

[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

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

Cache](custom-env-and-pkg-locations.html)

- * [Improving interoperability with pip](pip-interoperability.html)

- * [Using the free channel](free-channel.html)

- * [Concepts](../concepts/index.html) __

- * [Commands](../concepts/conda-commands.html)

- * [Packages](../concepts/packages.html)

- * [Package specification](../concepts/pkg-specs.html)

- * [Package search and install specifications](../concepts/pkg-search.html)

- * [Channels](../concepts/channels.html)

- * [Environments](../concepts/environments.html)

- * [Installing with conda](../concepts/installing-with-conda.html)

- * [Performance](../concepts/conda-performance.html)

- * [Conda for data scientists](../concepts/data-science.html)

- * [Plugins](../concepts/conda-plugins.html)

- * [Troubleshooting](../troubleshooting.html)

- * [Cheatsheet](../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](../dev-guide/index.html) __
- * [Architecture](../dev-guide/architecture.html)
- * [Contributing to conda](../dev-guide/contributing.html)
- * [Development Environment](../dev-guide/development-environment.html)
- * [Deep dives](../dev-guide/deep-dives/index.html) __
- * [`conda install`](../dev-guide/deep-dives/install.html)
- * [`conda init` and `conda activate`](../dev-guide/deep-dives/activation.html)

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

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

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

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

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

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

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

health_checks`](../../dev-guide/api/conda/plugins/subcommands/doctor/health_checks/index.html)

* [`types`](../../dev-guide/api/conda/plugins/types/index.html)

* [`virtual_packages`](../../dev-guide/api/conda/plugins/virtual_packages/index.html) __

* [` archspec`](../../dev-guide/api/conda/plugins/virtual_packages/archspec/index.html)

* [`conda`](../../dev-guide/api/conda/plugins/virtual_packages/conda/index.html)

* [`cuda`](../../dev-guide/api/conda/plugins/virtual_packages/cuda/index.html)

* [`freebsd`](../../dev-guide/api/conda/plugins/virtual_packages/freebsd/index.html)

* [`linux`](../../dev-guide/api/conda/plugins/virtual_packages/linux/index.html)

* [`osx`](../../dev-guide/api/conda/plugins/virtual_packages/osx/index.html)

* [`windows`](../../dev-guide/api/conda/plugins/virtual_packages/windows/index.html)

* [`reporters`](../../dev-guide/api/conda/reporters/index.html)

* [`resolve`](../../dev-guide/api/conda/resolve/index.html)

* [`testing`](../../dev-guide/api/conda/testing/index.html) __

* [` cases`](../../dev-guide/api/conda/testing/cases/index.html)

* [` fixtures`](../../dev-guide/api/conda/testing/fixtures/index.html)

* [` gateways`](../../dev-guide/api/conda/testing/gateways/index.html) __

* [` fixtures`](../../dev-guide/api/conda/testing/gateways/fixtures/index.html)

* [`helpers`](../../dev-guide/api/conda/testing/helpers/index.html)

* [`integration`](../../dev-guide/api/conda/testing/integration/index.html)

* [`notices`](../../dev-guide/api/conda/testing/notices/index.html) __

* [` fixtures`](../../dev-guide/api/conda/testing/notices/fixtures/index.html)

* [`helpers`](../../dev-guide/api/conda/testing/notices/helpers/index.html)

* [`solver_helpers`](../../dev-guide/api/conda/testing/solver_helpers/index.html)

* [`trust`](../../dev-guide/api/conda/trust/index.html) __

* [` constants`](../../dev-guide/api/conda/trust/constants/index.html)

* [`signature_verification`](../../dev-guide/api/conda/trust/signature_verification/index.html)

* [`utils`](../../dev-guide/api/conda/utils/index.html)

- * [`conda_env``](../../dev-guide/api/conda_env/index.html) __
- * [`cli``](../../dev-guide/api/conda_env/cli/index.html)
- * [`installers``](../../dev-guide/api/conda_env/installers/index.html)

- * [__](../../index.html)
- * [User guide](../index.html)
- * [Configuration](index.html)
- * Settings

Settings#

This page contains an overview of many important settings available in conda with examples where possible.

