

Installation

The Arnold for Katana plugin comes with Arnold core bundled but without the SDK. Setting it up generally requires pointing to your license server(s), setting `KATANA_RESOURCES` and launching Katana.

Installing KtoA

The download package comes in a self-extracting installer. Downloads of KtoA can be obtained [here](#).

Linux

Change the downloaded installer to be executable, for example, if you had downloaded KtoA version 3.4 for Katana version 3.6 in a terminal:

```
chmod a+x KtoA-3.4.0.0-kat3.6-linux.run
```

Then run the installer in a graphical environment to get a walk-through of the installation. You should have installed your desired Katana version previously.

```
./KtoA-3.4.0.0-kat3.6-linux.run
```

Windows

Run the downloaded installer, which will pop up a window to walk you through the installation process. You should have installed your desired Katana version previously.

Running KtoA

KtoA has a bundled script for launching KtoA with your desired version of Katana. It is recommended to customize that script (or a copy of it) for your environment and use it or a shortcut to launch KtoA. See the files `launchKtoA.sh` (Linux) and `launchKtoA.bat` (Windows) for a typical launch script.

KtoA comes bundled with Arnold core and is tuned to work with the version it is bundled with. It also has additional plugins, such as common shaders, volume plugins (OpenVDB volumes), kick, the Arnold Python API, and so on. Generally, to launch Katana with KtoA, you need to expand the KtoA tarball and set the following environment variables, customized to your environment (the `PATH` and license variables are not required):

```
export foundy_LICENSE=4101@localhost
export ADSKFLEX_LICENSE_FILE=5053@localhost
export solidangle_LICENSE=5053@localhost
export DEFAULT_RENDERER=arnold
export "KATANA_TAGLINE=With KtoA and Arnold"
export KTOA_ROOT=/path/to/KtoA
export PATH=/path/to/Katana:${KTOA_ROOT}/bin:$PATH
# NOTE: this is the only absolutely required variable
export KATANA_RESOURCES=${KTOA_ROOT}
```



Each KtoA version comes pre-qualified for particular Arnold core and Katana versions and comes bundled with a particular Arnold core version. Please see the release notes for each KtoA version for compatibility.

Setting Environment Variables

Note that on Windows, KtoA comes with an example script to launch Katana with KtoA: `launchScript.bat`. You can readily customize that script rather than set your own environment variables.

On Linux, after installing Katana and downloading KtoA, the following environment variables ought to be set, and an example of each is given:

- `KTOA_ROOT` is convenient for setting the other environment variables; neither KtoA nor Katana make direct use of it.
- `PATH` could be modified to include at least the KtoA bin directory, and additionally, you can include `$KATANA_HOME` as well to place the `katana binary` in your path for convenience.
- `KATANA_RESOURCES` should be set or modified to include the KtoA renderer resources; this is simply the path to the KtoA installation directory.

If you are using a bash shell on Linux, you should consider creating a script to set the main environment variables and launch Katana for you.

Configuring USD Plugins

KtoA ships with multiple USD plugins, a Render Delegate, a Node Registry Plugin, a custom build of Foundry's KatanaUsdPlugins, and plugins for UsdIn. These plugins can be activated by either using the launch scripts or manually setting the environment variables listed below.

Render Delegate and Node Registry plugin

KtoA includes two components from [Arnold-USD](#) to allow using Arnold directly in the Hydra Viewport.

On Linux:

```
export "FNPXR_PLUGINPATH=${KTOA_ROOT}/USD/Viewport:${FNPXR_PLUGINPATH}"
```

On Windows:

```
set "FNPXR_PLUGINPATH=%KTOA_HOME%\USD\Viewport;%FNPXR_PLUGINPATH%"
```

Katana USD Plugins

KtoA includes a custom build of Foundry's Katana USD Plugins, which replaces Katana's one when using the launch scripts. There is also a set of plugins for UsdIn, named KatanaUsdArnold, included in the installation. These allow importing Arnold specific information, like shaders, using UsdIn.

On Linux:

```
export "LD_LIBRARY_PATH=${KTOA_ROOT}/USD/KatanaUsdPlugins/lib:${KTOA_ROOT}/USD/KatanaUsdPlugins/plugin/Libs:${KTOA_ROOT}/USD/KatanaUsdArnold:${LD_LIBRARY_PATH}"
export "KATANA_RESOURCES=${KTOA_ROOT}/USD/KatanaUsdPlugins/plugin:${KATANA_RESOURCES}"
export "PYTHONPATH=${KTOA_ROOT}/USD/KatanaUsdPlugins/lib/python:${PYTHONPATH}"
```

On Windows:

```
set "path=%KTOA_HOME%\USD\KatanaUsdPlugins\lib;%KTOA_HOME%\USD\KatanaUsdPlugins\plugin\Libs;%path%"
set "KATANA_RESOURCES=%KTOA_HOME%\USD\KatanaUsdPlugins\plugin;%KTOA_HOME%\USD\KatanaUsdArnold;%KATANA_RESOURCES%"
set "PYTHONPATH=%KTOA_HOME%\USD\KatanaUsdPlugins\lib\python;%PYTHONPATH%"
```



The customizations applied to the Foundry plugin can be found [here](#).

More Environment Variables

There are a few more environment variables of interest:

- `KTOA_LOAD_VERBOSITY`: how chatty do you want Arnold to be when loading shaders/plugins e.g. for Katana material nodes. Values are `quiet`, `normal`, and `debug` is the default. This can help diagnose issues such as missing shaders or OSL shaders that can't be compiled.
- `PYTHONPATH`: you may want to extend this to include the python subdirectory of your Arnold installation (`${KTOA_ROOT}/python`)
- `(deprecated) ARNOLD_SHADERLIB_PATH`: you may extend the areas that Katana searches for your Arnold shaders with this variable
- `ARNOLD_PLUGIN_PATH`: you may extend the directories that KtoA searches for your Arnold shaders, procedurals, volumes, drivers, etc. plugins
- `PATH`: if you wish to use kick or maketx directly from KtoA, please append to this the path to the bin subdirectory of KtoA
- `LD_LIBRARY_PATH`: usually does not need to be set, but to get access to libai.so it may be necessary to append the bin subdirectory of KtoA
- `(deprecated) ARNOLDTREE`: This is not used anymore. It is used to specify the path to your Arnold installation root. This is not used by Arnold itself, but by Katana's Arnold integration; if you do not set this, you could instead symlink the path from your Arnold installation to `/usr/local/arnold`

Licensing Environment Variables

Although not specific to Katana, you may want to set the following environment variables to remove the Arnold watermark from your rendered images:

- `foundry_LICENSE` is set to your port and license server for Katana, e.g. `4101@localhost`; please see The Foundry's documentation on licensing of Katana for more information
- `ADSKFLEX_LICENSE_FILE` is set to your `port@host` for Autodesk CLM licensing of Arnold
- `solidangle_LICENSE` is set to your port and license server for Arnold, e.g. `5053@localhost`; or, it is set to the path of your node-lock license file
- `(deprecated) ARNOLD_LICENSE_HOST` is set to your license server for Arnold, possibly localhost

- (deprecated) `ARNOLD_LICENSE_PORT` is set to your license server's port, usually 5053