



ASAP™ Documentation | Release 0.1.0

June 19th, 2017

LICENSE

Copyright (c) 2017, Laughlin Research, LLC

Terms of Use:

The ASAP Code, including its source code and related software documentation (collectively, the "ASAP Code"), as distributed herein and as may be subsequently revised, in whole and in part, is for government use only pursuant to development agreements between NASA, Georgia Institute of Technology, and Laughlin Research, LLC. At the time of distribution hereof, none of the ASAP Code is believed or intended to be open source. Disclosure of the ASAP Code is strictly subject to one or more restrictive covenants, including non-disclosure and non-circumvention covenants, and any use of the whole or a part of the ASAP Code constitutes acknowledgement and acceptance of said covenants. Any unauthorized use, disclosure, and/or sale of the ASAP Code or any portion thereof may be actionable under current law.

Laughlin Research, LLC retains all commercial rights to the ASAP Code.

RELEASE NOTES

- This release uses a new version of OpenCASCADE Community Edition (OCE 0.18.1 from 0.17.2) and customized PythonOCC
 - OCE 0.18.1 includes over 600 new features, enhancements, and bug fixes
 - OCE 0.18.1 is forked from OpenCASCADE Technology 6.9.1
- It is recommended that user install ASAP 0.1.0 into its own new environment as opposed to updating an existing one
 - Many new dependencies are required for OCE 0.18.1 and could be problematic if installed into existing environment
- To delete an existing Anaconda environment use:
 - `conda remove -n env_name --all`
 - `env_name` is the name of the environment to delete
- If a new (of if old one is deleted) environment is used then the instructions are similar to previous installations
 - Note the extra channel when installing `pythonocc-core` (`-c conda-forge`) and the version 0.18.2
- Images are not updated in this documentation

CONTENTS

- Prerequisites
- Installing Anaconda Python
- Installing Dependencies
- Installing ASAP
- Examples

PREREQUISITES

- ASAP is currently developed for Windows 64-bit Python 3.5
 - Pre-built binaries for geometry kernel and meshing tools are available
 - Other platforms and Python versions will be available in the future
- Anaconda Python is **strongly** recommended
 - Simple and powerful package management system
 - Pre-built binaries will self-install from Anaconda cloud

INSTALLING ANACONDA PYTHON

- Anaconda Python installers can be found here:
 - <https://www.continuum.io/downloads>
 - Recommended install directory is something like “C:\Anaconda” (avoid spaces)

Install Python 3.6 64-bit as root environment. Python 3.5 environment specifically for ASAP will be created afterwards.

The screenshot shows the Anaconda 4.3.1 download page for Windows. The page has three tabs: 'Download for Windows' (selected), 'Download for macOS', and 'Download for Linux'. The main content area is titled 'Anaconda 4.3.1 For Windows'. It includes a brief description: 'Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.' Below this is a 'Changelog' link and a list of three steps: 1. Download the installer, 2. Optional: Verify data integrity with MD5 or SHA-256 (with a 'More info' link), and 3. Double-click the .exe file to install Anaconda and follow the instructions on the screen. At the bottom, it says 'Behind a firewall? Use these zipped Windows installers'. On the right side, there are two sections for Python versions. The 'Python 3.6 version' section has a green button for '64-BIT INSTALLER (422M)' and a blue link for '32-BIT INSTALLER (348M)'. The 'Python 2.7 version' section has a blue button for '64-BIT INSTALLER (414M)' and a blue link for '32-BIT INSTALLER (339M)'. A red arrow points from the text box above to the '64-BIT INSTALLER (422M)' button.

Download for Windows Download for macOS Download for Linux

Anaconda 4.3.1

For Windows

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

[Changelog](#)

1. Download the installer
2. Optional: Verify data integrity with [MD5](#) or [SHA-256](#) [More info](#)
3. Double-click the **.exe** file to install Anaconda and follow the instructions on the screen

Behind a firewall? Use these [zipped Windows installers](#)

Python 3.6 version

64-BIT INSTALLER (422M)

[32-BIT INSTALLER \(348M\)](#)