General configuration#

``channels``: Channel locations#

Listing channel locations in the ``.condarc`` file overrides conda defaults, causing conda to search only the channels listed there in the order given.

Use ``defaults`` to automatically include all default channels. Non-URL channels are interpreted as Anaconda.org user or organization names. You can change this by modifying the ``channel_alias`` as described in `channel_alias`: Set a channel alias. The default is just ``defaults``.

Example:

channels:

- <anaconda_dot_org_username>
- <http://some.custom/channel>
- <file:///some/local/directory>
- defaults

To select channels for a single environment, put a `.condarc` file in the root directory of that environment (or use the `--env` option when using `conda config`).

Example: If you have installed Miniconda with Python 3 in your home directory and the environment is named "flowers", the path may be:

```
~/miniconda3/envs/flowers/.condarc
```

```
### `default_channels`: Default channels#
```

Normally, the defaults channel points to several channels at the

[repo.anaconda.com](<https://repo.anaconda.com/>) repository, but if

``default_channels`` is defined, it sets the new list of default channels. This

is especially useful for airgapped and enterprise installations.

To ensure that all users only pull packages from an on-premises repository, an administrator can set both channel alias and ``default_channels``.

`default_channels:`

- `http://some.custom/channel`
- `file:///some/local/directory`

``channel_settings``: Extra settings for individual channels#

Added in version 23.3.0.

With ``channel_settings``, it is possible to add extra configuration options for individual channels. This is currently used to register additional authentication handlers for conda via the `[Auth Handlers](../../dev-guide/plugins/auth_handlers.html)` plugin hook, but may also accommodate more use cases in the future.

Here is an example of how it may be defined provided there was an available authentication handler called, "test-auth-handler" registered via the aforementioned plugin hook:

channel_settings:

- channel: https://some.custom/channel
auth: test-auth-handler
user: my-user-account
- channel: https://some.base-url-prefix/*
auth: another-auth-handler

Note

Each entry in `channel_settings`` needs to define the `channel`` attribute so that the configuration knows which channel these settings are associated with. The `channel`` attribute may specify a glob-like URL pattern for matching. Note that in this case, the HTTP schema must match exactly to the channel URL, so a pattern like ``*`` is not valid.

`allowlist_channels`` and `denylist_channels``: Allow or deny specific channels#

Added in version 24.9.0: The `denylist_channels`` setting was introduced in conda 24.9.0 complementing the existing `allowlist_channels`` setting.

With `allowlist_channels`` and `denylist_channels``, you can allow or deny specific channels from being used in conda operations. This is useful for restricting the channels that conda can access, especially in enterprise or multi-user environments.

The denylist takes precedence over the allowlist. If a channel is in both lists, it is denied.

****Examples:****

An example which allows the `defaults` and `conda-forge` channels with the `allowlist_channels` setting is:

allowlist_channels:

- defaults
- conda-forge

An example which denies the `conda-forge` channel with the `denylist_channels` setting is:

denylist_channels:

- conda-forge

An example which explicitly allows the `defaults` channel but denies the `conda-forge` channel by using both the `allowlist_channels` and

`denylist_channels` settings is:

allowlist_channels:

- defaults

denylist_channels:

- conda-forge

An example to show that channels are automatically normalized based on their base URLs, so you can use either the full channel URL or just the base URL:

allowlist_channels:

- defaults

denylist_channels:

- <https://conda.anaconda.org/conda-forge/linux-64>

An example that denies using `defaults` (which maps to the default_channels)

configuration option:

denylist_channels:

- defaults

Note

The defaults channel points to a list of channels at the `repo.anaconda.com` repository by default.

An example to explicitly deny the channels that are hosted on ``repo.anaconda.com``:

`denylist_channels:`

- `https://repo.anaconda.com/pkgs/main`
- `https://repo.anaconda.com/pkgs/r`
- `https://repo.anaconda.com/pkgs/msys2`

`### `auto_update_conda`: Update conda automatically#`

When ``True``, conda updates itself any time a user updates or installs a package in the root environment. When ``False``, conda updates itself only if the user manually issues a ``conda update`` command. The default is ``True``.

