Using a MEL/Python Script to Load Plug-Ins

You may come across a time where you want to tell Maya to load/unload a plug-in during "plug-in load time" (i.e. while Maya is launching) or maybe you have certain plug-ins that conflict with each other, and you would like to have an automatic way of unloading (or "not" loading) various plug-ins.

To go about doing this there are four options, I highly recommend option 1, as it is the most perdictable and safest.

Option 1: Using Maya API

Within the class MSceneMessage there are callbacks for monitoring plug-in loading:

1. kBeforePluginLoad
2. kAfterPluginLoad
3. kBeforePluginUnload
4. kAfterPluginUnload

Whether you use the method addCallback which just allows you to add additional functionality after one of the above actions or you use the method addCheckCallback which allows you to abort the above actions all together and have Maya do something different (like load another plug-in).

There is a pluginCallbacks example in devkit\plug-ins

Option 2: Modifying userPrefs.mel

Maya determines which plug-ins should be loaded or not from the userPrefs.mel file (found in the folder: /username/maya/x.x/prefs/). The lines that do this are at the bottom of the file. They go something like: evalDeferred("autoLoadPlugin(.....); So a simple but messy solution is to comment out the lines which load plug-ins you don't want loaded, in the userPrefs.mel file.

Option 3: Modifying initialPluginLoad.mel

If the first solution is too messy and is not automated (i.e. you have to re-edit the userPrefs.mel file each time), another possible but solution is to edit the file: /scripts/startup/initialPluginLoad.mel and add something like this at the end of the initialPluginLoad function: if (`exists userPluginLoad`) eval "source userPluginLoad"; Then put userPluginLoad.mel somewhere on your script path, and this script file should contain something like:

```
if (`about -batch`)
{
   catch(`loadPlugin "batchPlugin1"`);
```

```
    catch(`loadPlugin "batchPlugin2"`);
    // ...
}
else
{
    catch(`loadPlugin "interactivePlugin1"`);
    catch(`loadPlugin "interactivePlugin2"`);
    // ...
}
```

By doing this, please note the following:

- This bypasses the standard autoload feature of plug-ins. If you set a plug-in to autoload, it will be loaded before Maya gets to userPluginLoad.mel.
- These files normally should NOT be edited, and great caution must be taken if you decide to proceed in doing so.
- The userPluginLoad.mel script should just be MEL code, not a MEL procedure. The above example assumes that sourcing the script will do all the work.

Option 4: Modifying autoLoadPlugin.mel

Plug-ins that are set to autoload are loaded by the MEL script "autoLoadPlugin.mel". This script is actually invoked directly from C++ code or by the user's preference file, so there is no easy way that you can avoid having this script invoked... BUT you can change what it does. If you rewrite autoLoadPlugin.mel to simply ignore some plug-ins if Maya is in batch mode, then you may be able to do what you want. The other alternative is to modify the user's preferences (userPrefs.mel) to remove the offending autoLoadPlugin command, as indicated in the first solution. This is more complicated because you have to restore the user's preferences again, after you are done with the batch mode. It may not be too bad, if all you've got to do is use the `system` MEL command to invoke OS shell command (e.g. cp, mv, etc...) to create a "backup" of the original prefs to restore in the end.