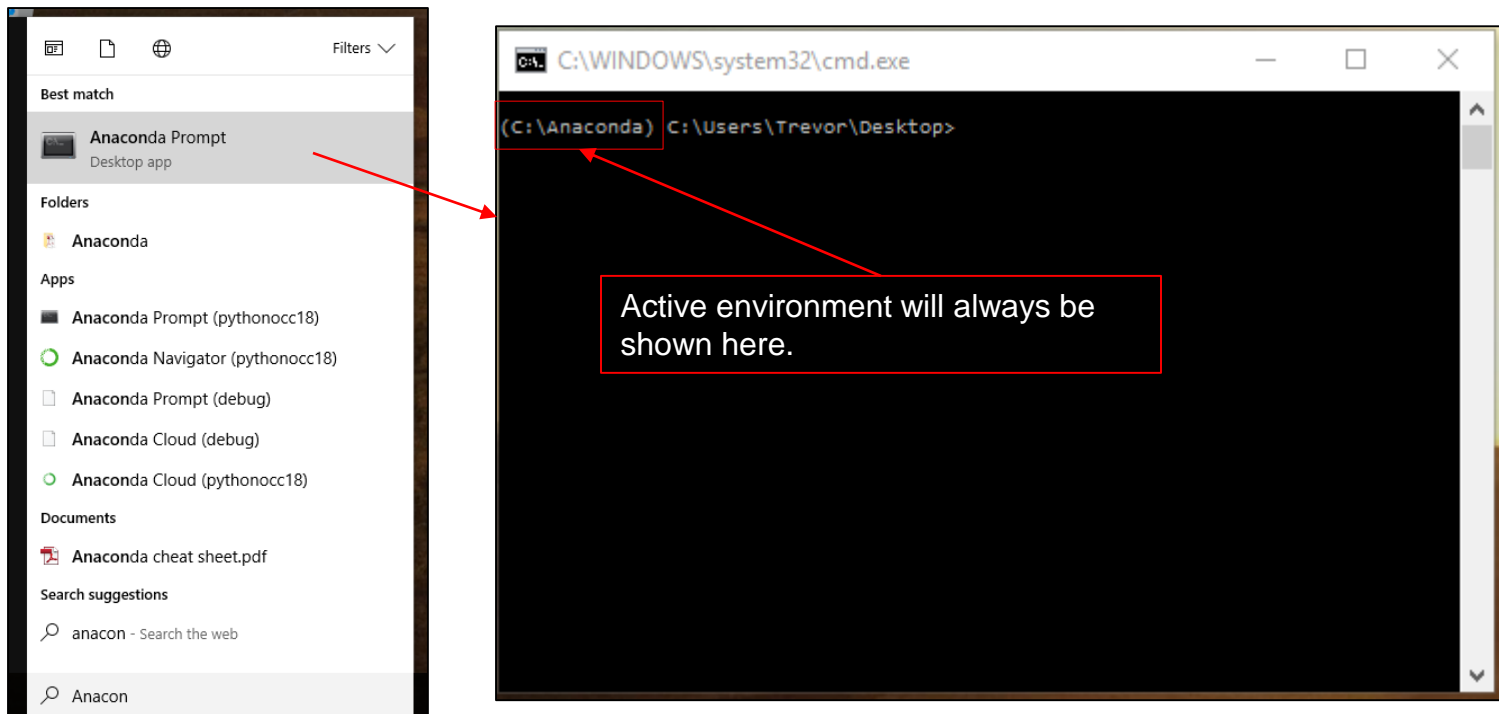
Python 2.7 version

64-BIT INSTALLER (414M)

[32-BIT INSTALLER \(339M\)](#)

INSTALLING ANACONDA PYTHON

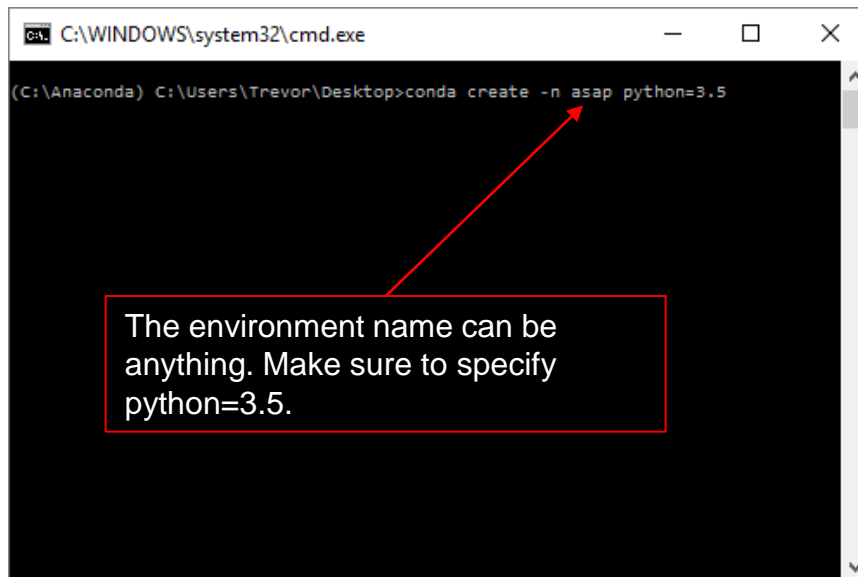
- The “Anaconda Prompt” should now be available
 - Use “conda” command to create environments, manage packages, etc.