****Example:****

auto_update_conda: False

`always_yes`: Always yes#

Choose the `yes` option whenever asked to proceed, such as when installing.

Same as using the `--yes` flag at the command line. The default is `False`.

Example:

always_yes: True

`show_channel_urls`: Show channel URLs#

Show channel URLs in `conda list` and when displaying what is going to be downloaded. The default is `False`.

Example:

show_channel_urls: True

`change_ps1`: Change command prompt#

When using `conda activate`, change the command prompt from `\$PS1` to include the activated environment. The default is `True`.

Example:

change_ps1: False

`add_pip_as_python_dependency`: Add pip as Python dependency#

Add pip, wheel, and setuptools as dependencies of Python. This ensures that pip, wheel, and setuptools are always installed any time Python is installed. The default is `True`.

Example:

add_pip_as_python_dependency: False

`use_pip`: Use pip#

Use pip when listing packages with `conda list`. This does not affect any conda command or functionality other than the output of the command `conda list`. The default is `True`.

Example:

```
use_pip: False
```

`proxy_servers`: Configure conda for use behind a proxy server#

By default, proxy settings are pulled from the HTTP_PROXY and HTTPS_PROXY environment variables or the system. Setting them here overrides that default:

```
proxy_servers:
```

```
http: http://user:{{email protected}}/cdn-cgi/l/email-protection):8080
```

```
https: http://user:{{email protected}}/cdn-cgi/l/email-protection):8080
```

Mixing HTTPS and HTTP

The protocol in the URL (either ``http://`` or ``https://``) should match the actual protocol of your proxy server. The keys ``http`` and ``https`` in the above example merely indicate the type of traffic to route, not the protocol of the proxy server itself. Ensure that both keys use the correct protocol based on your proxy server's configuration.

To give a proxy for a specific scheme and host, use the ``scheme://hostname`` form for the key. This matches for any request to the given scheme and exact host name:

`proxy_servers:`

```
'http://10.20.1.128': 'http://10.10.1.10:5323'
```

If you do not include the username and password or if authentication fails, conda prompts for a username and password.

If your password contains special characters, you need to escape them as described in [Percent-encoding reserved characters](https://en.wikipedia.org/wiki/Percent-encoding#Percent-encoding_reserved_characters) on Wikipedia.

Be careful not to use ``http`` when you mean ``https`` or ``https`` when you mean ``http``.

`ssl_verify`: SSL verification#

If you are behind a proxy that does SSL inspection, such as a Cisco IronPort Web Security Appliance (WSA), you may need to use `ssl_verify` to override the SSL verification settings.

By default, this variable is `True`, which means that SSL verification is used and conda verifies certificates for SSL connections. Setting this variable to `False` disables the connection's normal security and is not recommended:

```
ssl_verify: False
```

Added in version 23.9.0: The `ssl_verify: truststore` setting is only available with conda 23.9.0 or later and using Python 3.10 or later.

If the certificate authority is already trusted by the operating system, for instance because it was installed by a system administrator, you can tell conda to use the operating system certificate store by setting `ssl_verify` to "truststore":

```
ssl_verify: truststore
```


You can also set `ssl_verify` to a string path to a certificate, which can be used to verify SSL connections:

```
ssl_verify: corp.crt
```

```
### `offline`: Offline mode only#
```

Filters out all channel URLs that do not use the `file://` protocol. The default is `False`.

Example:

```
offline: True
```

```
## Advanced configuration#
```

```
### `allow_softlinks`: Disallow soft-linking#
```

When `allow_softlinks` is `True`, conda uses hard links when possible and soft links (symlinks) when hard links are not possible, such as when installing on

a different file system than the one that the package cache is on.

When ``allow_softlinks`` is ``False``, conda still uses hard links when possible, but when it is not possible, conda copies files. Individual packages can override this option, specifying that certain files should never be soft linked.

