

---

# **IPython Kernel Documentation**

*Release 5.3.4*

**IPython Development Team**

**Jul 22, 2020**



---

# Contents

---

<b>1</b>	<b>Changes in IPython kernel</b>	<b>3</b>
1.1	5.3	3
1.2	5.2	4
1.3	5.1	4
1.4	5.0	5
1.5	4.10	6
1.6	4.9	6
1.7	4.8	6
1.8	4.7	7
1.9	4.6	7
1.10	4.5	8
1.11	4.4	8
1.12	4.3	9
1.13	4.2	9
1.14	4.1	10
1.15	4.0	10
<b>2</b>	<b>Indices and tables</b>	<b>11</b>



This contains minimal version-sensitive documentation for the IPython kernel package. Most IPython kernel documentation is in the [IPython documentation](#).

Contents:



---

## Changes in IPython kernel

---

### 1.1 5.3

#### 1.1.1 5.3.4

- Only run Qt eventloop in the shell stream. (PR #531)

#### 1.1.2 5.3.3

- Fix QSocketNotifier in the Qt event loop not being disabled for the control channel. (PR #525)

#### 1.1.3 5.3.2

- Restore timer based event loop as a Windows-compatible fallback. (PR #523)

#### 1.1.4 5.3.1

- Fix #520: run post\_execute and post\_run\_cell on async cells (PR #521)
- Fix exception causes in zmqshell.py (PR #516)
- Make pdb on Windows interruptible (PR #490)

#### 1.1.5 5.3.0

5.3.0 Adds support for Trio event loops and has some bug fixes.

- Fix ipython display imports (PR #509)
- Skip test\_unc\_paths if OS is not Windows (PR #507)

- Allow interrupting `input()` on Windows, as part of effort to make `pdb` interruptible (PR #498)
- Add Trio Loop (PR #479)
- Flush from process even without newline (PR #478)

## 1.2 5.2

### 1.2.1 5.2.1

- Handle system commands that use UNC paths on Windows (PR #500)
- Add offset argument to `seek` in `io` test (PR #496)

### 1.2.2 5.2.0

5.2.0 Includes several bugfixes and internal logic improvements.

- Produce better traceback when kernel is interrupted (PR #491)
- Add `InProcessKernelClient.control_channel` for compatibility with `jupyter-client` v6.0.0 (PR #489)
- Drop support for Python 3.4 (PR #483)
- Work around issue related to Tornado with `python3.8` on Windows (PR #480, PR #481)
- Prevent entering event loop if it is `None` (PR #464)
- Use `shell.input_transformer_manager` when available (PR #411)

## 1.3 5.1

### 1.3.1 5.1.4

5.1.4 Includes a few bugfixes, especially for compatibility with Python 3.8 on Windows.

- Fix pickle issues when using inline `matplotlib` backend (PR #476)
- Fix an error during kernel shutdown (PR #463)
- Fix compatibility issues with Python 3.8 (PR #456, PR #461)
- Remove some dead code (PR #474, PR #467)

### 1.3.2 5.1.3

5.1.3 Includes several bugfixes and internal logic improvements.

- Fix `comm` shutdown behavior by adding a `deleting` option to `close` which can be set to prevent registering new `comm` channels during shutdown (:ghpull: 433, :ghpull: 435)
- Fix `Heartbeat._bind_socket` to return on the first bind (:ghpull: 431)
- Moved `InProcessKernelClient.flush` to `DummySocket` (:ghpull: 437)
- Don't redirect `stdout` if `nose` machinery is not present (:ghpull: 427)



- Rename `_asyncio.py` to `_asyncio_utils.py` to avoid name conflicts on Python 3.6+ (:ghpull: 426)
- Only generate kernelspec when installing or building wheel (:ghpull: 425)
- Fix priority ordering of control-channel messages in some cases (PR #443)

### 1.3.3 5.1.2

5.1.2 fixes some socket-binding race conditions that caused testing failures in nbconvert.

- Fix socket-binding race conditions (:ghpull: 412, :ghpull: 419)
- Add a no-op `flush` method to `DummySocket` and comply with stream API (:ghpull: 405)
- Update kernel version to indicate kernel v5.3 support (:ghpull: 394)
- Add testing for upcoming Python 3.8 and PEP 570 positional parameters (:ghpull: 396, :ghpull: 408)

### 1.3.4 5.1.1

5.1.1 fixes a bug that caused cells to get stuck in a busy state.

