

[Skip to main content](#)

[__Back to top](#)

[__`Ctrl`+`K`](#)

[[!\[conda 25.1.2.dev29 documentation -](#)

[Home\]\(../../../../../_static/conda_logo_full.svg\) \]\(../../../../../index.html\)](#)

* [\[Conda\]\(https://docs.conda.io/projects/conda/\)](#)

* [\[Conda-build\]\(https://docs.conda.io/projects/conda-build\)](#)

* [\[Miniconda\]\(https://docs.anaconda.com/free/miniconda/\)](#)

* [\[conda.org\]\(https://conda.org\)](#)

* [\[__ GitHub\]\(https://github.com/conda/conda "GitHub"\)](#)

* [!\[Element\]\(../../../../../_static/element_logo.svg\)\]\(http://bit.ly/conda-chat-room "Element"\)](#)

* [\[__ Discourse\]\(https://conda.discourse.group/ "Discourse"\)](#)

* [\[Conda\]\(https://docs.conda.io/projects/conda/\)](#)

* [\[Conda-build\]\(https://docs.conda.io/projects/conda-build\)](#)

* [\[Miniconda\]\(https://docs.anaconda.com/free/miniconda/\)](#)

* [\[conda.org\]\(https://conda.org\)](#)

* [\[__ GitHub\]\(https://github.com/conda/conda "GitHub"\)](#)

* [!\[Element\]\(../../../../../_static/element_logo.svg\)\]\(http://bit.ly/conda-chat-room "Element"\)](#)

* [\[__ Discourse\]\(https://conda.discourse.group/ "Discourse"\)](#)

Navigation

- * [User guide](../../user-guide/index.html) __
- * [Getting started with conda](../../user-guide/getting-started.html)
- * [Installing conda](../../user-guide/install/index.html) __
- * [Installing on Windows](../../user-guide/install/windows.html)
- * [Installing on macOS](../../user-guide/install/macos.html)
- * [Installing on Linux](../../user-guide/install/linux.html)
- * [RPM and Debian Repositories for Miniconda](../../user-guide/install/rpm-debian.html)
- * [Tasks](../../user-guide/tasks/index.html) __
- * [Managing conda](../../user-guide/tasks/manage-conda.html)
- * [Managing environments](../../user-guide/tasks/manage-environments.html)
- * [Managing channels](../../user-guide/tasks/manage-channels.html)
- * [Managing packages](../../user-guide/tasks/manage-pkgs.html)
- * [Managing Python](../../user-guide/tasks/manage-python.html)
- * [Managing virtual packages](../../user-guide/tasks/manage-virtual.html)
- * [Creating custom channels](../../user-guide/tasks/create-custom-channels.html)
- * [Creating projects](../../user-guide/tasks/creating-projects.html)
- * [Viewing command-line help](../../user-guide/tasks/view-command-line-help.html)
- * [Configuration](../../user-guide/configuration/index.html) __
 - * [Using the .condarc conda configuration file](../../user-guide/configuration/use-condarc.html)
 - * [Settings](../../user-guide/configuration/settings.html)
 - * [Administering a multi-user conda installation](../../user-guide/configuration/admin-multi-user-install.html)
 - * [Mirroring channels](../../user-guide/configuration/mirroring.html)
 - * [Disabling SSL verification](../../user-guide/configuration/disable-ssl-verification.html)

- * [Using non-standard certificates](../../user-guide/configuration/non-standard-certs.html)
- * [Using Custom Locations for Environment and Package Cache](../../user-guide/configuration/custom-env-and-pkg-locations.html)
- * [Improving interoperability with pip](../../user-guide/configuration/pip-interoperability.html)
- * [Using the free channel](../../user-guide/configuration/free-channel.html)
- * [Concepts](../../user-guide/concepts/index.html) __
- * [Commands](../../user-guide/concepts/conda-commands.html)
- * [Packages](../../user-guide/concepts/packages.html)
- * [Package specification](../../user-guide/concepts/pkg-specs.html)
- * [Package search and install specifications](../../user-guide/concepts/pkg-search.html)
- * [Channels](../../user-guide/concepts/channels.html)
- * [Environments](../../user-guide/concepts/environments.html)
- * [Installing with conda](../../user-guide/concepts/installing-with-conda.html)
- * [Performance](../../user-guide/concepts/conda-performance.html)
- * [Conda for data scientists](../../user-guide/concepts/data-science.html)
- * [Plugins](../../user-guide/concepts/conda-plugins.html)
- * [Troubleshooting](../../user-guide/troubleshooting.html)
- * [Cheatsheet](../../user-guide/cheatsheet.html)
- * [Configuration](../../configuration.html)
- * [Commands](../../commands/index.html) __
- * [``conda clean`](../../commands/clean.html)
- * [``conda compare`](../../commands/compare.html)
- * [``conda config`](../../commands/config.html)
- * [``conda create`](../../commands/create.html)
- * [``conda doctor`](../../commands/doctor.html)
- * [``conda env`](../../commands/env/index.html) __
- * [``conda env config`](../../commands/env/config/index.html) __