The default is ``True``.

Example:

```
allow_softlinks: False
```

``channel_alias``: Set a channel alias#

Whenever you use the ``-c`` or ``--channel`` flag to give conda a channel name that is not a URL, conda prepends the ``channel_alias`` to the name that it was given. The default ``channel_alias`` is `<https://conda.anaconda.org>`.

If ``channel_alias`` is set to ``https://my.anaconda.repo:8080/conda/``, then a user who runs the command ``conda install -c conda-forge some-package`` will install the package `some-package` from ``https://my.anaconda.repo:8080/conda/conda-forge``.

For example, the command:

```
conda install --channel asmeurer <package>
```

is the same as:

```
conda install --channel https://conda.anaconda.org/asmeurer <package>
```

You can set ``channel_alias`` to your own repository.

Example: To set ``channel_alias`` to your repository at
<https://your.repo.com>:

```
channel_alias: https://your.repo/
```

On Windows, you must include a slash ("/") at the end of the URL:

Example: <https://your.repo/conda/>

When ``channel_alias`` set to your repository at `<https://your.repo.com>`:

```
conda install --channel jsmith <package>
```

is the same as:

```
conda install --channel https://your.repo.com/jsmith <package>
```

``create_default_packages``: Always add packages by default#

When creating new environments, add the specified packages by default. The default packages are installed in every environment you create. You can override this option at the command prompt with the ``--no-default-packages`` flag. The default is to not include any packages.

****Example:****

```
create_default_packages:
```

- pip
- ipython
- scipy=0.15.0

`track_features`: Track features#

Enable certain features to be tracked by default. The default is to not track any features. This is similar to adding MKL to the `create_default_packages` list.

Example:

track_features:

- mkl

`update_dependencies`: Disable updating of dependencies#

By default, `conda install` updates the given package to the latest version and installs any dependencies necessary for that package. However, if dependencies that satisfy the package's requirements are already installed, conda will not update those packages to the latest version.

In this case, if you would prefer that conda update all dependencies to the

latest version that is compatible with the environment, set

``update_dependencies` to `True`.`

The default is ``False``.

Example:

```
update_dependencies: True
```

Note

Conda still ensures that dependency specifications are satisfied. Thus, some dependencies may still be updated or, conversely, this may prevent packages given at the command line from being updated to their latest versions. You can always specify versions at the command line to force conda to install a given version, such as ``conda install numpy=1.9.3``.

To avoid updating only specific packages in an environment, a better option may be to pin them. For more information, see [\[Preventing packages from updating \(pinning\)\]\(../tasks/manage-pkgs.html#pinning-packages\)](#).

`disallow`: Disallow installation of specific packages#

Disallow the installation of certain packages. The default is to allow

installation of all packages.

****Example:****

disallow:

- anaconda

`add_anaconda_token`: Add Anaconda.org token to automatically see private packages#

When the channel alias is Anaconda.org or an Anaconda Server GUI, you can set the system configuration so that users automatically see private packages.

Anaconda.org was formerly known as binstar.org. This uses the Anaconda command-line client, which you can install with `conda install anaconda-client`, to automatically add the token to the channel URLs.

The default is `True`.

****Example:****

add_anaconda_token: False

Note

Even when set to ``True``, this setting is enabled only if the Anaconda command-line client is installed and you are logged in with the ``anaconda login`` command.

``envs_dirs``: Specify environment directories#

Specify directories in which environments are located. If this key is set, the root prefix ``envs_dir`` is not used unless explicitly included. This key also determines where the package caches are located.

For each envs here, ``envs/pkgs`` is used as the pkgs cache, except for the standard ``envs`` directory in the root directory, for which the normal ``root_dir/pkgs`` is used.

