



# SideFX<sup>®</sup> EDU

## A guide on contributing to SideFX EDU

If you're reading this, you might be thinking about submitting a tool to SideFX EDU, or even better, you've submitted something already and you're looking for a refresher.

If you are in doubt as to how to proceed please contact the Education & Training Department ([education@sidefx.com](mailto:education@sidefx.com)) and we can help you out.

Before we look at what it means to contribute to SideFX EDU, we want to make it clear that it is not an experimental graveyard. This means that if you intend on contributing as SideFX staff you intend on maintaining the tool (or delegate the responsibility) if issues arise due to updates to Houdini and so forth. If you are not SideFX staff but still want to contribute, we recommend you get in touch with us before going through this guide.

We also want to make sure that SideFX EDU provides quality experiences to the instructors and learners; and user experience is key to this. Many of the tools wrap up common concepts and workflows into high level tools so learners find them valuable as-is for learning and experimenting.

Depending on the type of contribution, there are different types of requirements. This document will try to explain most of them. If what you are trying to do is not described, please contact us.

Finally, SideFXEDU is open source. This means we expect contributors to provide us the source code / unlocked assets to share with the community. You should not expect any direct returns from your contribution, except for being credited in the release journal on the forum.

# Contribution Checklist

## HDAs

1) The tool definition must use the following formatting with the **'edu' namespace**, i.e.:

```
edu::my_tool_name::1.0
```

2) The tool must be put in the right **tab submenu**. You can set this in the *Type Properties > Interactive > Context* tab. Valid values are Education/<category\_name>, i.e.:

*Education/Rendering, Education/Math, Education/Utility...*

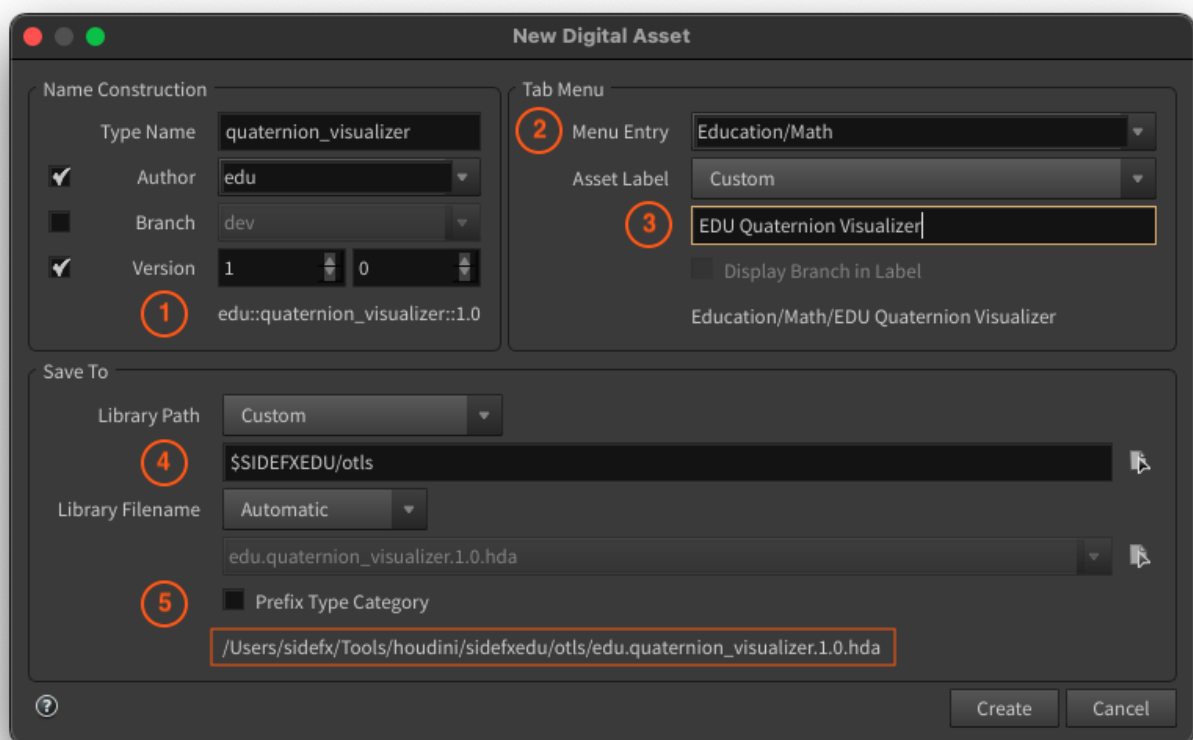
3) The tool label must use **EDU as a prefix**, i.e.

EDU My Tool Label

4) You can store the definition of the HDA anywhere on disk, but if you want it to be loaded with the SideFX EDU package, you should put it inside the /otls folder, inside the SideFX EDU package folder, i.e.:

```
$SIDEFXEDU/otls
```

5) The definition must **not be prefixed** with the type category



- The tool **must have an icon**; the icon must look similar in style and color to the icons used in other tools found in EDU. It must **use the EDU hat** icon. This icon must also be of **type SVG**, and not a rasterized image to ensure scalability. Refer to the [Icon section](#) for some examples and the color palette.