- * [`conda env config vars`](../../../../../commands/env/config/vars/index.html) __
- * [`conda env config vars list`](../../../../../commands/env/config/vars/list.html)
- * [`conda env config vars set`](../../../../../commands/env/config/vars/set.html)
- * [`conda env config vars unset`](../../../../../commands/env/config/vars/unset.html)
- * [`conda env create`](../../../../../commands/env/create.html)
- * [`conda env export`](../../../../../commands/env/export.html)
- * [`conda env list`](../../../../../commands/env/list.html)
- * [`conda env remove`](../../../../../commands/env/remove.html)
- * [`conda env update`](../../../../../commands/env/update.html)
- * [`conda info`](../../../../../commands/info.html)
- * [`conda init`](../../../../../commands/init.html)
- * [`conda install`](../../../../../commands/install.html)
- * [`conda list`](../../../../../commands/list.html)
- * [`conda notices`](../../../../../commands/notices.html)
- * [`conda package`](../../../../../commands/package.html)
- * [`conda remove`](../../../../../commands/remove.html)
- * [`conda rename`](../../../../../commands/rename.html)
- * [`conda run`](../../../../../commands/run.html)
- * [`conda search`](../../../../../commands/search.html)
- * [`conda update`](../../../../../commands/update.html)
- * [Release notes](../../../../../release-notes.html)
- * [Glossary](../../../../../glossary.html)
- * [Developer guide](../../../../../index.html) __
- * [Architecture](../../../../../architecture.html)
- * [Contributing to conda](../../../../../contributing.html)
- * [Development Environment](../../../../../development-environment.html)
- * [Deep dives](../../../../../deep-dives/index.html) __

- * [`conda install``](../../deep-dives/install.html)
- * [`conda init`` and `conda activate``](../../deep-dives/activation.html)
- * [`conda config`` and context](../../deep-dives/context.html)
- * [Solvers](../../deep-dives/solvers.html)
- * [Logging](../../deep-dives/logging.html)
- * [Writing Tests](../../writing-tests/index.html) __
- * [Integration Tests](../../writing-tests/integration-tests.html)
- * [Deprecations](../../deprecations.html)
- * [Releasing](../../releasing.html)
- * [Plugins](../../plugins/index.html) __
 - * [Auth Handlers](../../plugins/auth_handlers.html)
 - * [Health Checks](../../plugins/health_checks.html)
 - * [Request Headers](../../plugins/request_headers.html)
 - * [Post-commands](../../plugins/post_commands.html)
 - * [Pre-commands](../../plugins/pre_commands.html)
 - * [Reporter Backends](../../plugins/reporter_backends.html)
 - * [Settings](../../plugins/settings.html)
 - * [Solvers](../../plugins/solvers.html)
 - * [Subcommands](../../plugins/subcommands.html)
 - * [Virtual Packages](../../plugins/virtual_packages.html)
- * [Specifications](../../specs/index.html) __
 - * [Technical specification: solver state](../../specs/solver-state.html)
- * [API](../../api.html) __
 - * [`conda``](./index.html) __
 - * [`__main__``](./__main__/index.html)
 - * [`_vendor``](./_vendor/index.html) __
 - * [`frozendict``](./_vendor/frozendict/index.html)

