**NIM Files**

NIM is designed to be a flexible application to streamline file I/O for major CG applications. It consists of a suite of files that store data, and can be used to easily save, publish and version up files, navigate complicated project directory hierarchies, and helps to autmoate tasks inside of specific CG applications.

**General NIM Files :**

The NIM module is made up of the following, central files that work independent of whatever CG application you are in :

1. nim.py

This constructs a central NIM dictionary, containing all of the information needed to build a NIM GUI, and information relevant to saving, versioning up, publishing and importing documents. It instantiates a NIM class object, and allows you to set information such as the name a combo box is set to, the corresponding NIM ID number, dictionaries, server paths, and can be used to read and store preferences and scene variables in a NIM dictionary, storing the name of the current application, user information, etc. The "set\_dictionary" function is a convenient call for populating an element's dictionary, based on the information contained in the current NIM dictionary. Valid elements are "job", "asset", "show", "shot", "filter", "task", "base" (basename), and "ver" (version). The setting of the Asset/Show&Shot tab, is referred to as "tab".

2. nim\_api.py

Contains several wrapper functions to make calls to the NIM API more pythonic. Contains wrappers that are called to populate NIM dictionaries with information for the various elements. It also contains a few functions that construct directories, basenames, filenames and file paths, when passed the current NIM dictionary. Other important functions of note include the "versionUp" command, which gets called any time a file is saved, version'ed up, or published. nim\_api.versionUp() has an important call to nim\_file.verUp() inside it, which performs the actual file save operation, along with building many of the file directories and names. Once this has run, nim\_api.add\_file() is called, to add the new file information to the NIM API.

3. nim\_file.py

Contains several functions related to file operations. You can query the user, application and the list of supported applications. You can also query the current application scene file path, swap platform specific paths, and reload the scripts inside of any supported application. Most importantly, is the "verUp()" command, which is run to version up a scene file in any NIM supported application. This will also set and get variables from the supported scene file, make calls to construct the Maya project, set render and comp directories, etc.

4. nim\_prefs.py

General file for dealing with NIM preferences. Includes helper functions to construct the NIM home directory, prompt the user to input a NIM URL, verifies the URL, verifies that all required preference attributes are present, and builds the default preferences. Also includes a function to read preferences into a dictionary, and update preferences with new information. Also includes a function that will turn debug mode on or off, which results in more verbose information being printed.

5. nim\_print.py

A simple file for printing information. "info" is used to print information normally. "debug" will print only when the debug option in the preferences file is turned on. "warning" should be used to return non-fatal warnings, and "error" should be used to print fatal errors. "log" is designed to be used to print to a log file, in future versions (not yet implimented).

6. nim\_tools.py

A generic file for holding various tools. Currently, the main function in here is one used to construct a dialog window to get a comment from the user (This can be moved over to nim\_win.py, in the future).

7. nim\_win.py

This is designed to be a general, all purpose window constructor, which should build simple dialog windows to confirm ("OK" button), give a choice ("Yes"/"No"), or ask for a string input. There is also a window to allow the user to set their username to be a valid NIM username, which also updates the preferences file.

8. UI.py

This is the main file that constructs a NIM GUI inside of any application that supports PySide or PyQt. Currently, this only includes Maya and Nuke, but in the future, it can easily be expanded to work inside of any PySide/PyQt environment. It contains application specific calls to open, save, import and references files, and calls external files to get and set variables inside any supported application, as well as any other application specific calls.

**Application Specific Files :**

Additionally, the following files are application specific :

1. nim\_maya.py

Contains functions to get the name of the main Maya window (to parent the NIM and related windows under), as well as calls to get and set NIM variables inside of Maya (stored in custom NIM attributes on the defaultRenderGlobals node), and also contains functions to create the workspace file, and build and set the Maya Project. Pub\_Chex is the custom NIM Maya publish checks, that is still under construction.

2. nim\_nuke.py

Contains functions to get and set variables inside of Nuke (stored on the Project Settings node). it can also be used to create a custom NIM node (currently supports Read and Write nodes). The Win\_SavePySide function can be used to prompt the user if they would like to save, version up or not save the current scene file, before opening a new one.

3. nim\_c4d.py

Contains the functions to get and set variables inside of C4D (stored in document containers), as well as the main code for the NIM C4D GUI. The "nim\_fileUI" class builds the NIM GUI, and functionally, should be pretty much the same as the regular Maya/Nuke PySide/PyQt GUI.

4. nim\_v5-4-r04.pyp (C4D plugin file)

This is the C4D plugin file that is used to register the NIM plugins, and creates the custom NIM menu. The whole nim\_v5-4-r04 plugin folder should be dropped in the user's C4D plugins directory.

5. nim4d\_userWin.py

Creates a custom C4D GUI used for setting the current username to a valid NIM username.