

Installation

The release archive contains everything you need to run HtoA, including the Arnold libraries, executables, and Python modules.

- [Houdini Compatibility](#)
- [Setting up HtoA with the installer](#)
- [Setting up HtoA manually](#)
 - [Modifying the HOUDINI_PATH](#)
 - [Setting up the Hydra render delegate for Solaris](#)
 - [Simple installation with houdini.env](#)
 - [Using a Wrapper Script](#)
 - [Modifying individual HOUDINI_*_PATHs](#)
- [Further information](#)

Houdini Compatibility

- The HtoA build must match **exactly** your Houdini install, down to the build number, or random crashes and glitches will occur.
- HtoA requires a 64-bit Houdini, Houdini FX or Houdini Education.

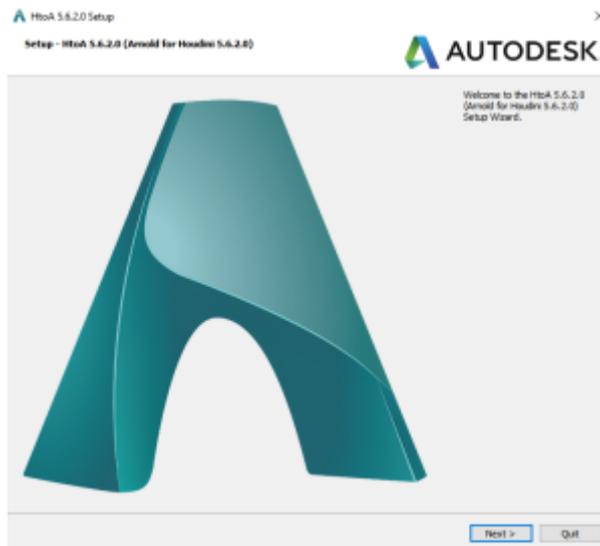


Houdini Apprentice, unfortunately, *cannot run HtoA* as they lack support for third-party renderers, see [Houdini products comparison](#).

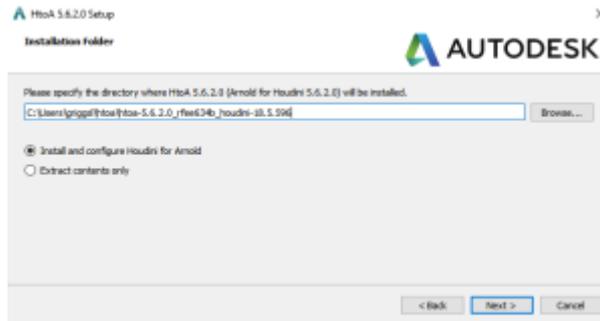
Setting up HtoA with the installer

Here are step-by-step instructions to install HtoA for the current user.

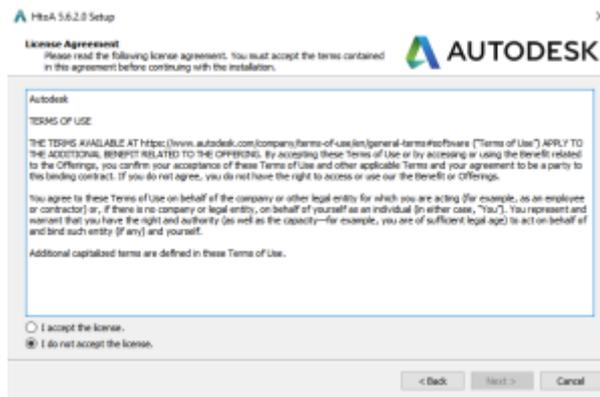
- Download the installer matching your **exact Houdini version** and platform from [Downloads](#).
- Launch the installer:



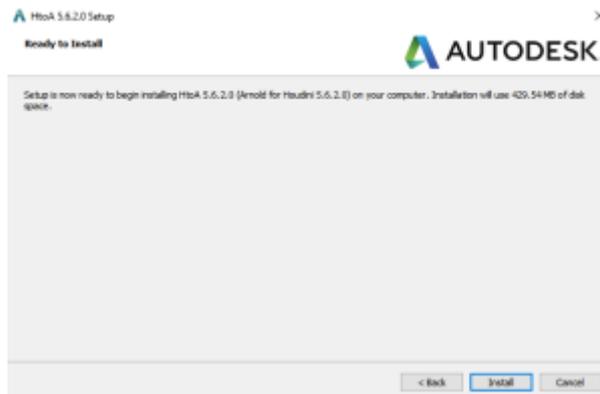
- Select the folder where you want the HtoA files to be extracted:



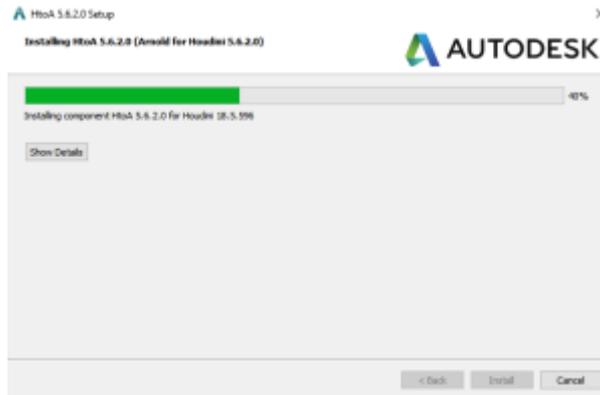
- You can choose to either have the installer set up your Houdini config file for HtoA automatically for you or just extract the contents to do a **manual installation**. The automatic installation will modify a few lines in your `houdini.env` file and create a backup.
- You need to accept the End User License Agreement to continue the installation:



