Environment Variables in Katana

Installation Path Variables

You will need to set a few environment variables to get RfK up and running with PRMan and Katana. RfK looks for RenderMan on the $RMANTREE environment variable. As long as $RMANTREE is set correctly, RfK will find the appropriate PRMan libraries and commands. Katana finds RfK in the same way it would find any other plugin - through the $KATANA_RESOURCES environment variable. The paths for these environment variables will depend on whether you install RfK/PRMan in the default locations or not.

Environment Variable	Default Installation Paths
RMANTREE	`/opt/pixar/RenderManProServer-21.2`
KATANA_RESOURCES	`/opt/pixar/RenderManForKatana-21.2-katana2.2/plugins/Resources/PRMan21`

Default shader search paths for RenderMan match the default paths in the rendermn.ini file. If additional search paths are needed then you will need to set up the `RMAN_SHADERPATH` and/or `RMAN_RIXPLUGINPATH` environment variables to include your specific paths.

Output Path Variable

You can direct the render log output to another location if necessary using the following environment variable: `RFK_REDIRECT_OUTPUT`

Shader Discovery

During startup RenderMan for Katana will automatically load all "discoverable" shaders. Shaders are discoverable if they are found in a search path and (for non-OSL shaders) have an associated Args file. The standard shader search path mechanism is used when searching for shaders with search paths specified either with PrmanGlobalStatements settings or environment variables:

OSL	`RMAN_SHADERPATH`	`options.searchpath.shader`	`$RMANTREE/lib/shaders`
Plugin (C++)	`RMAN_RIXPLUGINPATH`	`options.searchpath.rixplugin`	`$RMANTREE/lib/plugins`

If both the environment variable and the attribute are set the resulting search path will be the union of the two strings.

Args Files

The args files need to be in an Args directory. See the setup in RMANTREE/lib/plugins for an example. Something like this:

RMAN_RIXPLUGINPATH directory:

_TAF_pattern_remap.so
_ Args/
_TAF_pattern_remap.args

You can find out more about Args files in the developer docs download.