IP[y]: IPython Interactive Computing

Note

This documentation is for a development version of IPython. There may be significant differences from the latest stable release.

IPython requires Python 2.7 or ≥ 3.3.

Note

If you need to use Python 2.6 or 3.2, you can find IPython 1.0 here.

Quickstart

If you have **setuptools**, the quickest way to get up and running with IPython is:

```
$ pip install "ipython[all]"
```

This will download and install IPython and its main optional dependencies:

- jinja2, needed for the notebook
- sphinx, needed for nbconvert
- pyzmq, needed for IPython's parallel computing features, qt console and notebook
- pygments, used by nbconvert and the Qt console for syntax highlighting
- tornado, needed by the web-based notebook
- nose, used by the test suite
- readline (on OS X) or pyreadline (on Windows), needed for the terminal

To run IPython's test suite, use the **iptest** command:

```
$ iptest
```

Overview

This document describes in detail the steps required to install IPython, and its various optional dependencies. For a few quick ways to get started with package managers or full Python distributions, see the install page of the IPython website.

IPython is organized into a number of subpackages, each of which has its own dependencies. All of the subpackages come with IPython, so you don't need to download and install them separately. However, to use a given subpackage, you will need to install all of its dependencies.

Please let us know if you have problems installing IPython or any of its dependencies.

Table Of Contents

Previous topic

Installation

Next topic

Using IPython for interactive work

This Page

Show Source

Quick search

Go

Enter search terms or a module, class or function name.

IPython and most dependencies can be installed via **pip**. In many scenarios, this is the simplest method of installing Python packages. More information about **pip** can be found on its PyPI page.

> Note
>
> On Windows, IPython *requires* **setuptools**. We hope to change this in the future, but for now on Windows, you *must* install **setuptools** to use IPython.

More general information about installing Python packages can be found in Python's documentation.

Installing IPython itself

Given a properly built Python, the basic interactive IPython shell will work with no external dependencies. However, some Python distributions (particularly on Windows and OS X), don't come with a working **readline** module. The IPython shell will work without **readline**, but will lack many features that users depend on, such as tab completion and command line editing. If you install IPython with **setuptools**, (e.g. with `pip`), then the appropriate **readline** for your platform will be installed. See below for details of how to make sure you have a working **readline**.

Installation using pip

If you have **setuptools** or **pip**, the easiest way of getting IPython is to simply use **pip**:

```
$ pip install ipython
```

That's it.

Installation from source

If you don't want to use **pip**, or don't have it installed, just grab the latest stable build of IPython from here. Then do the following:

```
$ tar -xzf ipython.tar.gz
$ cd ipython
$ python setup.py install
```

If you are installing to a location (like `/usr/local`) that requires higher permissions, you may need to run the last command with **sudo**.

Windows

As mentioned above, on Windows, IPython requires **setuptools**, and it also requires the PyReadline library to properly support coloring and keyboard management (features that the default windows console doesn't have). So on Windows, the installation procedure is:

1. Install setuptools.
2. Install pyreadline. You can use the command `pip install pyreadline` from a terminal, or the binary installer

appropriate for your platform from the PyPI page.
3. Install IPython itself, which you can download from PyPI or from our site. Note that on Windows 7, you *must* right-click and 'Run as administrator' for the Start menu shortcuts to be created.

IPython by default runs in a terminal window, but the normal terminal application supplied by Microsoft Windows is very primitive. You may want to download the excellent and free Console application instead, which is a far superior tool. You can even configure Console to give you by default an IPython tab, which is very convenient to create new IPython sessions directly from the working terminal.

Installing the development version

It is also possible to install the development version of IPython from our Git source code repository. To do this you will need to have Git installed on your system. Then just do:

```
$ git clone --recursive
https://github.com/ipython/ipython.git
$ cd ipython
$ python setup.py install
```

Some users want to be able to follow the development branch as it changes. If you have **setuptools** installed, this is easy. Simply replace the last step by:

```
$ python setupegg.py develop
```

This creates links in the right places and installs the command line script to the appropriate places.

Then, if you want to update your IPython at any time, just do:

```
$ git pull
```

IPython now uses git submodules to ship its javascript dependencies. If you run IPython from git master, you may need to update submodules once in a while with:

```
$ git submodule update
```

or

```
$ python setup.py submodule
```

Another option is to copy git hooks to your `./git/hooks/` directory to ensure that your submodules are up to date on each pull.

Basic optional dependencies

There are a number of basic optional dependencies that most users will want to get. These are:

- readline (for command line editing, tab completion, etc.)
- nose (to run the IPython test suite)

If you are comfortable installing these things yourself, have at it, otherwise read on for more details.

IPython uses several other modules, such as pexpect and path.py, if they are installed on your system, but it can also use bundled versions from **IPython.external**, so there's no need to install them separately.

readline

As indicated above, on Windows, to get full functionality in the console version of IPython, PyReadline is needed. PyReadline is a separate, Windows only implementation of readline that uses native Windows calls through **ctypes**. The easiest way of installing PyReadline is you use the binary installer available here.

On OSX, if you are using the built-in Python shipped by Apple, you will be missing a proper readline implementation as Apple ships instead a library called `libedit` that provides only some of readline's functionality. While you may find libedit sufficient, we have occasional reports of bugs with it and several developers who use OS X as their main environment consider libedit unacceptable for productive, regular use with IPython.