- The inputs and outputs of the tool have been given a **proper tooltip**. You can set these in the *Type Properties > Input/Output* tab of the tool.

- If your HDA is a SOP or a TOP, you need to make sure you **have an output node** at the bottom of your network. This ensures that regardless of where the display flag is set that it produces the same output.

- On the Scripts tab, add the OnCreated() function and paste the following code in it:

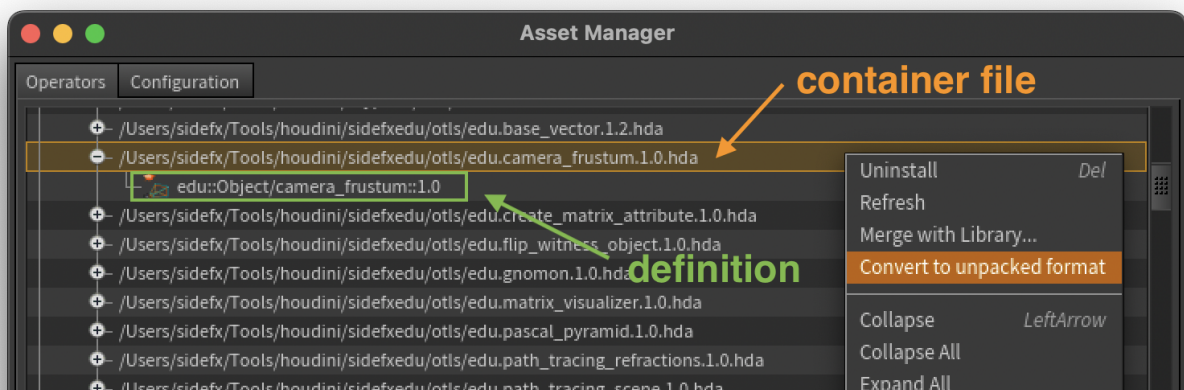
```
# This code changes the name of the node and adds a comment
# So the user is well aware this is an Educational tool
node = kwargs['node']
name = node.name()
node.setName('edu_'+name, unique_name=True)
node.setComment("Education tool")
node.setGenericFlag(hou.nodeFlag.DisplayComment, True)
```

- The tool needs documentation. To create documentation the right way, please refer to the [Documentation](#) section below.

- The tool needs a **simple demo scene inside the /hip folder** where the functionality of the tool is demonstrated. See examples inside the SideFXEDU repo, under the /hip folder: `$SIDEFXEDU/hip/`.

- All tools need to be in the **expanded format** for diffing on Github.

- Right-click on an instance of your HDA and choose "Show in Asset Manager".
- The Asset Manager opens with the HDA definition selected.
- Select the container (i.e. the file containing the definition) and right-click on it.
- Select "Convert to unpacked format".



Please refer to the following page for more information on how to expand your HDAs: [Working with files and assets as text](#).

---

## Misc tips

If you are making significant changes to an HDA, please bump the version of the tool, i.e.

`edu::gnomon::1.0` becomes `edu::gnomon::2.0`. This ensures backwards compatibility, and ensures we don't break any automatic scripts users have made. It is recommended to use the new HDA tool to create and manage HDAs. This makes namespaces and labeling very easy.

If you don't have Github access, you should send the following in a .zip to the [education@sidefx.com](mailto:education@sidefx.com):

- ☐ HDA compliant with the above checklist
- ☐ Icon in SVG format
- ☐ Help documentation in txt file
- ☐ Demoscene .hip (as simple as possible, with comments and stickies where appropriate)
- ☐ A video tutorial no longer than 5 mins (Optional but much appreciated!)

---

## Icons

- Your .svg file icon must be of **ratio 1:1**. Houdini expects icons to have a 1:1 aspect ratio, so make sure your SVG has that to prevent any weird distortion.

- Your icon must include the little **EDU Hat** at the top left of your icon, i.e.



- The colors we typically use are: Black `#000000` / White `#ffffff` / Orange `#ff6600`

---

## Documentation

### HDAs

Any tool submitted should have a basic pass of tooltips added to the helpcard. Assuming you've installed SideFX Labs, you should be able to generate a help card template with python in Houdini.

*Tip: set the icon before running the script below, so it can add it automatically to the page.*

- Select an instance of your HDA and run this code in the Houdini Python Shell to create a template page:

```
from edu_utils import create_node_help_auto
create_node_help_auto(hou.selectedNodes()[0])
```

- A file named after the HDA should be created inside `$SIDEFXEDU/help/nodes/<HDA_TYPE>/`.

For instance for the `edu::gnomon::1.0` HDA, the help file will be created here:

```
$SIDEFXEDU/help/nodes/sop/edu--gnomon-1.0.txt
```

- Make sure to add the `#sidefxedu` tag at the beginning of the page, as well as any other tag you deem useful. These tags can be used on the SideFX EDU help page to help the user filter and search for nodes:

`#tags: education, rendering, math, utility`

- Make sure the title of your help page also prefixes the title with EDU.

- Use this [Wiki Markup Reference page](#) as a reference on how to write the node's help page.

- Of course, once you've finished writing the help page, check that it works by pressing the help button on the parameter interface of the tool (or hover the parameters to see the tooltips).