- * [`_version``](../_version/index.html)
- * [`activate``](../activate/index.html)
- * [`api``](../api/index.html)
- * [`auxlib``](../auxlib/index.html) __
 - * [`collection``](../auxlib/collection/index.html)
 - * [`compat``](../auxlib/compat/index.html)
 - * [`decorators``](../auxlib/decorators/index.html)
 - * [`entity``](../auxlib/entity/index.html)
 - * [`exceptions``](../auxlib/exceptions/index.html)
 - * [`ish``](../auxlib/ish/index.html)
 - * [`logz``](../auxlib/logz/index.html)
 - * [`type_coercion``](../auxlib/type_coercion/index.html)
- * [`base``](../base/index.html) __
 - * [`constants``](../base/constants/index.html)
 - * [`context``](../base/context/index.html)
- * [`cli``](../cli/index.html) __
 - * [`actions``](../cli/actions/index.html)
 - * [`common``](../cli/common/index.html)
 - * [`conda_argparse``](../cli/conda_argparse/index.html)
 - * [`find_commands``](../cli/find_commands/index.html)
 - * [`helpers``](../cli/helpers/index.html)
 - * [`install``](../cli/install/index.html)
 - * [`main``](../cli/main/index.html)
 - * [`main_clean``](../cli/main_clean/index.html)
 - * [`main_commands``](../cli/main_commands/index.html)
 - * [`main_compare``](../cli/main_compare/index.html)
 - * [`main_config``](../cli/main_config/index.html)

- * [main_create](../cli/main_create/index.html)
- * [main_env](../cli/main_env/index.html)
- * [main_env_config](../cli/main_env_config/index.html)
- * [main_env_create](../cli/main_env_create/index.html)
- * [main_env_export](../cli/main_env_export/index.html)
- * [main_env_list](../cli/main_env_list/index.html)
- * [main_env_remove](../cli/main_env_remove/index.html)
- * [main_env_update](../cli/main_env_update/index.html)
- * [main_env_vars](../cli/main_env_vars/index.html)
- * [main_export](../cli/main_export/index.html)
- * [main_info](../cli/main_info/index.html)
- * [main_init](../cli/main_init/index.html)
- * [main_install](../cli/main_install/index.html)
- * [main_list](../cli/main_list/index.html)
- * [main_mock_activate](../cli/main_mock_activate/index.html)
- * [main_mock_deactivate](../cli/main_mock_deactivate/index.html)
- * [main_notices](../cli/main_notices/index.html)
- * [main_package](../cli/main_package/index.html)
- * [main_pip](../cli/main_pip/index.html)
- * [main_remove](../cli/main_remove/index.html)
- * [main_rename](../cli/main_rename/index.html)
- * [main_run](../cli/main_run/index.html)
- * [main_search](../cli/main_search/index.html)
- * [main_update](../cli/main_update/index.html)
- * [python_api](../cli/python_api/index.html)
- * [common](../common/index.html) __
- * [_logic](../common/_logic/index.html)

- * [`_os``](../common/_os/index.html) __
- * [`linux``](../common/_os/linux/index.html)
- * [`osx``](../common/_os/osx/index.html)
- * [`unix``](../common/_os/unix/index.html)
- * [`windows``](../common/_os/windows/index.html)
- * [`compat``](../common/compat/index.html)
- * [`configuration``](../common/configuration/index.html)
- * [`constants``](../common/constants/index.html)
- * [`disk``](../common/disk/index.html)
- * [`io``](../common/io/index.html)
- * [`iterators``](../common/iterators/index.html)
- * [`logic``](../common/logic/index.html)
- * [`path``](../common/path/index.html) __
- * [`_cygpath``](../common/path/_cygpath/index.html)
- * [`directories``](../common/path/directories/index.html)
- * [`python``](../common/path/python/index.html)
- * [`windows``](../common/path/windows/index.html)
- * [`pkg_formats``](../common/pkg_formats/index.html) __
- * [`python``](../common/pkg_formats/python/index.html)
- * [`serialize``](../common/serialize/index.html)
- * [`signals``](../common/signals/index.html)
- * [`toposort``](../common/toposort/index.html)
- * [`url``](../common/url/index.html)
- * [`core``](../core/index.html) __
- * [`envs_manager``](../core/envs_manager/index.html)
- * [`index``](../core/index/index.html)
- * [`initialize``](../core/initialize/index.html)

