
Jupyter console Documentation

Release 5.0.0

The Jupyter Development Team

Mar 27, 2017

CONTENTS

1	Making a release as a maintainer	3
1.1	Check installed tools	3
1.2	Clean the repository	3
1.3	Create the release	3
2	Changes in Jupyter console	5
2.1	5.0	5
2.2	4.1	5
2.3	4.0	6

The Jupyter console is a terminal frontend for kernels using the Jupyter protocol. The console can be installed with:

```
pip install jupyter-console
```

And started with:

```
jupyter console
```

To see configuration options:

```
jupyter console -h
```

To start the console with a particular kernel, ask for it by name:

```
jupyter console --kernel=julia-0.4
```

A list of available kernels can be seen with:

```
jupyter kernelspec list
```

You can connect to a live kernel (e.g. one running in a notebook) with its ID:

```
jupyter console --existing KERNEL_ID
```

or even connect to the most recently started kernel by default:

```
jupyter console --existing
```

Contents:

MAKING A RELEASE AS A MAINTAINER

This document guides a maintainer through creating a release of the Jupyter console.

1.1 Check installed tools

Review `CONTRIBUTING.rst`. Make sure all the tools needed are properly installed.

1.2 Clean the repository

Remove all non-tracked files with:

```
git clean -xdi
```

This will ask you for confirmation before removing all untracked files. Make sure the `dist/` folder is clean and does not contain any stale builds from previous attempts.

1.3 Create the release

1. Set Environment variables

Set environment variables to document current release version, and git tag:

```
VERSION=4.1.0
```

2. Update version number in `jupyter_console/_version.py`. Make sure that a valid [PEP 440](#) version is being used.

3. Commit and tag the release with the current version number:

```
git commit -am "release $VERSION"  
git tag $VERSION
```

4. You are now ready to build the `sdist` and `wheel`:

```
python setup.py sdist --formats=zip,gztar  
python setup.py bdist_wheel
```

5. You can now test the `wheel` and the `sdist` locally before uploading to PyPI. Make sure to use [twine](#) to upload the archives over SSL.

```
twine upload dist/*
```

6. If all went well, change the `jupyter_console/_version.py` to the next release.
7. Push directly on master, not forgetting to push `--tags` too.

CHANGES IN JUPYTER CONSOLE

A summary of changes in Jupyter console releases.

2.1 5.0

2.1.1 Interactive Shell architecture

- disinherit shell class from IPython Interactive Shell [#68](#) This separates jupyter_console's ZMQTerminalInteractiveShell from IPython's TerminalInteractiveShell and InteractiveShell classes.
- update SIGINT handler to not use the old interactive API shell [#80](#)

2.1.2 Image Handling improvement

- use PIL as default image handler [#79](#)
- better indication of whether image data was handled [#77](#)

2.1.3 Prompts improvement

- use prompt_toolkit 1.0 [#74](#)
- don't use prompt_manager [#75](#)
- remove `colors_force` flag that have no effects: [#88](#)

2.2 4.1

- fix for readline history
- don't confuse `sys.path` with `virtualenvs`
- readline/completion fixes
- use `is_complete` messages to determine if input is complete (important for non-Python kernels)
- fix: 4.0 was looking for `jupyter_console_config` in IPython config directories, not Jupyter

2.3 4.0

- fix `jupyter console --generate-config`
- `setuptools` fixes for Windows

First release as a standalone package.