INSTALLING DEPENDENCIES

- Create a designated conda environment for ASAP
 - This will create a clean separation between ASAP required packages and other Python installations and packages

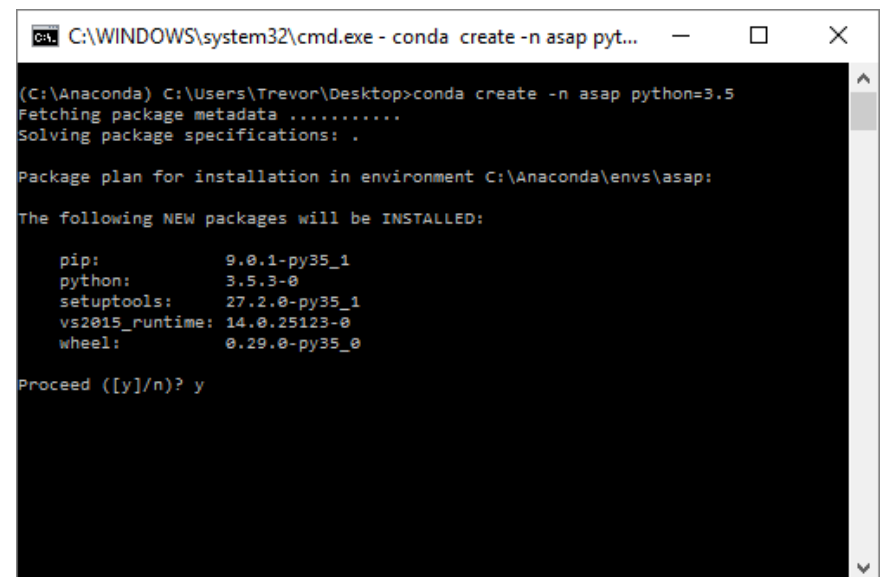
`conda create -n name python=3.5`



```
C:\WINDOWS\system32\cmd.exe

(C:\Anaconda) C:\Users\Trevor\Desktop>conda create -n asap python=3.5
```

The environment name can be anything. Make sure to specify python=3.5.



```
C:\WINDOWS\system32\cmd.exe - conda create -n asap pyt...

(C:\Anaconda) C:\Users\Trevor\Desktop>conda create -n asap python=3.5
Fetching package metadata .....
Solving package specifications: .

Package plan for installation in environment C:\Anaconda\envs\asap:

The following NEW packages will be INSTALLED:

  pip: 9.0.1-py35_1
  python: 3.5.3-0
  setuptools: 27.2.0-py35_1
  vs2015_runtime: 14.0.25123-0
  wheel: 0.29.0-py35_0

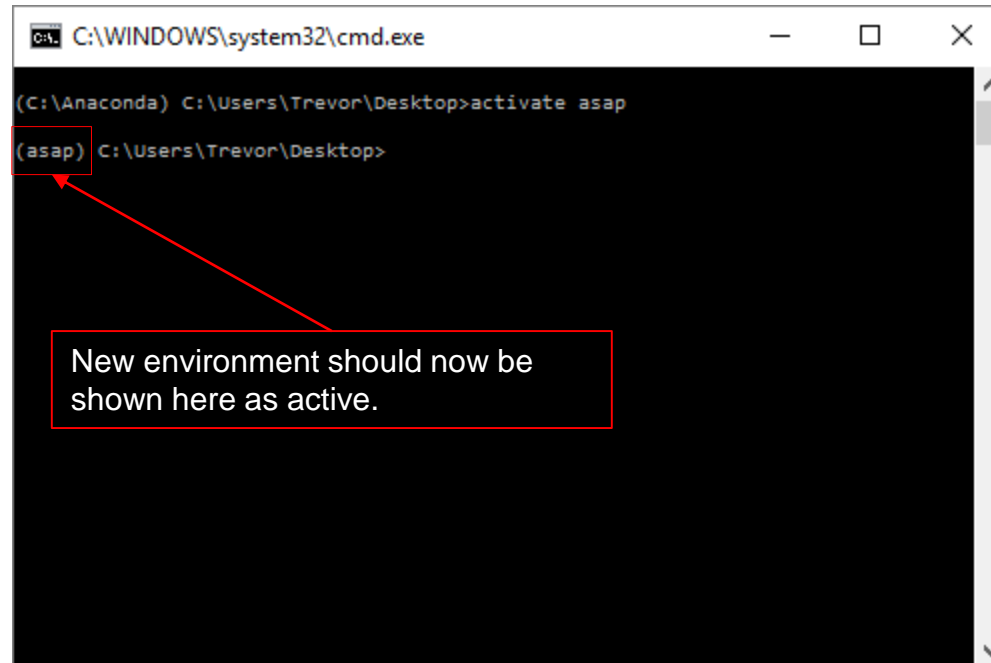
Proceed ([y]/n)? y
```

Note: A lot of junk may show up after installation in the command prompt. Ignore this. It appears to be a bug in conda.