- * [`link``](../core/link/index.html)
- * [`package_cache_data``](../core/package_cache_data/index.html)
- * [`path_actions``](../core/path_actions/index.html)
- * [`portability``](../core/portability/index.html)
- * [`prefix_data``](../core/prefix_data/index.html)
- * [`solve``](../core/solve/index.html)
- * [`subdir_data``](../core/subdir_data/index.html)
- * [`deprecations``](../deprecations/index.html)
- * [`env``](../env/index.html) __
- * [`env``](../env/env/index.html)
- * [`installers``](../env/installers/index.html) __
- * [`base``](../env/installers/base/index.html)
- * [`conda``](../env/installers/conda/index.html)
- * [`pip``](../env/installers/pip/index.html)
- * [`pip_util``](../env/pip_util/index.html)
- * [`specs``](../env/specs/index.html) __
- * [`binstar``](../env/specs/binstar/index.html)
- * [`requirements``](../env/specs/requirements/index.html)
- * [`yaml_file``](../env/specs/yaml_file/index.html)
- * [`exception_handler``](../exception_handler/index.html)
- * `exceptions``
- * [`exports``](../exports/index.html)
- * [`gateways``](../gateways/index.html) __
- * [`anaconda_client``](../gateways/anaconda_client/index.html)
- * [`connection``](../gateways/connection/index.html) __
- * [`adapters``](../gateways/connection/adapters/index.html) __
- * [`ftp``](../gateways/connection/adapters/ftp/index.html)

- * [`http`](../gateways/connection/adapters/http/index.html)
- * [`localfs`](../gateways/connection/adapters/localfs/index.html)
- * [`s3`](../gateways/connection/adapters/s3/index.html)
- * [`download`](../gateways/connection/download/index.html)
- * [`session`](../gateways/connection/session/index.html)
- * [`disk`](../gateways/disk/index.html) ____
- * [`create`](../gateways/disk/create/index.html)
- * [`delete`](../gateways/disk/delete/index.html)
- * [`link`](../gateways/disk/link/index.html)
- * [`lock`](../gateways/disk/lock/index.html)
- * [`permissions`](../gateways/disk/permissions/index.html)
- * [`read`](../gateways/disk/read/index.html)
- * [`test`](../gateways/disk/test/index.html)
- * [`update`](../gateways/disk/update/index.html)
- * [`logging`](../gateways/logging/index.html)
- * [`repodata`](../gateways/repodata/index.html) ____
- * [`jlap`](../gateways/repodata/jlap/index.html) ____
- * [`core`](../gateways/repodata/jlap/core/index.html)
- * [`fetch`](../gateways/repodata/jlap/fetch/index.html)
- * [`interface`](../gateways/repodata/jlap/interface/index.html)
- * [`lock`](../gateways/repodata/lock/index.html)
- * [`subprocess`](../gateways/subprocess/index.html)
- * [`history`](../history/index.html)
- * [`instructions`](../instructions/index.html)
- * [`misc`](../misc/index.html)
- * [`models`](../models/index.html) ____
- * [`channel`](../models/channel/index.html)

- * [`dist``](../models/dist/index.html)
- * [`enums``](../models/enums/index.html)
- * [`leased_path_entry``](../models/leased_path_entry/index.html)
- * [`match_spec``](../models/match_spec/index.html)
- * [`package_info``](../models/package_info/index.html)
- * [`prefix_graph``](../models/prefix_graph/index.html)
- * [`records``](../models/records/index.html)
- * [`version``](../models/version/index.html)
- * [`notices``](../notices/index.html) __
- * [`cache``](../notices/cache/index.html)
- * [`core``](../notices/core/index.html)
- * [`fetch``](../notices/fetch/index.html)
- * [`types``](../notices/types/index.html)
- * [`views``](../notices/views/index.html)
- * [`plan``](../plan/index.html)
- * [`plugins``](../plugins/index.html) __
- * [`hookspec``](../plugins/hookspec/index.html)
- * [`manager``](../plugins/manager/index.html)
- * [`post_solves``](../plugins/post_solves/index.html) __
- * [`signature_verification``](../plugins/post_solves/signature_verification/index.html)
- * [`reporter_backends``](../plugins/reporter_backends/index.html) __
- * [`console``](../plugins/reporter_backends/console/index.html)
- * [`json``](../plugins/reporter_backends/json/index.html)
- * [`solvers``](../plugins/solvers/index.html)
- * [`subcommands``](../plugins/subcommands/index.html) __
- * [`doctor``](../plugins/subcommands/doctor/index.html) __
- * [`health_checks``](../plugins/subcommands/doctor/health_checks/index.html)