Example:

`envs_dirs:`

- `~/my-envs`
- `/opt/anaconda/envs`

The ``CONDA_ENVS_PATH`` environment variable overwrites the ``envs_dirs`` setting:

* For macOS and Linux: ``CONDA_ENVS_PATH=~/.my-envs:/opt/anaconda/envs``

* For Windows: ``set CONDA_ENVS_PATH=C:\Users\joe\envs;C:\Anaconda\envs``

``pkgs_dirs``: Specify package directories#

Specify directories in which packages are located. If this key is set, the root prefix ``pkgs_dirs`` is not used unless explicitly included.

If the ``pkgs_dirs`` key is not set, then ``envs/pkgs`` is used as the pkgs cache, except for the standard ``envs`` directory in the root directory, for which the normal ``root_dir/pkgs`` is used.

Example:

`pkgs_dirs:`

- `/opt/anaconda/pkgs`

The ``CONDA_PKGS_DIRS`` environment variable overwrites the ``pkgs_dirs`` setting:

* For macOS and Linux: ``CONDA_PKGS_DIRS=/opt/anaconda/pkgs``

* For Windows: ``set CONDA_PKGS_DIRS=C:\Anaconda\pkgs``

`use_only_tar_bz2`: Force conda to download only .tar.bz2 packages#

Conda 4.7 introduced a new `.conda` package file format. `.conda` is a more compact and faster alternative to `.tar.bz2` packages. It's thus the preferred file format to use where available.

Nevertheless, it's possible to force conda to only download `.tar.bz2` packages by setting the `use_only_tar_bz2` boolean to `True`.

The default is `False`.

Example:

```
use_only_tar_bz2: True
```

Note

This is forced to `True` if conda-build is installed and older than 3.18.3, because older versions of conda break when conda feeds it the new file format.

`console`: Configure display type#

Added in version 24.11.0: the `console` setting is only available after this

version.

The ``console`` setting allows you to modify the way output is rendered for conda commands. This setting is primarily used as a way to select new reporter backends made available by plugins.

For example, a plugin may create a new reporter backend called "colors". As a user, you would configure it in your ``.condarc`` file as shown below:

```
console: colors
```

or specify it on the command line with the `--console`` option

```
conda info --console=colors
```

```
## Conda-build configuration#
```

```
### `root-dir`: Specify conda-build output root directory#
```

Build output root directory. You can also set this with the ``CONDA_BLD_PATH`` environment variable. The default is ``<CONDA_PREFIX>/conda-bld/``. If you do

not have write permissions to ``<CONDA_PREFIX>/conda-bld/``, the default is
`~/conda-bld/`.

Example:

conda-build:

root-dir: ~/conda-builds

`output_folder`: Specify conda-build build folder (conda-build 3.16.3+)#

Folder to dump output package to. Packages are moved here if build or test succeeds. If unset, the output folder corresponds to the same directory as
`root-dir`: the root build directory. .. code-block:: yaml

> conda-build:

>

>

> output_folder: conda-bld

`pkg_version`: Specify conda-build package version#

Conda package version to create. Use `2` for `.conda` packages. If not set, conda-build defaults to `.tar.bz2`.

conda-build:

pkg_format: 2

`anaconda_upload`: Automatically upload conda-build packages to
Anaconda.org#

Automatically upload packages built with conda-build to
[Anaconda.org](http://anaconda.org). The default is `False`.

Example:

anaconda_upload: True

`anaconda_token`: Token to be used for Anaconda.org uploads (conda-build
3.0+)#

Tokens are a means of authenticating with Anaconda.org without logging in. You
can pass your token to conda-build with this `.condarc` setting, or with a CLI
argument. This is unset by default. Setting it implicitly enables
`anaconda_upload`.

conda-build:

anaconda_token: gobbledygook

`quiet`: Limit build output verbosity (conda-build 3.0+)#

Conda-build's output verbosity can be reduced with the `quiet` setting. For more verbosity, use the CLI flag `--debug`.

conda-build:

quiet: true

`filename_hashing`: Disable filename hashing (conda-build 3.0+)#

Conda-build 3 adds hashes to filenames to allow greater customization of dependency versions. If you find this disruptive, you can disable the hashing with the following config entry:

conda-build:

filename_hashing: false

Warning

Conda-build does not check when clobbering packages. If you utilize conda-build 3's build matrices with a build configuration that is not reflected in the build string, packages will be missing due to clobbering.

```
### `no_verify`: Disable recipe and package verification (conda-build 3.0+)#
```