Therefore, IPython on OS X depends on the **gnureadline** module. We will *not* consider completion/history problems to be bugs for IPython if you are using libedit.

To get a working **readline** module on OS X, just do (with **pip** installed):

```
$ pip install gnureadline
```

Note

Other Python distributions on OS X (such as Anaconda, fink, MacPorts) already have proper readline so you likely don't have to do this step.

When IPython is installed with **setuptools**, (e.g. using the `pip` command), the correct readline should be installed if you specify the `terminal` optional dependencies:

```
$ pip install "ipython[terminal]"
```

nose

To run the IPython test suite you will need the **nose** package. Nose provides a great way of sniffing out and running all of the IPython tests. The simplest way of getting nose is to use **pip**:

```
$ pip install nose
```

Another way of getting this is to do:

```
$ pip install "ipython[test]"
```

For more installation options, see the nose website.

Once you have nose installed, you can run IPython's test suite using the iptest command:

```
$ iptest
```

Dependencies for IPython.parallel (parallel computing)

IPython.parallel provides a nice architecture for parallel computing, with a focus on fluid interactive workflows. These features require just one package: PyZMQ. See the next section for PyZMQ details.

On a Unix style platform (including OS X), if you want to use **setuptools**, you can just do:

```
$ pip install "ipython[zmq]"     # will include pyzmq
```

Security in IPython.parallel is provided by SSH tunnels. By default, Linux and OSX clients will use the shell ssh command, but on Windows, we also support tunneling with paramiko.

Dependencies for IPython.kernel.zmq

pyzmq

IPython 0.11 introduced some new functionality, including a two-process execution model using ZeroMQ for communication. The Python bindings to ZeroMQ are found in the PyZMQ project, which is pip install-able. If you are on Python 2.7, 3.3, or 3.4 on OSX or Windows, pyzmq has eggs and wheels that include ZeroMQ itself.

IPython.kernel.zmq depends on pyzmq >= 2.2.

Dependencies for the IPython QT console

pyzmq

Like the **IPython.parallel** package, the QT Console requires ZeroMQ and PyZMQ.

Qt

Also with 0.11, a new GUI was added using the work in **IPython.kernel.zmq**, which can be launched with `ipython qtconsole`. The GUI is built on Qt, and works with either PyQt, or PySide.

pygments

The syntax-highlighting in `ipython qtconsole` is done with the pygments project, which is pip install-able.

Dependencies for the IPython HTML notebook

The IPython notebook is a notebook-style web interface to IPython and can be started with the command `ipython notebook`.

pyzmq

Like the **IPython.parallel** and **IPython.frontend.qt.console** packages, the HTML notebook requires ZeroMQ and PyZMQ.

Tornado

The IPython notebook uses the Tornado project for its HTTP server. Tornado 2.1 is required, in order to support current versions of browsers, due to an update to the websocket protocol.

Jinja

The IPython notebook uses the Jinja templating tool to render HTML pages.

MathJax

The IPython notebook uses the MathJax Javascript library for rendering LaTeX in web browsers. Because MathJax is large, we don't include it with IPython. Normally IPython will load MathJax from a CDN, but if you have a slow network connection, or want to use LaTeX without an internet connection at all, you can install MathJax locally.

A quick and easy method is to install it from a python session:

```
from IPython.external.mathjax import install_mathjax
install_mathjax()
```

If you need tighter configuration control, you can download your own copy of MathJax from http://www.mathjax.org/download/ - use the MathJax-2.0 link. When you have the file stored locally, install it with:

```
python -m IPython.external.mathjax /path/to/source/mathjax-
MathJax-v2.0-20-g07669ac.zip
```

For unusual needs, IPython can tell you what directory it wants to find MathJax in:

```
python -m IPython.external.mathjax -d /some/other/mathjax
```

By default Mathjax will be installed in your ipython profile directory, but you can make system wide install, please refer to the documentation and helper function of **IPython.external.mathjax**

Browser Compatibility ¶

The IPython notebook is officially supported on the following browsers:

- Chrome ≥ 13
- Safari ≥ 5
- Firefox ≥ 6

The is mainly due to the notebook's usage of WebSockets and the flexible box model.

The following browsers are unsupported:

- Safari < 5
- Firefox < 6
- Chrome < 13
- Opera (any): CSS issues, but execution might work
- Internet Explorer < 10

The following specific combinations are known **NOT** to work:

- Safari, IPython 0.12, tornado ≥ 2.2.0
- Safari with HTTPS connection to notebook and an untrusted certificate (websockets will fail)
- The diigo Chrome extension seems to interfere with scrolling

There are some early reports that the Notebook works on Internet Explorer 10, but we expect there will be some CSS issues related to the flexible box model.

Dependencies for nbconvert (converting notebooks to various formats)

pandoc

The most important dependency of nbconvert is Pandoc 1.10 or later, a document format translation program. This is not a Python package, so it cannot be expressed as a regular IPython dependency with setuptools.

To install pandoc on Linux, you can generally use your package manager:

```
sudo apt-get install pandoc
```

On other platforms, you can get pandoc from their website.