- * [`types`](../plugins/types/index.html)
- * [`virtual_packages`](../plugins/virtual_packages/index.html) ____
- * [`archspeg`](../plugins/virtual_packages/archspeg/index.html)
- * [`conda`](../plugins/virtual_packages/conda/index.html)
- * [`cuda`](../plugins/virtual_packages/cuda/index.html)
- * [`freebsd`](../plugins/virtual_packages/freebsd/index.html)
- * [`linux`](../plugins/virtual_packages/linux/index.html)
- * [`osx`](../plugins/virtual_packages/osx/index.html)
- * [`windows`](../plugins/virtual_packages/windows/index.html)
- * [`reporters`](../reporters/index.html)
- * [`resolve`](../resolve/index.html)
- * [`testing`](../testing/index.html) ____
- * [`cases`](../testing/cases/index.html)
- * [`fixtures`](../testing/fixtures/index.html)
- * [`gateways`](../testing/gateways/index.html) ____
- * [`fixtures`](../testing/gateways/fixtures/index.html)
- * [`helpers`](../testing/helpers/index.html)
- * [`integration`](../testing/integration/index.html)
- * [`notices`](../testing/notices/index.html) ____
- * [`fixtures`](../testing/notices/fixtures/index.html)
- * [`helpers`](../testing/notices/helpers/index.html)
- * [`solver_helpers`](../testing/solver_helpers/index.html)
- * [`trust`](../trust/index.html) ____
- * [`constants`](../trust/constants/index.html)
- * [`signature_verification`](../trust/signature_verification/index.html)
- * [`utils`](../utils/index.html)
- * [`conda_env`](.././conda_env/index.html) ____

* [`cli``](../conda_env/cli/index.html)

* [`installers``](../conda_env/installers/index.html)

* [`__`](../../../index.html)

* [Developer guide](../../index.html)

* `__`

* [`conda``](../index.html)

*

`exceptions``

Conda exceptions.

Functions#

`maybe_raise``(error, context) |

---|---

`print_conda_exception``(exc_val[, exc_tb]) |

`_format_exc`` ([exc_val, exc_tb]) |

`_exception _ResolvePackageNotFound(_bad_deps_)` #

Bases: [`conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _LockError(_message_)`#

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _ArgumentError(_message_ , ** kwargs_)`#

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`return_code _ = 2_`#

`_exception _Help(_message_ , _caused_by =None_, ** kwargs_)#`

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _ActivateHelp#`

Bases: ``Help``

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _DeactivateHelp#`

Bases: ``Help``

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _GenericHelp(_command_)`#

Bases: ``Help``

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _CondaSignalInterrupt(_signum_)`#

Bases: `[`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")`

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _TooManyArgumentsError(_expected_ , _received_ ,
_offending_arguments_ , _optional_message = "", _* args_)`#

Bases: ``ArgumentError``

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _ClobberError(_message_ , _path_conflict_ , _** kwargs_)`#

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`__repr__()`#

Return `repr(self)`.

`_exception _BasicClobberError(_source_path_ , _target_path_ , _context_)`#

Bases: `ClobberError`

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _KnownPackageClobberError(_target_path_ ,  
_colliding_dist_being_linked_ , _colliding_linked_dist_ , _context_)#
```

Bases: `ClobberError`

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _UnknownPackageClobberError(_target_path_ ,  
_colliding_dist_being_linked_ , _context_)#
```

Bases: `ClobberError`

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _SharedLinkPathClobberError(_target_path_ ,  
_incompatible_package_dists_ , _context_)#
```

Bases: ``ClobberError``

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

```
_exception _CommandNotFoundError(_command_)#
```

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

```
_exception _PathNotFoundError(_path_)#
```

Bases: [``conda.CondaError``](../index.html#conda.CondaError

"conda.CondaError"),

[`OSError`](https://docs.python.org/3/library/exceptions.html#OSError "(in Python v3.13\))")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _DirectoryNotFoundError(_path_)`#

Bases: [`conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _EnvironmentLocationNotFound(_location_)`#

Bases: [`conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _EnvironmentNameNotFound(_environment_name_)#`

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _NoBaseEnvironmentError#`

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _DirectoryNotACondaEnvironmentError(_target_directory_)#`

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _CondaEnvironmentError(_message_ , _* args_)#
```

Bases: [`conda.CondaError``](../index.html#conda.CondaError

"conda.CondaError"),

[`OSError``](https://docs.python.org/3/library/exceptions.html#OSError "(in

Python v3.13\))