By default, conda-build uses conda-verify to ensure that your recipe and package meet some minimum sanity checks. You can disable these:

```
conda-build:
```

```
    no_verify: true
```

```
### `set_build_id`: Disable per-build folder creation (conda-build 3.0+)#
```

By default, conda-build creates a new folder for each build, named for the package name plus a timestamp. This allows you to do multiple builds at once. If you have issues with long paths, you may need to disable this behavior. You should first try to change the build output root directory with the ``root-dir`` setting described above, but fall back to this as necessary:

conda-build:

set_build_id: false

`skip_existing`: Skip building packages that already exist (conda-build 3.0+)#

By default, conda-build builds all recipes that you specify. You can instead skip recipes that are already built. A recipe is skipped if and only if `_all_` of its outputs are available on your currently configured channels.

conda-build:

skip_existing: true

`include_recipe`: Omit recipe from package (conda-build 3.0+)#

By default, conda-build includes the recipe that was used to build the package. If this contains sensitive or proprietary information, you can omit the recipe.

conda-build:

include_recipe: false

Note

If you do not include the recipe, you cannot use conda-build to test the package after the build completes. This means that you cannot split your build and test steps across two distinct CLI commands (`conda build --notest recipe`` and `conda build -t recipe``). If you need to omit the recipe and split your steps, your only option is to remove the recipe files from the tarball artifacts after your test step. Conda-build does not provide tools for doing that.

``activate``: Disable activation of environments during build/test (conda-build 3.0+)#

By default, conda-build activates the build and test environments prior to executing the build or test scripts. This adds necessary PATH entries, and also runs any `activate.d` scripts you may have. If you disable activation, the PATH will still be modified, but the `activate.d` scripts will not run. This is not recommended, but some people prefer this.

conda-build:

activate: false

```
### `long_test_prefix`: Disable long prefix during test (conda-build 3.16.3+)#
```

By default, conda-build uses a long prefix for the test prefix. If you have recipes that fail in long prefixes but would still like to test them in short prefixes, you can disable the long test prefix. This is not recommended.

```
conda-build:
```

```
    long_test_prefix: false
```

The default is `true`.

```
### `pypirc`: PyPI upload settings (conda-build 3.0+)#
```

Unset by default. If you have wheel outputs in your recipe, conda-build will try to upload them to the PyPI repository specified by the `pypi_repository` setting using credentials from this file path.

```
conda-build:
```

```
    pypirc: ~/.pypirc
```

```
### `pypi_repository`: PyPI repository to upload to (conda-build 3.0+)#
```

Unset by default. If you have wheel outputs in your recipe, conda-build will try to upload them to this PyPI repository using credentials from the file specified by the `pypirc` setting.

conda-build:

```
pypi_repository: pypi
```

```
## Expansion of environment variables#
```

Conda expands environment variables in a subset of configuration settings.

These are:

- * `channel`

- * `channel_alias`

- * `channels`

- * `client_cert_key`

- * `client_cert`

* `custom_channels`

* `custom_multichannels`

* `default_channels`

* `envs_dirs`

* `envs_path`

* `migrated_custom_channels`

* `pkgs_dirs`

* `proxy_servers`

* `verify_ssl`

* `allowlist_channels`

* `denylist_channels`

This allows you to store the credentials of a private repository in an environment variable, like so:

channels:

- `https://{USERNAME}:{PASSWORD}@my.private.conda.channel`

Configuring number of threads#

You can use your `.condarc` file or environment variables to add configuration to control the number of threads. You may want to do this to tweak conda to better utilize your system. If you have a very fast SSD, you might increase the number of threads to shorten the time it takes for conda to create environments and install/remove packages.

