SItoA Paths

Search Paths

In the rendering options, you can set the paths where Arnold will look for shaders, procedurals, and textures.

For shaders and procedurals, they are automatically loaded if contained into the directory pointed by the ARNOLD_PLUGIN_PATH directory.

Also, you can use the following environment variables that will be read and set as your defaults when creating a new scene:

- `SITOA_SHADERS_PATH`: for shaders

- `SITOA_PROCEDURALS_PATH`: for procedurals

The paths can be changed at any time from the Arnold Render Options. If doing so, the following rule applies:

> Except for ARNOLD_PLUGIN_PATH, which is Arnold-native, all and only the paths in the search paths are used for searching the shaders (procedurals). If the field is void, the SItoA bin path is used.

If you don't need to link to shaders other than the ones coming with SItoA, leave the field void. So, when loading a scene saved with a previous SItoA version, the shaders of your current SItoA version will be loaded.

The rule also means that if you set the shader search path to, say, a workgroup location only, none of the Softimage legacy shaders will be found. So, if you need to have your workgroup's shaders used, be sure to add also the SItoA bin directory to the search path, separated by a ';' character. The order does not matter.

Example: I have some custom shaders in a workgroup located under:

`C:\dev\Arnold\WG`

and my SItoA addon installed under:

`C:\Users\stefano\Autodesk\Softimage_2010_SP1\Addons\SItoA`

Then, to have everything working, I have to set:

`Shaders Search Path = C:\Users\stefano\Autodesk\Softimage_2010_SP1\Addons\SItoA\Application\Plugins\bin\`

Note that the full path to the shader DLLs is required.

Environment Variables

Environment variables are supported for the search paths, through the [ENV_VAR] syntax.

Let's reuse the above example, and say that in my setenv.bat, I set two env vars:

`set MY_WG_BIN=C:\dev\Arnold\WG\Application\bin\nt-x86-64`

`set MY_SITOA_ADDON=C:\Users\stefano\Autodesk\Softimage_2010_SP1\Addons\SItoA`

Then, I can simply set:

`Shaders Search Path = [MY_SITOA_ADDON]\Application\Plugins\bin\nt-x86-64;[MY_WG_BIN]`

An environment variable must point to a single location. Something like:

`set MY_PATHS = C:\dev\Arnold\WG\Application\bin\nt-x86-64;C:\Users\stefano\Autodesk\Softimage_2010_SP1\A`

will **not** work.

Absolute vs Relative paths

If Use Absolute Texture Paths (From the ASS Archives tab of the rendering options) is enabled, the absolute paths is exported and optionally translated by the linktab file if Translate Paths is enabled.
Otherwise, the relative paths, with respect to Textures Search Path, is exported.

Say we have to export a texture whose full path is `C:\img\head\eye.jpg`

If you set Textures Search Path `to "C:\img",` then the texture is exported as `"head\eye.jpg",`

If set to `"C:\temp",` the texture is exported as `"..\img\head\eye.jpg",`

and so on.

In order to have the conversion working, the Textures Search Path must exist.

Example: Textures Search Path == `C:\invalid` (where C:\invalid does not exist on disk)

then, a warning message is fired, and we export the plain file name, so

Exported path == `eye.jpg`

Textures Search Path can be made of more that one path, separated by ';'

Example: Textures Search Path == `C:\img\head;C:\temp`

For multiple search paths, we convert to relative paths with respect to the first of the search paths for which the conversion was successful.

So, if we have 2 search paths, we always try to convert with respect to the first one. If failing, we try the second one, and so on. In the case of failure (none of the search paths worked), we export the plain file name and we log a warning.

This information also applies to procedural search paths, with respect to the Absolute Procedural Path flag.

Cross-platform Path Translation

For cross-platform path translations we need to specify an environment variable:

•`SITOA_LINKTAB_LOCATION`

When this variable is declared, the "Translate Paths" parameter in the Arnold Render Options will be enabled. This variable must point to the Softimage linktab.ini file with an absolute path including the filename.

Defining the linktab File

The linktab file is used when the render network uses a mix of Windows and other operating systems. Each line in a linktab file contains a Windows-style path and a UNIX-style path, indicating where Softimage, and resources such as textures, are located on both operating systems. This allows the rendering master to find the required files on the slaves regardless of the operating system.

Most linktab files contain only one line, indicating where Softimage is located on both platforms. Here is a sample linktab file:

`C:\Softimage      /usr/Softimage`

•The Windows path must come before the Unix-style path and they must be separated by a tab, not spaces.

•Linux path names are case-sensitive, but Windows paths are not (they are case-insensitive).

•Do not use slashes (\ or /) at the end of the linktab paths.

•If you are using textures or memory-mapped images, you must have entries that point to the directories containing them.

•You can use an exclamation mark to distinguish a mounted volume from the rest of the path. For example, if \\foobar\users\fred (Windows) is equivalent to /home/fred (Linux) and F:\foobar\users is a mounted volume, the line in the linktab file would look like this:

```
\\foobar\users!fred        /home/fred
```

The linktab file must be present on the master and slave computers, in the directory specified by the **SI_LINKTAB_LOCATION** environment variable.

• `SI_LINKTAB_LOCATION` can define both the path and file name or just the path; if no file name is explicitly set, Softimage assumes the name will be `linktab.ini`.

• When a scene is imported from SOFTIMAGE|3D into Autodesk Softimage, Softimage searches in SI_LOCATION for `linktab.ini`. Define `SI_LOCATION` in `setenv.bat` (Windows) or `.xsi_5.0` (Linux) as the directory where Softimage is installed and place a copy of `linktab.ini` there (if you want to reuse the linktab file used by SOFTIMAGE|3D; otherwise, just make `SI_LOCATION` point to the same path and linktab name as defined by `SI_LINKTAB_LOCATION` for Softimage).

> For details on how Softimage uses the linktab file, see 'Configuring the `linktab.ini` File' in the Softimage 'Setup & Licensing Guide'.