

Pandeia Engine Installation

- [1. Code](#)
- [2. Data](#)
 - [2.1. Required Data](#)
 - [2.2. Recommended Data](#)
- [3. Verify Installation](#)
- [4. Helper Script](#)

For Pandeia Engine v3.0/v3.1/v3.2

Python Requirements:

Required	Recommended
python 3.8+	3.11.3
numpy 1.17+	1.23.5
scipy	1.10.1
astropy 5+	5.2.2
photutils	1.7.0
synphot 1.0+	1.2.0
stsynphot 1.0+	1.2.0
six	1.16.0
setuptools	67.7.0

Installing the Pandeia Engine requires a python environment with the dependencies listed above, a set of telescope-specific datafiles, and an optional (but highly recommended) second set of data files for additional spectral manipulation capabilities.

1. Code

The engine can be installed with

```
pip install pandeia.engine==3.2
```

or for Roman:

```
pip install pandeia.engine==3.1
```



We highly recommend installing into a fresh python environment, such as ones provided by the Anaconda Python distribution, to avoid problems. If you experience issues after installing the Pandeia Engine, consider starting again with a fresh python installation and a clean new set of data files.

Then set up the data files.

2. Data

2.1. Required Data

In order to function, the Pandeia Engine requires one set of reference data (containing instrument definitions):

- **James Webb Space Telescope:** <https://stsci.app.box.com/v/pandeia-refdata-v3p2-jwst>
- **Nancy Grace Roman Space Telescope:** <https://stsci.box.com/v/pandeia-refdata-v3p1-roman>

Download and untar the files to a suitable location.

Environment variable: \$pandeia_refdata must point to the location of the top of the data file tree (the directory containing the VERSION_PSF file)



Installing both data packages

You CAN untar both sets of files to the same location and run calculations for Webb and Roman. The sole difference between the packages are the `jwst/` and `roman/` directories (and the `VERSION_PSF` file, to reflect that Roman PSFs were regenerated with WebbPSF 1.0, instead of WebbPSF 1.1). You can either untar the second telescope on top of the first one and overwrite any files when prompted, or copy just the second telescope's directory into the `$pandeia_refdata` location.

2.2. Recommended Data

- An **additional dataset**: <https://archive.stsci.edu/hlsp/reference-atlases> can be used to provide more spectroscopic functionality (phoenix spectral models, normalization by HST bandpasses, among others).

Once untarred, the files will produce a directory tree of `grp/redcat/trds`. The `pandeia.engine` uses the contents of the `trds` directory.

Environment variable: `$PYSYN_CDBS` must point to the `trds` directory (NOT `grp`)

3. Verify Installation

Open up a terminal with access to the conda installation, and type

```
python -c "import pandeia.engine; pandeia.engine.pandeia_version()"
```

If properly installed and configured, it should show the `refdata` version and `synphot` data directory, like this:

```
Pandeia Engine version: 3.2
Pandeia RefData version: 3.2
Pysynphot Data: /your/data/directory/synphot
```

or for Roman:

```
Pandeia Engine version: 3.1
Pandeia RefData version: 3.1
Pysynphot Data: /your/data/directory/synphot
```

4. Helper Script

The following script (which can also be added to your `.bashrc` or `.bash_profile` file) assumes you are operating in a BASH shell where a python installation has been installed and activated, that you've installed the `pandeia` refdata in `${HOME}/data/pandeia`, and the `cdbs` folder from the additional dataset moved and renamed to `${HOME}/data/synphot`

Installation script

```
#!/bin/sh
export PYSYN_CDBS=${HOME}/data/synphot
export pandeia_refdata=${HOME}/data/pandeia
```

Once in place, "source"ing this file will set up the `pandeia` environment.