INSTALLING DEPENDENCIES

- Activate the new environment

activate name



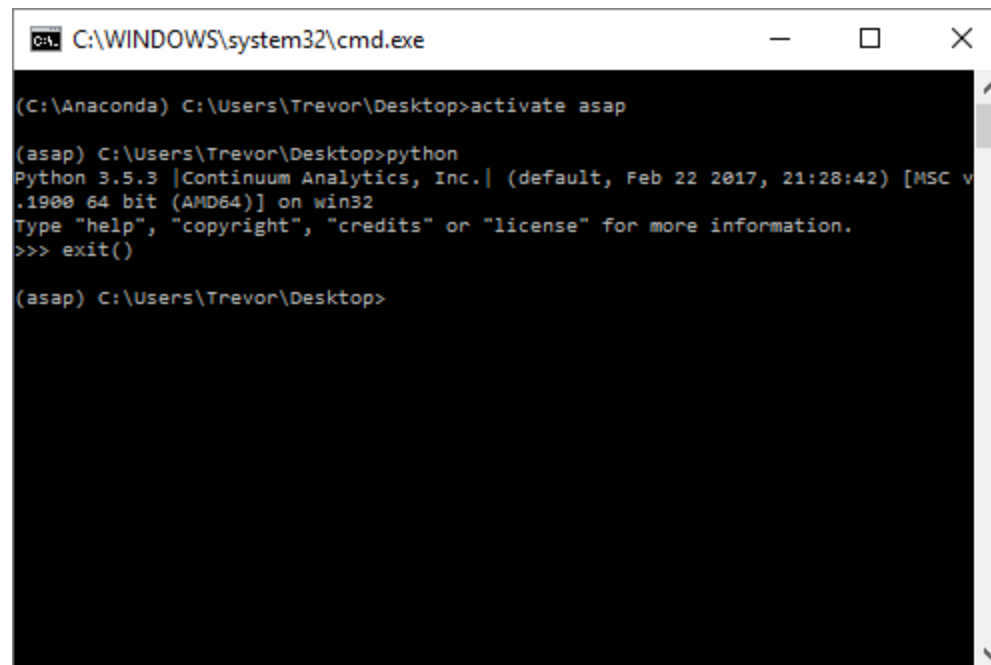
```
C:\WINDOWS\system32\cmd.exe

(C:\Anaconda) C:\Users\Trevor\Desktop>activate asap
(asap) C:\Users\Trevor\Desktop>
```

New environment should now be shown here as active.

INSTALLING DEPENDENCIES

- Activating an environment sets the systems paths such that they point to the environment Python interpreter and packages
 - Launching “python” from the activated environment should show Python 3.5
 - By default packages will be installed to active environment unless a name is given



```
C:\WINDOWS\system32\cmd.exe

(C:\Anaconda) C:\Users\Trevor\Desktop>activate asap

(asap) C:\Users\Trevor\Desktop>python
Python 3.5.3 [Continuum Analytics, Inc.] (default, Feb 22 2017, 21:28:42) [MSC v
.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

(asap) C:\Users\Trevor\Desktop>
```

INSTALLING DEPENDENCIES

- First install OpenCascade Community Edition, PythonOCC, SMESH, and Netgen from the Anaconda cloud using an Anaconda Prompt with the previously created environment activated
- All of this will be done with one command:

```
conda install -c trelau -c oce -c dlr-sc -c conda-forge pythonocc-core=0.18.2
```

install command will install the packages in the active environment

-c command adds channels to search for dependencies on the Anaconda cloud

pythonocc-core is the name of package we are searching for and installing

INSTALLING DEPENDENCIES

```
C:\WINDOWS\system32\cmd.exe - conda install -c trelau -c oce -c dlr-sc ...  
(C:\Anaconda) C:\Users\Trevor\Desktop>activate asap  
(asap) C:\Users\Trevor\Desktop>conda install -c trelau -c oce -c dlr-sc pythonocc-core  
Fetching package metadata .....  
Solving package specifications: .  
  
Package plan for installation in environment C:\Anaconda\envs\asap:  
  
The following NEW packages will be INSTALLED:  
  
  expat:          2.1.0-vc14_2      dlr-sc [vc14]  
  freeimage:     3.17.0-vc14_3     dlr-sc [vc14]  
  freetype:      2.6.3-vc14_1      dlr-sc [vc14]  
  future:        0.16.0-py35_1     [vc14]  
  gl2ps:         1.3.8-vc14_3      dlr-sc [vc14]  
  hdf5:          1.8.17-vc14_0     [vc14]  
  icu:           57.1-vc14_0       [vc14]  
  jpeg:          9b-vc14_0         dlr-sc [vc14]  
  libpng:        1.6.27-vc14_0     [vc14]  
  libtiff:       4.0.6-vc14_3      [vc14]  
  libxml2:       2.9.3-vc14_1      dlr-sc [vc14]  
  netgen:        6.2-vc14_0        trelau [vc14]  
  oce:           0.17.2-vc14_1     oce [vc14]  
  openssl:       1.0.2k-vc14_0     [vc14]  
  pyqt:          5.6.0-py35_2      [vc14]  
  pythonocc-core: smesh-py35_vc14_49 trelau [vc14]  
  qt:           5.6.2-vc14_3      [vc14]  
  sip:           4.18-py35_0       [vc14]  
  smesh:         7.7.1-vc14_61     trelau [vc14]  
  tbb:           4.3.6-vc14_1      dlr-sc [vc14]  
  vtk:           7.0.0-py35_vc14_2 dlr-sc [vc14]  
  zlib:          1.2.8-vc14_3      [vc14]  
  
Proceed ([y]/n)?
```