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _DryRunExit#
```

Bases: [`conda.CondaExitZero``](../index.html#conda.CondaExitZero

"conda.CondaExitZero")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _CondaSystemExit(_ * args_)`#

Bases: [``conda.CondaExitZero``](../index.html#conda.CondaExitZero

`"conda.CondaExitZero")`,

[``SystemExit``](https://docs.python.org/3/library/exceptions.html#SystemExit

`"\n(in Python v3.13\n)")`

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _PaddingError(_dist_ , _placeholder_ , _placeholder_length_)`#

Bases: [``conda.CondaError``](../index.html#conda.CondaError `"conda.CondaError"`)

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _LinkError(_message_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _CondaOSError(_message_ , ** kwargs_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError"),
[`OSError`](https://docs.python.org/3/library/exceptions.html#OSError "(in Python v3.13\\)")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _ProxyError`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _CondaIOError(_message_ , _* args_)#
```

Bases: [`conda.CondaError`](../index.html#conda.CondaError`

`"conda.CondaError"),`

[`OSError`](https://docs.python.org/3/library/exceptions.html#OSError "(in`

`Python v3.13\))`

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _CondaFileIOError(_filepath_ , _message_ , _* args_)#
```

Bases: ``CondaIOError``

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _CondaKeyError(_key_ , _message_ , _* args_)#
```

Bases: [`conda.CondaError`](../index.html#conda.CondaError
"conda.CondaError"),
[`KeyError`](https://docs.python.org/3/library/exceptions.html#KeyError "(in
Python v3.13\))

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _ChannelError(_message_ , _caused_by =None_, ** kwargs_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _ChannelNotAllowed(_channel_)`#

Bases: `ChannelError`

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

warning _ = 'Channel not included in allowlist' _#

_exception _ChannelDenied(_channel_)#

Bases: `ChannelNotAllowed`

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

warning _ = 'Channel included in denylist' _#

_exception _UnavailableInvalidChannel(_channel_ , _status_code_ , _response :
requests.models.Response | [None])(<https://docs.python.org/3/library/constants.html#None> "\n(in
Python v3.13\`) = None_)#

Bases: ``ChannelError``

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`status_code` `_`: `[str](https://docs.python.org/3/library/stdtypes.html#str "(in Python v3.13\))" |`
`[int](https://docs.python.org/3/library/functions.html#int "(in Python v3.13\))_#`

`_exception` `_OperationNotAllowed(_message_)`#

Bases: `[`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")`

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception` `_CondaImportError(_message_)`#

Bases: `[`conda.CondaError`](../index.html#conda.CondaError`
`"conda.CondaError"),`

[`ImportError`](https://docs.python.org/3/library/exceptions.html#ImportError

"\n(in Python v3.13\n)")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

_exception _ParseError(_message_)#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

_exception _CouldntParseError(_reason_)#

Bases: `ParseError`

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _ChecksumMismatchError(_url_ , _target_full_path_ , _checksum_type_  
, _expected_checksum_ , _actual_checksum_ , _partial_download =False_)#
```

Bases: [`conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

```
_exception _PackageNotInstalledError(_prefix_ , _package_name_)#
```

Bases: [`conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

```
_exception _CondaHTTPError(_message_ , _url_ , _status_code_ , _reason_  
, _elapsed_time_ , _response =None_ , _caused_by =None_)#
```

Bases: [`conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _CondaSSLError(_message_ , _caused_by =None_, ** kwargs_)#
```

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _AuthenticationError(_message_ , _caused_by =None_, ** kwargs_)#
```

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _PackagesNotFoundError(_packages_ , _channel_urls =())#
```

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _UnsatisfiableError(_bad_deps_ , _chains =True_, _strict =False_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

An exception to report unsatisfiable dependencies.

Parameters:

* **`bad_deps`** -- a list of tuples of objects (likely `MatchSpecs`).

* **`chains`** -- (optional) if `True`, the tuples are interpreted as chains of dependencies, from top level to bottom. If `False`, the tuples are interpreted as simple lists of conflicting specs.

Returns:

Raises an exception with a formatted message detailing the unsatisfiable specifications.

Initialize self. See help(type(self)) for accurate signature.

