RenderMan for Maya

Introduction

Assuming that your site happens to be using the default, straight-out-of-the-box configuration of Tractor, the default configuration of RenderMan for Maya (RfM) can distribute work via Tractor. If, perchance, you think you're special and opted to change the default configuration, fear not. RfM provides all the hooks you need, via preferences and other doodads, to maintain a harmonious symbiosis between your RfM workstations and your Tractor work dispatching system.

RfM Custom Configuration

Tractor configuration for RfM is done via the rfm.json file. For site configuration of RfM, we recommend not changing the shipped rfm.json file. Instead, sites should use the environment variables RFM_SHOW_PATH or RFM_SITE_PATH, to source your custom overrides. For more information, please see the RenderMan for Maya documentation page, Installing Custom Nodes.

Engine Location and Tractor Users

The rfm.json file provides three preferences for sealing RfM and Tractor's bond, all under the "tractor_cfg" section. The defaults look like this:

```
"tractor_cfg": {
    "engine": "tractor-engine",
    "port": 80,
    "user": ""
}
```

- "engine" is the hostname of the machine that is running your site's tractor engine. The default is "tractor-engine"
- "port" is the port number your tractor engine is listening on. The default is 80.
- "user" is the username that the job should be submitted as. The default is "", meaning to submit the job as the current logged in user.

Here's a full example of a rfm.json file that simply overrides the Tractor configuration settings for RfM and leaves everything else at default.

```
{
    "tractor_cfg": {
         "engine": "woody-engine",
         "port": 8080,
         "user": "andy"
   }
}
```

Directory Mapping

As with setting the engine location and user, directory mappings are also specified in the rfm.json file. Multiple directory mappings can be listed under the "dirmaps" section. The default is a generic mapping:

```
    "dirmaps": {
        "myserver_to_z": {
            "to": "Z:/",
            "from": "/myserver/",
            "zone": "UNC"
        }
    },
```

- "myserver_to_z" is a generic descriptive name for the dirmap
- "from" is the part of the path you want to match and mapped to
- "to" specifies what the substring in the "from" setting should be mapped to
- The "zone" specifies the zone of the blade that should use this dirmap.

For example, using this dirmapping, if RenderMan and/or Tractor sees this path and it is running on a blade that is in the "UNC" zone:

/myserver/path/to/maya/scenes/char_animation_01.ma

This will get translated to:

Z:/path/to/maya/scenes/char_animation_01.ma

Here's a full example of a rfm.json that combines the previous configuration with custom directory mappings. We have a mappings to go from Linux to a Windows blade for our project and texture paths, as well as the reverse Windows to Linux.

```
{
    "tractor_cfg": {
         "engine": "woody-engine",
         "port": 8080,
         "user": "andy"
   },
    "dirmaps": {
        "project_linux_to_windows": {
            "to": "Z:/",
            "from": "/mnt/",
            "zone": "UNC"
        },
        "textures_linux_to_windows": {
            "to": "Y:/",
            "from": "/texture_server/",
            "zone": "UNC"
        },
        "project_windows_to_linux": {
            "to": "/mnt/",
            "from": "Z:/",
            "zone": "NFS"
        },
        "textures_windows_to_linux": {
            "to": "/texture_server/",
            "from": "Y:/",
            "zone": "NFS"
        }
    }
}
```