Verify packages and channels as shown. Other dependencies should resolve automatically. Install may take a few minutes.

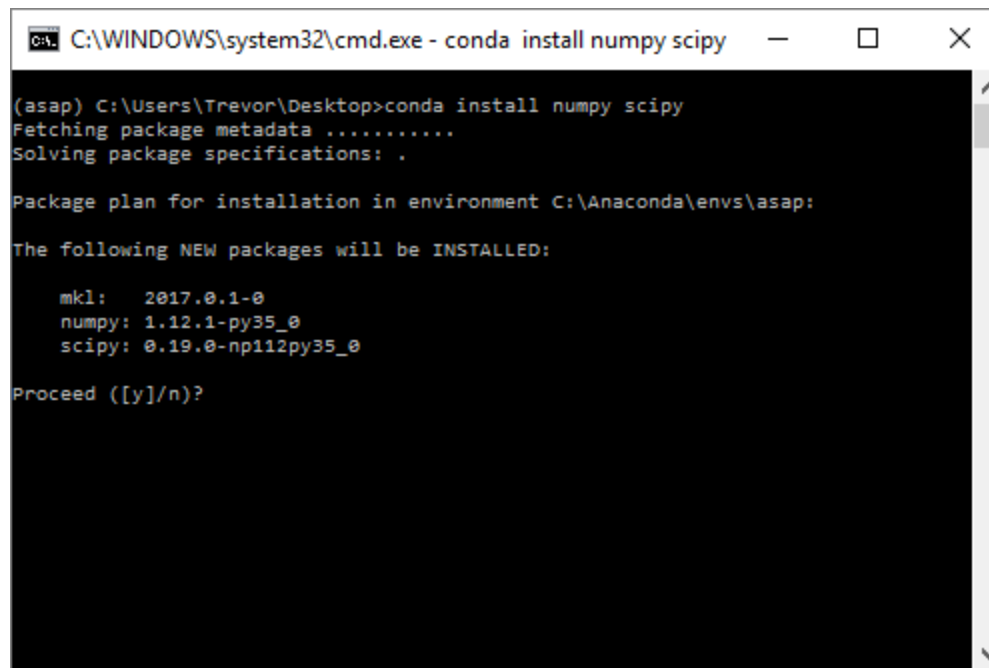
This image from ASAP 0.0.1

INSTALLING DEPENDENCIES

- The conda directory can be cleaned using the command “conda clean -a”

INSTALLING DEPENDENCIES

- Install numpy and scipy
 - “conda install numpy scipy”



```
C:\WINDOWS\system32\cmd.exe - conda install numpy scipy

(asap) C:\Users\Trevor\Desktop>conda install numpy scipy
Fetching package metadata .....
Solving package specifications: .

Package plan for installation in environment C:\Anaconda\envs\asap:

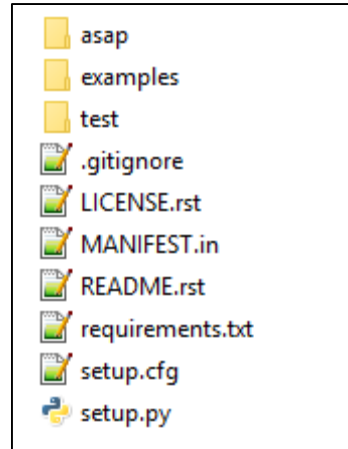
The following NEW packages will be INSTALLED:

    mkl:      2017.0.1-0
    numpy:    1.12.1-py35_0
    scipy:    0.19.0-np112py35_0

Proceed ([y]/n)?
```