`repodata_threads`#

- * Default number of threads: None

- * Threads used when downloading, parsing, and creating repodata structures from `repodata.json` files. Multiple downloads from different channels may occur simultaneously. This speeds up the time it takes to start solving.

`verify_threads`#

- * Default number of threads: 1

- * Threads used when verifying the integrity of packages and files to be installed in your environment. Defaults to 1, as using multiple threads here can run into problems with slower hard

drives.

```
### `execute_threads`#
```

- * Default number of threads: 1

- * Threads used to unlink, remove, link, or copy files into your environment. Defaults to 1, as using multiple threads here can run into problems with slower hard drives.

```
### `default_threads`#
```

- * Default number of threads: None

- * When set, this value is used for all of the above thread settings. With its default setting (None), it does not affect the other settings.

Setting any of the above can be done in ``.condarc`` or with conda config:

At your terminal:

```
conda config --set repodata_threads 2
```

In ``.condarc``:

verify_threads: 4

__On this page

- * General configuration

- * ``channels``: Channel locations

- * ``default_channels``: Default channels

- * ``channel_settings``: Extra settings for individual channels

- * ``allowlist_channels`` and ``denylist_channels``: Allow or deny specific channels

- * ``auto_update_conda``: Update conda automatically

- * ``always_yes``: Always yes

- * ``show_channel_urls``: Show channel URLs

- * ``change_ps1``: Change command prompt

- * ``add_pip_as_python_dependency``: Add pip as Python dependency

- * ``use_pip``: Use pip

- * ``proxy_servers``: Configure conda for use behind a proxy server

- * ``ssl_verify``: SSL verification

- * ``offline``: Offline mode only

- * Advanced configuration

- * ``allow_softlinks``: Disallow soft-linking

- * ``channel_alias``: Set a channel alias

- * ``create_default_packages``: Always add packages by default

- * ``track_features``: Track features

- * ``update_dependencies``: Disable updating of dependencies

- * ``disallow``: Disallow installation of specific packages
- * ``add_anaconda_token``: Add Anaconda.org token to automatically see private packages
- * ``envs_dirs``: Specify environment directories
- * ``pkgs_dirs``: Specify package directories
- * ``use_only_tar_bz2``: Force conda to download only .tar.bz2 packages
- * ``console``: Configure display type
- * Conda-build configuration
 - * ``root-dir``: Specify conda-build output root directory
 - * ``output_folder``: Specify conda-build build folder (conda-build 3.16.3+)
 - * ``pkg_version``: Specify conda-build package version
 - * ``anaconda_upload``: Automatically upload conda-build packages to Anaconda.org
 - * ``anaconda_token``: Token to be used for Anaconda.org uploads (conda-build 3.0+)
 - * ``quiet``: Limit build output verbosity (conda-build 3.0+)
 - * ``filename_hashing``: Disable filename hashing (conda-build 3.0+)
 - * ``no_verify``: Disable recipe and package verification (conda-build 3.0+)
 - * ``set_build_id``: Disable per-build folder creation (conda-build 3.0+)
 - * ``skip_existing``: Skip building packages that already exist (conda-build 3.0+)
 - * ``include_recipe``: Omit recipe from package (conda-build 3.0+)
 - * ``activate``: Disable activation of environments during build/test (conda-build 3.0+)
 - * ``long_test_prefix``: Disable long prefix during test (conda-build 3.16.3+)
 - * ``pypirc``: PyPI upload settings (conda-build 3.0+)
 - * ``pypi_repository``: PyPI repository to upload to (conda-build 3.0+)
- * Expansion of environment variables
- * Configuring number of threads
 - * ``reodata_threads``
 - * ``verify_threads``
 - * ``execute_threads``

* ``default_threads``

[[__Edit on GitHub](https://github.com/conda/conda/edit/main/docs/source/user-guide/configuration/settings.rst)](https://github.com/conda/conda/edit/main/docs/source/user-guide/configuration/settings.rst)

[[__Show Source](https://github.com/conda/conda/edit/main/docs/source/user-guide/configuration/settings.rst.txt)](../../_sources/user-guide/configuration/settings.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)](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.