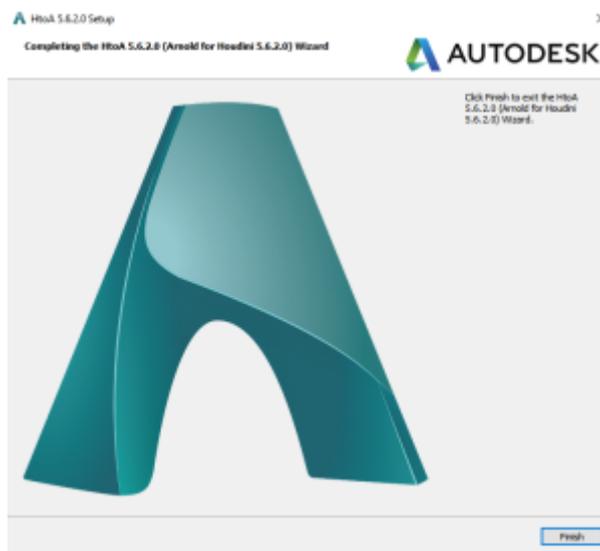
- You are now ready to install HtoA on your computer:



- The actual installation is then performed:



- Done! You can now launch Houdini and start using Arnold:



Setting up HtoA manually

If you need to have complete control of the HtoA installation, you can select to only extract the contents of the installer and configure Houdini to pick up HtoA as shown below.

Note the render delegate for Solaris is added automatically for Houdini versions 18.0 and above, so no action is required when using the installer.

Modifying the HOUDINI_PATH

The installation boils down to inserting the HtoA folder first in the `HOUDINI_PATH` environment variable. On Windows, you also need to update the `PATH` environment variable.

 The path to the HtoA folder must be inserted *before* the factory Houdini folders in the `HOUDINI_PATH`. We suggest you put it first.

Setting up the Hydra render delegate for Solaris

Note that from HtoA 5.1.1 and Houdini 18.0.391, the render delegate for Solaris is installed into the `dso/usd_plugins` directory so it is no longer necessary to set the `PXR_PLUGINPATH_NAME` either in the `houdini.env` file or as an environment variable.

For previous versions, the Arnold hydra render delegate was added automatically by the installer to the `houdini.env` file, by adding the `PXR_PLUGINPATH_NAME` variable, set to the location of the `hydra` folder within the HtoA install tree as below.

Simple installation with houdini.env

The simplest installation is to insert the HtoA folder path to the `houdini.env` file. This file is created by Houdini after your first run, its location varies per platform:

Platform	Location
Linux	~/houdini18.0/houdini.env
Windows	C:\Documents and Settings\ <username>\My Documents\houdini18.0\houdini.env</username>
Mac	~/Library/Preferences/houdini/18.0/houdini.env

On Linux and Mac, append a single line to enable HtoA for the user:

houdini.env (Linux)

```
HOUDINI_PATH = /path/to/htoa/htoa-5.0.2_raba8949_houdini-18.0.348;&
```

houdini.env (Mac)

```
HOUDINI_PATH = /path/to/htoa/htoa-5.0.2_raba8949_houdini-18.0.348;&
```

On Windows, you need to set the `PATH` to point to the `scripts/bin` as well:

houdini.env (Windows)

```
PATH = "$PATH;C:/path/to/htoa/htoa-5.0.2_raba8949_houdini-18.0.348  
/scripts/bin"  
HOUDINI_PATH = "C:/path/to/htoa/htoa-5.0.2_raba8949_houdini-  
18.0.348;&"
```



- Prior to Houdini 15.5.559, must use forward slashes ("/) on Windows instead of backslashes as path separators. For example: "C:/htoadeploy/12".
- In the `PATH`, the HtoA binaries folder must be **appended** (not *prepended*) to avoid issues with the *Houdini Engine* plugin in other applications.

Using a Wrapper Script

The most flexible way to deploy HtoA is to use a wrapper to launch Houdini. On Linux for example, you could use this script:

htoa.sh

```
#!/bin/bash
# Sample bash script to launch Houdini with HtoA enabled

# edit these to suit your environment
HOUDINI_ROOT="/opt/hfs18.0.348"
HTOA="/path/to/htoa/htoa-5.0.2_raba8949_houdini-18.0.348"

# source houdini environment
cd ${HOUDINI_ROOT}
source houdini_setup
cd - &> /dev/null

# View docs in the default browser
export HOUDINI_EXTERNAL_HELP_BROWSER=xdg-open

# set HOUDINI_PATH
export HOUDINI_PATH="${HOME}
/houdini${HOUDINI_MAJOR_RELEASE}.${HOUDINI_MINOR_RELEASE};${HTOA};${
```

Modifying individual HOUDINI_*_PATHs

If you have already used the individual HOUDINI_*_PATH environments for other plugins and scripts, you may find you need to set them for HtoA instead of using only the HOUDINI_PATH environment.

In case you also set one of the following environment variables, make sure you are appending ";@;&", at the end of its value, or alternatively, add the corresponding HtoA subfolder path to it:

Environment variable	Sub-folder	Notes
HOUDINI_DSO_PATH	<htoa_folder>/ dso	
HOUDINI_OTLSCAN_PATH	<htoa_folder>/ otls	
HOUDINI_SCRIPT_PATH	<htoa_folder>/ scripts	
HOUDINI_SOHO_PATH	<htoa_folder>/ soho	It is critical that the HtoA soho sub-folder be placed before the factory Houdini folder.
HOUDINI_TOOLBAR_PATH	<htoa_folder>/ toolbar	
HOUDINI_UI_ICON_PATH	<htoa_folder>/ config /Icons	
HOUDINI_UI_PATH	<htoa_folder>/ config	

Further information

- [Environment Variables](#)