INSTALLING ASAP

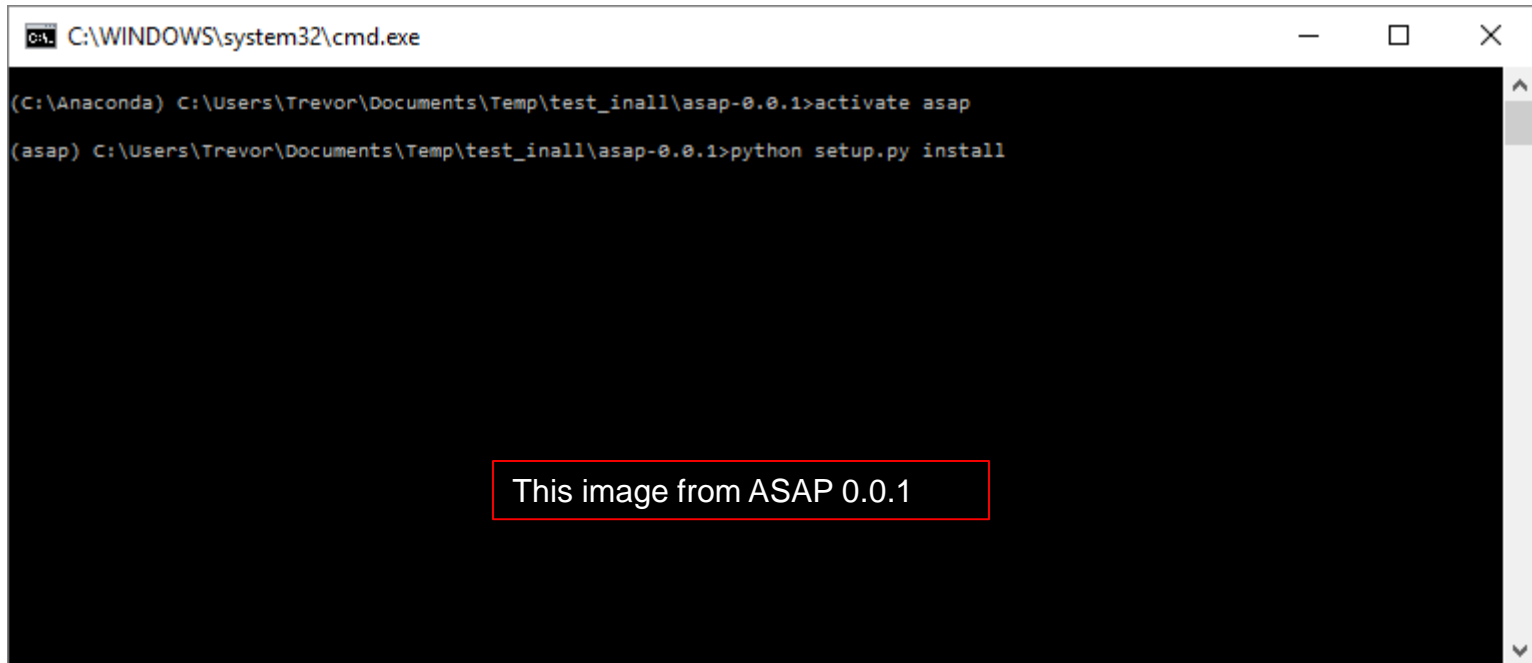
- ASAP is currently distributed as a “source distribution”
- Place “ASAP-0.1.0” in desired directory
- Contents should look like:



INSTALLING ASAP

- Launch an Anaconda Prompt from the ASAP-0.1.0 folder, activate the previously created environment, and install using the command

python setup.py install



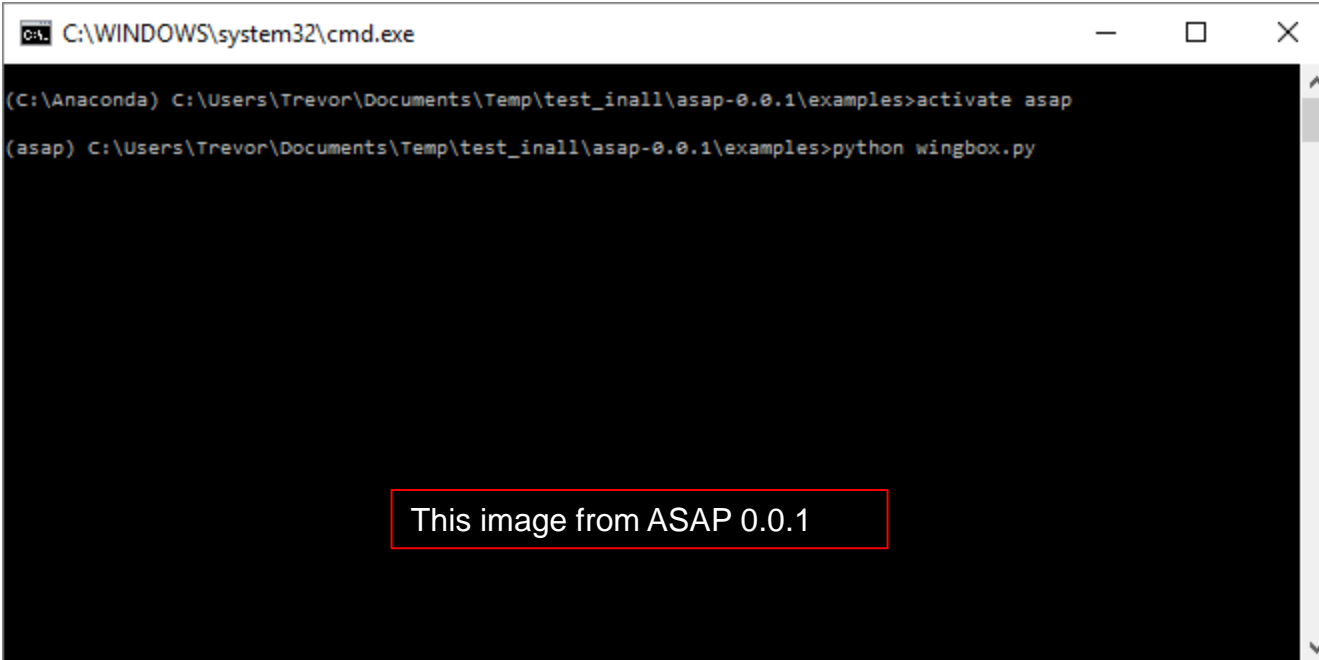
The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The prompt is at the directory "C:\Users\Trevor\Documents\Temp\test_inall\asap-0.0.1". The user has entered the command "activate asap" and the prompt has changed to "(asap) C:\Users\Trevor\Documents\Temp\test_inall\asap-0.0.1>". The user then enters the command "python setup.py install".

```
C:\WINDOWS\system32\cmd.exe
(C:\Anaconda) C:\Users\Trevor\Documents\Temp\test_inall\asap-0.0.1>activate asap
(asap) C:\Users\Trevor\Documents\Temp\test_inall\asap-0.0.1>python setup.py install
```

