –

Art Format: IDENTIFIER_NAME_VERSION##.MAX
Translation: SPECIALEFFECT_DUST_VERSION00.MAX
Source: SPFX_Dust_VER00.MAX
Game Ready: SPFX_Dust_VER00.NIF

3D File Format Example for Terrain –

Art Format: IDENTIFIER_TYPE_NAME_VERSION##.MAX
Translation: TERRAIN_MOUNTAIN_ROAD_VERSION00.MAX
Source: T_Mountain_Road_VER00.MAX
Game Ready: T_Mountain_Road_VER00.NIF

3D File Format Example for Vegetation –

Art Format: IDENTIFIER_VEGETATION_TYPE_NAME_VERSION##.MAX
Translation: VEGETATION_TREE_PALM_VERSION00.MAX
Source: VEG_Tree_Palm_VER00.MAX
Game Ready: VEG_Tree_Palm_VER00.NIF

3D File Format Example for Vehicles –

Art Format: IDENTIFIER_TYPE_VERSION##.MAX
IDENTIFIER01_IDENTIFIER02_TYPE_ANIMATION_VERSION##.MAX
Translation: VEHICLE_PORSCHE_VERSION00.MAX
ANIMATION_VEHICLE_PORSCHE_SUSPENSIONBOUNCE_VERSION00.MAX
Source: VEH_Porsche_VER00.MAX
A_VEH_Porsche_SuspensionBounce_VER00.MAX
Game Ready: VEH_Porsche_VER00.NIF
A_VEH_Porsche_SuspensionBounce_VER00.NIF

15 – All user interface file names will begin with "UI_".

- Followed by the category of UI art asset (e.g. Front End or In-Game).
- Followed by the type of that category of UI art asset (e.g. Options).

- Followed by the subject of that type of that category of UI art asset (e.g. Audio).
- Followed by the state (normal, mouse-over, selected) of that subject of that type of that category of UI art asset (e.g. selected).

Source Art Format:	UI_CATEGORY_TYPE_SUBJECT_STATE.PSD
Translation:	UI_FrontEnd_Options_Audio_Selected.PSD
Final Art Result:	UI_FrontEnd_Options_Audio_Selected.DDS

- ***PLEASE NOTE:*** All individual user interface art asset files will be programmatically compiled into one massive texture-page, or, one set of massive texture-pages.

16 – All 3D model and texture-map art assets will include a VERSION number, and each version number will begin with "00".

- Version numbers allow us to guard against the sudden requirement of *additional versions* of art assets, thus protecting the naming convention. There are no exceptions.
  - A specific 3D art asset, such as a mansion, for example, may require multiple texture-maps. The mansion may be version 00, thus, a texture-map naming convention for this mansion would include an additional "_##" before the file extension:
    - TEX_E_Mansion_VER00_00.DDS
    - TEX_E_Mansion_VER00_01.DDS
    - TEX_E_Mansion_VER00_02.DDS
      - In other games, "_00, _01, and _02" might refer to "_Leather, _Stucco, and _Wood". This is *not* the case with Relic Hunters.
  - If a specific 3D art asset only requires one texture-map version, then, for example, the name will simply read as TEX_E_Mansion_VER00.DDS

17 – 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_VER00** (Node_Mount_Prop_Environment_Chair12_Version00).
    - 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.

18 – All character models *must* be associated with a specific, unchanging set of nodes. SEE InfinityKids_Art_Pipeline_Characters.doc for further details. These nodes do not follow the "N_" naming convention, as they are part of an enhancement of the character rig originally utilized for Arabian Lords:

- ACCUM_NODE
- CAMERA_NODE
- HAT_NODE
- HANGING_PULL_NODE

- LH_ATTACH_NODE
- RH_ATTACH_NODE
- TORSO_PULL_NODE
- UI_NODE
- WAIST_PULL_NODE

19 – Relic Hunters Identifier Glossary:

- **A = Animation**
- **ABV = Alternate Bounding Volume**
- **BB = Bounding Box**
- **C = Character**
- **Concept = Concept Art**
- **D = Drone**
- **E = Environment**
- **G = Gadget**
- **RH = Relic Hunters**
- **LOD = Level of Detail (Progressive Downgrades to Geometry)**
- **N = Node**
- **M = Mount**
- **P = Prop (Non-Specific, Stand-Alone, 3D Object of Any Kind)**
- **SKY = Sky Box / Dome**
- **SPFX = Special Effects**
- **Stage = One of Several Versions of a Piece of Concept Art**
- **STR = Structure**
- **T = Terrain**
- **VEG = Vegetation**
- **VEH = Vehicle**
- **VER = Version**