`_format_chain_str(_bad_deps_)`#

`_exception _RemoveError(_message_)`#

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _DisallowedPackageError(_package_ref_ , _** kwargs_)`#

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _SpecsConfigurationConflictError(_requested_specs_ , _pinned_specs_  
, _prefix_)#
```

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _CondaIndexError(_message_)#
```

Bases: [`conda.CondaError`](../index.html#conda.CondaError

"conda.CondaError"),

[`IndexError`](https://docs.python.org/3/library/exceptions.html#IndexError

"\ (in Python v3.13\)"

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _CondaValueError(_message_ , _* args_ , _** kwargs_)#`

Bases: [``conda.CondaError``](../index.html#conda.CondaError

`"conda.CondaError")`,

[``ValueError``](https://docs.python.org/3/library/exceptions.html#ValueError

`"\n(in Python v3.13\)"`)

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _CyclicalDependencyError(_packages_with_cycles_ , _** kwargs_)#`

Bases: [``conda.CondaError``](../index.html#conda.CondaError

`"conda.CondaError")`,

[``ValueError``](https://docs.python.org/3/library/exceptions.html#ValueError

`"\n(in Python v3.13\)"`)

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

```
_exception _CorruptedEnvironmentError(_environment_location_ ,  
_corrupted_file_ , _** kwargs_)#
```

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

```
_exception _CondaHistoryError(_message_)#
```

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

```
_exception _CondaUpgradeError(_message_)#
```

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _CondaVerificationError(_message_)`#

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _SafetyError(_message_)`#

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _CondaMemoryError(_caused_by_ , _** kwargs_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError

`conda.CondaError`),

[`MemoryError`](https://docs.python.org/3/library/exceptions.html#MemoryError

`"(in Python v3.13\)"`)

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _NotWritableError(_path_ , _errno_ , ** kwargs_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError

`conda.CondaError`),

[`OSError`](https://docs.python.org/3/library/exceptions.html#OSError `"(in`

`Python v3.13\)"`)

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _NoWritableEnvsDirError(_envs_dirs_ , ** kwargs_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _NoWritablePkgsDirError(_pkgs_dirs_ , ** kwargs_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _EnvironmentNotWritableError(_environment_location_ , ** kwargs_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _CondaDependencyError(_message_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _BinaryPrefixReplacementError(_path_ , _placeholder_ , _new_prefix_ , _original_data_length_ , _new_data_length_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _InvalidSpec(_message :`
`[str](https://docs.python.org/3/library/stdtypes.html#str "(in Python`
`v3.13\))", _** kwargs_)`#

Bases: [`conda.CondaError`](../index.html#conda.CondaError

"conda.CondaError"),

[`ValueError`](https://docs.python.org/3/library/exceptions.html#ValueError

"\ (in Python v3.13\)"

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _InvalidVersionSpec(_invalid_spec :`

`[str](https://docs.python.org/3/library/stdtypes.html#str "\ (in Python
v3.13\)")_, _details :`

`[str](https://docs.python.org/3/library/stdtypes.html#str "\ (in Python
v3.13\)")_)#`

Bases: ``InvalidSpec``

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

`_exception _InvalidMatchSpec(_invalid_spec :`

`[str](https://docs.python.org/3/library/stdtypes.html#str "\ (in Python
v3.13\)")_, _details :`

`[str](https://docs.python.org/3/library/stdtypes.html#str "\ (in Python
v3.13\)")_)#`

Bases: ``InvalidSpec``

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _EncodingError(_caused_by_ , _** kwargs_)`#

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _NoSpaceLeftError(_caused_by_ , _** kwargs_)`#

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _CondaEnvException(_message_ , _* args_, _** kwargs_)#
```

Bases: [`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _EnvironmentFileNotFound(_filename_ , _* args_, _** kwargs_)#
```

Bases: `CondaEnvException`

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

```
_exception _EnvironmentFileExtensionNotValid(_filename_ , _* args_, _**  
kwargs_)#
```

Bases: ``CondaEnvException``

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _EnvironmentFileEmpty(_filename_ , _* args_ , _** kwargs_)#`

Bases: ``CondaEnvException``

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _EnvironmentFileNotDownloaded(_username_ , _packagename_ , _*
args_ , _** kwargs_)#`

Bases: `[`conda.CondaError`](../index.html#conda.CondaError "conda.CondaError")`

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _SpecNotFound(_msg_ , _* args_, _** kwargs_)#`

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`_exception _PluginError(_message_ , _caused_by =None_, _** kwargs_)#`

Bases: [``conda.CondaError``](../index.html#conda.CondaError "conda.CondaError")

Common base class for all non-exit exceptions.