This image from ASAP 0.0.1

EXAMPLES

- Navigate to the “examples” folder and run the wingbox.py script from an Anaconda Prompt with the appropriate environment activated

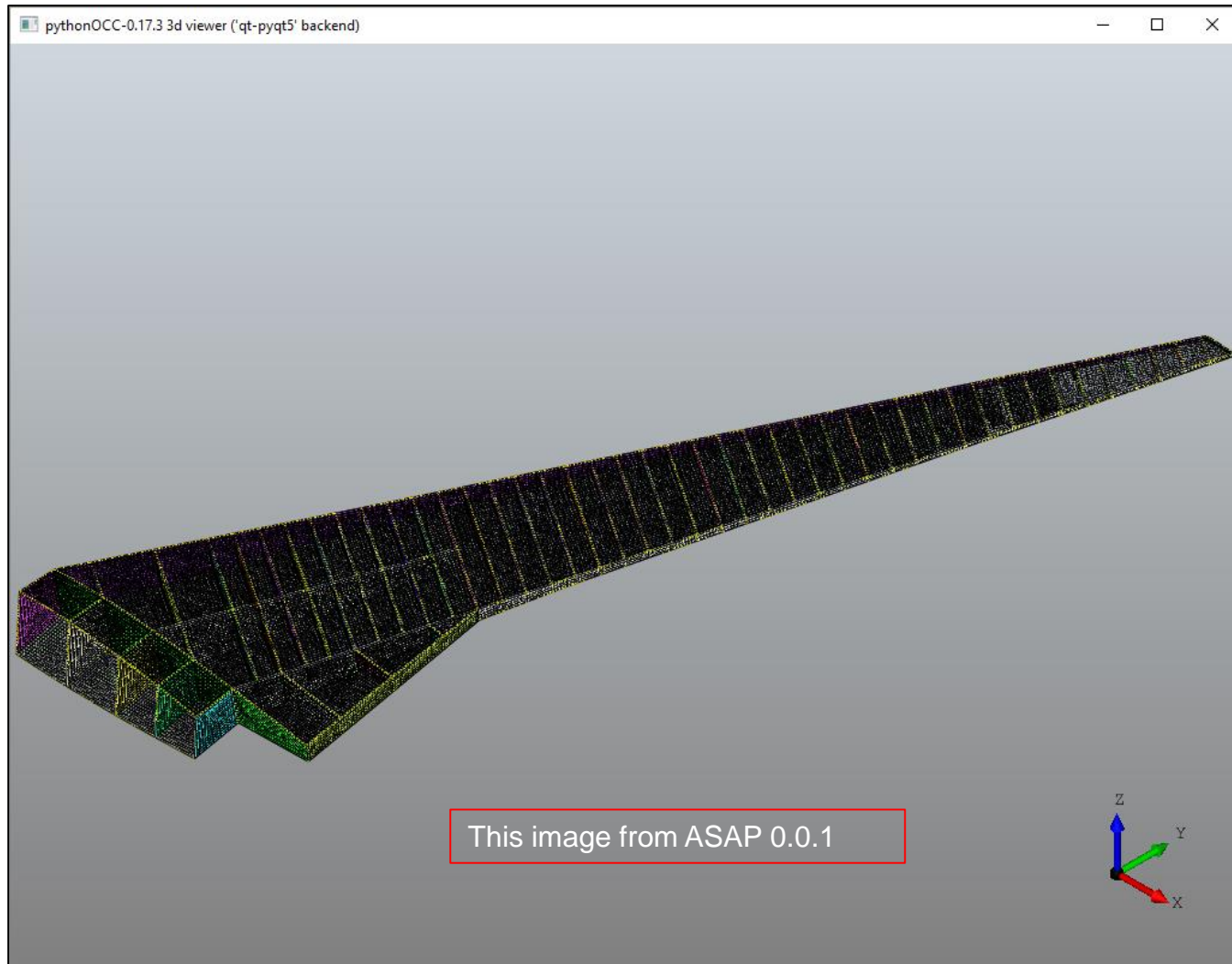


A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window shows the following commands and their output:

```
(C:\Anaconda) C:\Users\Trevor\Documents\Temp\test_inall\asap-0.0.1\examples>activate asap  