- Flush after sending replies (PR #390)

### 1.3.5 5.1.0

5.1.0 fixes some important regressions in 5.0, especially on Windows.

[5.1.0 on GitHub](#)

- Fix message-ordering bug that could result in out-of-order executions, especially on Windows (PR #356)
- Fix classifiers to indicate dropped Python 2 support (PR #354)
- Remove some dead code (PR #355)
- Support rich-media responses in `inspect_requests` (tooltips) (PR #361)

## 1.4 5.0

### 1.4.1 5.0.0

[5.0.0 on GitHub](#)

- Drop support for Python 2. `ipykernel 5.0` requires Python  $\geq 3.4$
- Add support for IPython's asynchronous code execution (PR #323)
- Update release process in `CONTRIBUTING.md` (PR #339)

## 1.5 4.10

4.10 on GitHub

- Fix compatibility with IPython 7.0 (PR #348)
- Fix compatibility in cases where `sys.stdout` can be `None` (PR #344)

## 1.6 4.9

### 1.6.1 4.9.0

4.9.0 on GitHub

- Python 3.3 is no longer supported (PR #336)
- Flush `stdout/stderr` in `KernelApp` before replacing (PR #314)
- Allow preserving `stdout` and `stderr` in `KernelApp` (PR #315)
- Override `writable` method on `OutStream` (PR #316)
- Add metadata to help display `matplotlib` figures legibly (PR #336)

## 1.7 4.8

### 1.7.1 4.8.2

4.8.2 on GitHub

- Fix compatibility issue with `qt eventloop` and `pyzmq 17` (PR #307).

### 1.7.2 4.8.1

4.8.1 on GitHub

- set `zmq.ROUTER_HANDOVER` socket option when available to workaround `libzmq` reconnect bug (PR #300).
- Fix `sdist`s including absolute paths for `kernelspec` files, which prevented installation from `sdist` on Windows (PR #306).

### 1.7.3 4.8.0

4.8.0 on GitHub

- Cleanly shutdown integrated event loops when shutting down the kernel. (PR #290)
- `%gui qt` now uses Qt 5 by default rather than Qt 4, following a similar change in terminal IPython. (PR #293)
- Fix event loop integration for `asyncio` when run with Tornado 5, which uses `asyncio` where available. (PR #296)

## 1.8 4.7

### 1.8.1 4.7.0

#### 4.7.0 on GitHub

- Add event loop integration for `asyncio`.
- Use the new IPython completer API.
- Add support for displaying GIF images (`mimetype image/gif`).
- Allow the kernel to be interrupted without killing the Qt console.
- Fix `is_complete` response with cell magics.
- Clean up encoding of bytes objects.
- Clean up help links to use `https` and improve display titles.
- Clean up `ioloop` handling in preparation for tornado 5.

## 1.9 4.6

### 1.9.1 4.6.1

#### 4.6.1 on GitHub

- Fix eventloop-integration bug preventing Qt windows/widgets from displaying with ipykernel 4.6.0 and IPython 5.2.
- Avoid deprecation warnings about naive datetimes when working with `jupyter_client` 5.0.

### 1.9.2 4.6.0

#### 4.6.0 on GitHub

- Add to API `DisplayPublisher.publish` two new fully backward-compatible keyword-args:
  - `update: bool`
  - `transient: dict`
- Support new `transient` key in `display_data` messages spec for `publish`. For a display data message, `transient` contains data that shouldn't be persisted to files or documents. Add a `display_id` to this transient dict by `display(obj, display_id=...)`
- Add `ipykernel_launcher` module which removes the current working directory from `sys.path` before launching the kernel. This helps to reduce the cases where the kernel won't start because there's a `random.py` (or similar) module in the current working directory.
- Add busy/idle messages on `IOPub` during processing of aborted requests
- Add `active` event loop setting to GUI, which enables the correct response to IPython's `is_event_loop_running_xxx`
- Include IPython kernelspec in wheels to reduce reliance on “native kernel spec” in `jupyter_client`
- Modify `OutStream` to inherit from `TextIOBase` instead of object to improve API support and error reporting

- Fix IPython kernel death messages at start, such as “Kernel Restarting...” and “Kernel appears to have died”, when parent-poller handles PID 1
- Various bugfixes

## 1.10 4.5

### 1.10.1 4.5.2

[4.5.2 on GitHub](#)

- Fix bug when instantiating Comms outside of the IPython kernel (introduced in 4.5.1).

