

Path Mapping

How Arnold handles paths

When you render a frame, here's what Arnold does for file paths:

1. Replace backslashes

When Arnold loads an ass file, Arnold replaces all backslashes (\) with forward slashes (/)

2. Expand environment variables

When Arnold uses a parameter, Arnold expands all environment variable references, which look like this: [MY_TEXTURE_PATH]

3. Map paths

After Arnold expands environment variables, Arnold applies the path-mapping rules specified by the Arnold pathmap file.

Arnold expands environment variable expansion and maps paths for:

- search paths in the options node
- filename parameters for these nodes:
 - alembic
 - all driver nodes such as driver_deepexr, driver_exr, driver_jpg, driver_png, and driver_tiff
 - image
 - include_graph
 - materialx
 - photometric_light
 - procedural
 - volume
 - volume_implicit

Setting up path mapping

You can automatically remap paths at render time using a pathmap file.

To use a pathmap

- Set the ARNOLD_PATHMAP environment variable to point a pathmap file

The pathmap file is a json file. For example:

```
{
  "windows": { "S:/" : "\\server\projects\" },
  "mac": { "S:/" : "/Volumes/projects/" },
  "linux": { "S:/" : "/mnt/projects/" }
}
```

Path mapping uses [regular expressions](#). The general format of an entry is this:

```
{
  "windows": { "regular expression": "replacement string" },
}
```

For example, this pathmap replaces all drive mappings like E:/ and S:/ with //SERVER/

```
{  
  "windows": { "[A-Z]:" : "//SERVER/" },  
}
```

Arnold converts backslashes (\) to forward slashes (/) when it loads the ASS file.

Path mapping happens after that, so pathmaps never have to deal with backslashes.

There can be multiple mappings for each OS:

```
{  
  "windows": { "[A-Z]:" : "//SERVER", "sourceimages/" : "textures/" },  
}
```