(asap) C:\Users\Trevor\Documents\Temp\test_inall\asap-0.0.1\examples>python wingbox.py
```

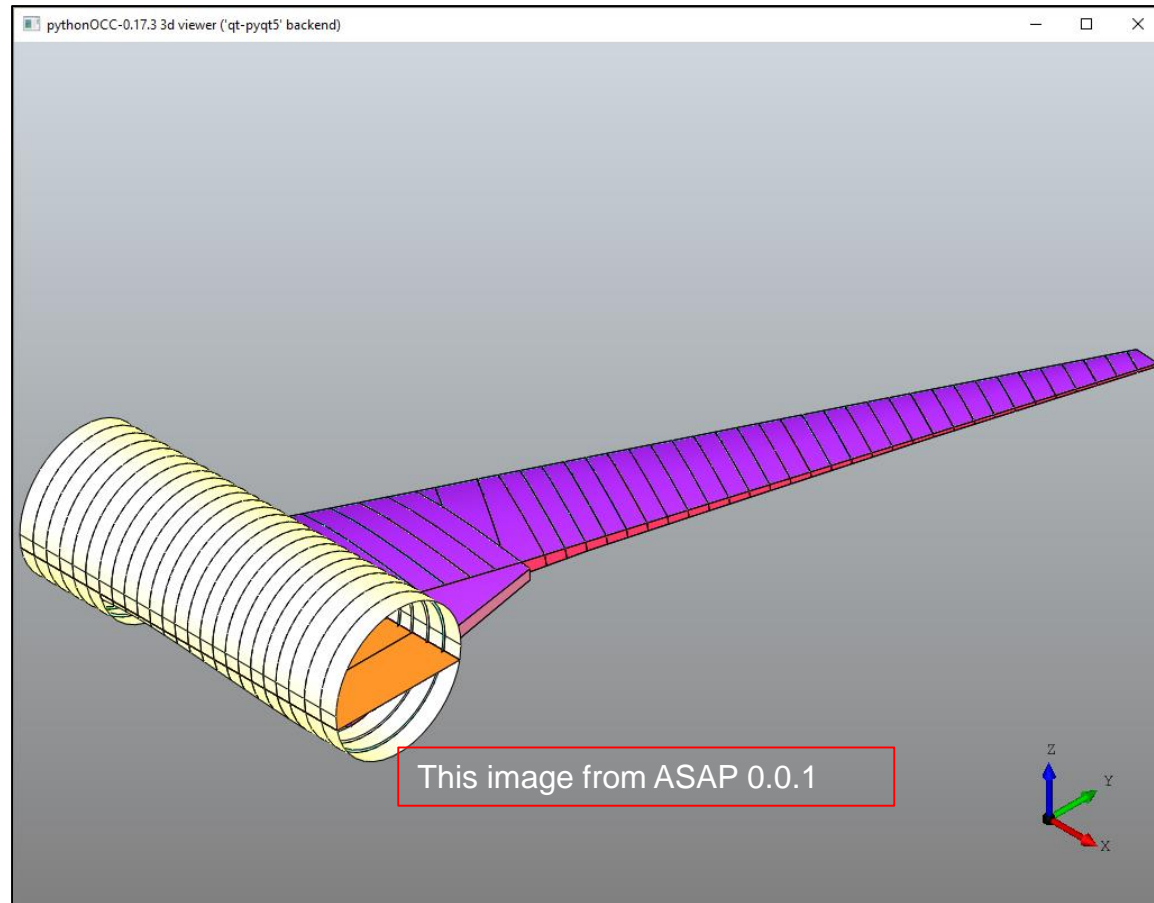
The command prompt is dark-themed. A red rectangular box is overlaid on the bottom center of the window, containing the text "This image from ASAP 0.0.1".

EXAMPLES



EXAMPLES

`python wing-body.py`





laughlinresearch