Initialize self. See `help(type(self))` for accurate signature.

`maybe_raise(_error_ , _context_)#`

`print_conda_exception(_exc_val_ , _exc_tb =None_)#`

`_format_exc(_exc_val=None, _exc_tb=None)#`

`_exception _InvalidInstaller(_name_)#`

Bases:

[`Exception`](https://docs.python.org/3/library/exceptions.html#Exception)

"\ (in Python v3.13\)"

Common base class for all non-exit exceptions.

Initialize self. See help(type(self)) for accurate signature.

__On this page

* Functions

* `ResolvePackageNotFound`

* `LockError`

* `ArgumentError`

* `ArgumentError.return_code`

* `Help`

* `ActivateHelp`

* `DeactivateHelp`

- * ``GenericHelp``
- * ``CondaSignalInterrupt``
- * ``TooManyArgumentsError``
- * ``ClobberError``
 - * ``ClobberError.__repr__()``
- * ``BasicClobberError``
- * ``KnownPackageClobberError``
- * ``UnknownPackageClobberError``
- * ``SharedLinkPathClobberError``
- * ``CommandNotFoundError``
- * ``PathNotFoundError``
- * ``DirectoryNotFoundError``
- * ``EnvironmentLocationNotFound``
- * ``EnvironmentNameNotFound``
- * ``NoBaseEnvironmentError``
- * ``DirectoryNotACondaEnvironmentError``
- * ``CondaEnvironmentError``
- * ``DryRunExit``
- * ``CondaSystemExit``
- * ``PaddingError``
- * ``LinkError``
- * ``CondaOSError``
- * ``ProxyError``
- * ``CondaIOError``
- * ``CondaFileIOError``
- * ``CondaKeyError``
- * ``ChannelError``

- * ``ChannelNotAllowed``
- * ``ChannelNotAllowed.warning``
- * ``ChannelDenied``
- * ``ChannelDenied.warning``
- * ``UnavailableInvalidChannel``
- * ``UnavailableInvalidChannel.status_code``
- * ``OperationNotAllowed``
- * ``CondaImportError``
- * ``ParseError``
- * ``CouldntParseError``
- * ``ChecksumMismatchError``
- * ``PackageNotInstalledError``
- * ``CondaHTTPError``
- * ``CondaSSLError``
- * ``AuthenticationError``
- * ``PackagesNotFoundError``
- * ``UnsatisfiableError``
- * ``UnsatisfiableError._format_chain_str()``
- * ``RemoveError``
- * ``DisallowedPackageError``
- * ``SpecsConfigurationConflictError``
- * ``CondaIndexError``
- * ``CondaValueError``
- * ``CyclicalDependencyError``
- * ``CorruptedEnvironmentError``
- * ``CondaHistoryError``
- * ``CondaUpgradeError``

- * ``CondaVerificationError``
- * ``SafetyError``
- * ``CondaMemoryError``
- * ``NotWritableError``
- * ``NoWritableEnvsDirError``
- * ``NoWritablePkgsDirError``
- * ``EnvironmentNotWritableError``
- * ``CondaDependencyError``
- * ``BinaryPrefixReplacementError``
- * ``InvalidSpec``
- * ``InvalidVersionSpec``
- * ``InvalidMatchSpec``
- * ``EncodingError``
- * ``NoSpaceLeftError``
- * ``CondaEnvException``
- * ``EnvironmentFileNotFound``
- * ``EnvironmentFileExtensionNotValid``
- * ``EnvironmentFileEmpty``
- * ``EnvironmentFileNotDownloaded``
- * ``SpecNotFound``
- * ``PluginError``
- * ``maybe_raise()``
- * ``print_conda_exception()``
- * ``_format_exc()``
- * ``InvalidInstaller``

[guide/api/conda/exceptions/index.rst](#))

[[__Show Source](#)](../../../../_sources/dev-guide/api/conda/exceptions/index.rst.txt)

© Copyright 2017, Anaconda, Inc.

Created using [\[Sphinx\]](https://www.sphinx-doc.org/) 7.4.7.

[[Analytics Dashboard](https://docs-conda-io.goatcounter.com)](<https://docs-conda-io.goatcounter.com> "Analytics Dashboard")

Built with the [\[PyData Sphinx Theme\]](https://pydata-sphinx-theme.readthedocs.io/en/stable/index.html) 0.15.4.