### 1.10.2 4.5.1

[4.5.1 on GitHub](#)

- Add missing `stream` parameter to overridden `getpass()`
- Remove locks from `iopub` thread, which could cause deadlocks during debugging
- Fix regression where `KeyboardInterrupt` was treated as an aborted request, rather than an error
- Allow instantiating Comms outside of the IPython kernel

### 1.10.3 4.5.0

[4.5 on GitHub](#)

- Use `figure.dpi` instead of `savefig.dpi` to set DPI for inline figures
- Support `ipympl` matplotlib backend (requires IPython update as well to fully work)
- Various bugfixes, including fixes for output coming from threads, and `input()` when called with non-string prompts, which `stdlib` allows.

## 1.11 4.4

### 1.11.1 4.4.1

[4.4.1 on GitHub](#)

- Fix circular import of `matplotlib` on Python 2 caused by the inline backend changes in 4.4.0.

### 1.11.2 4.4.0

[4.4.0 on GitHub](#)

- Use `MPLBACKEND` environment variable to tell `matplotlib >= 1.5` use the inline backend by default. This is only done if `MPLBACKEND` is not already set and no backend has been explicitly loaded, so setting `MPLBACKEND=Qt4Agg` or calling `%matplotlib notebook` or `matplotlib.use('Agg')` will take precedence.

- Fixes for logging problems caused by 4.3, where logging could go to the terminal instead of the notebook.
- Add `--sys-prefix` and `--profile` arguments to `ipython kernel install`
- Allow Comm (Widget) messages to be sent from background threads.
- Select inline matplotlib backend by default if `%matplotlib magic` or `matplotlib.use()` are not called explicitly (for matplotlib  $\geq$  1.5).
- Fix some longstanding minor deviations from the message protocol (missing status: ok in a few replies, connect\_reply format).
- Remove calls to NoOpContext from IPython, deprecated in 5.0.

## 1.12 4.3

### 1.12.1 4.3.2

- Use a nonempty dummy session key for inprocess kernels to avoid security warnings.

### 1.12.2 4.3.1

- Fix Windows Python 3.5 incompatibility caused by faulthandler patch in 4.3

### 1.12.3 4.3.0

#### 4.3.0 on GitHub

- Publish all IO in a thread, via `IOPubThread`. This solves the problem of requiring `sys.stdout.flush()` to be called in the notebook to produce output promptly during long-running cells.
- Remove references to outdated IPython guiref in kernel banner.
- Patch faulthandler to use `sys.__stderr__` instead of forwarded `sys.stderr`, which has no fileno when forwarded.
- Deprecate some vestiges of the Big Split: - `ipykernel.find_connection_file()` is deprecated. Use `jupyter_client.find_connection_file()` instead. - Various pieces of code specific to IPython parallel are deprecated in `ipykernel` and moved to `ipyparallel`.

## 1.13 4.2

### 1.13.1 4.2.2

#### 4.2.2 on GitHub

- Don't show interactive debugging info when kernel crashes
- Fix handling of numerical types in `json_clean`
- Testing fixes for output capturing

### 1.13.2 4.2.1

[4.2.1 on GitHub](#)

- Fix default display name back to “Python X” instead of “pythonX”

### 1.13.3 4.2.0

[4.2 on GitHub](#)

- Support sending a full message in initial opening of comms (metadata, buffers were not previously allowed)
- When using `ipython kernel install --name` to install the IPython kernelspec, default display-name to the same value as `--name`.

## 1.14 4.1

### 1.14.1 4.1.1

[4.1.1 on GitHub](#)

- Fix missing `ipykernel.__version__` on Python 2.
- Fix missing `target_name` when opening comms from the frontend.

### 1.14.2 4.1.0

[4.1 on GitHub](#)

- add `ipython kernel install` entrypoint for installing the IPython kernelspec
- provisional implementation of `comm_info` request/reply for msgspec v5.1

## 1.15 4.0

[4.0 on GitHub](#)

4.0 is the first release of `ipykernel` as a standalone package.

## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`