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

  * [__ 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

  * [__ 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`
* [`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`](../exceptions/index.html)
* [`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) __
* [` archspec`](../plugins/virtual_packages/archspec/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)
  *

`api`#

Collection of conda's high-level APIs.

Classes#

`Solver`	**Beta** While in beta, expect both major and minor changes across minor releases.
`SubdirData`	**Beta** While in beta, expect both major and minor changes across minor releases.
`PackageCacheData`	**Beta** While in beta, expect both major and minor changes across minor releases.
`PrefixData`	**Beta** While in beta, expect both major and minor changes across minor releases.

Attributes#

`DepsModifier` |
---|---
`UpdateModifier` |

DepsModifier#

UpdateModifier#

_class _Solver(_prefix_ , _channels_ , _subdirs =()_, _specs_to_add =()_,

_specs_to_remove =()_)#

Beta While in beta, expect both major and minor changes across minor

releases.

A high-level API to conda's solving logic. Three public methods are provided

to access a solution in various forms.

> * `solve_final_state()`
>
> * `solve_for_diff()`
>
> * `solve_for_transaction()`
>
>

Beta

Parameters:

* **prefix** ([_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)")) -- The conda prefix / environment location for which the `Solver` is being instantiated.

* **channels** (Sequence[`Channel`]) -- A prioritized list of channels to use for the solution.

* **subdirs** (_Sequence_ _[_[_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") _]_) -- A prioritized list of subdirs to use for the solution.

* **specs_to_add** (set[`MatchSpec`]) -- The set of package specs to add to the prefix.

* **specs_to_remove** (set[`MatchSpec`]) -- The set of package specs to remove from the prefix.

solve_final_state(_update_modifier =NULL_, _deps_modifier =NULL_, _prune =NULL_, _ignore_pinned =NULL_, _force_remove =NULL_)#

Beta While in beta, expect both major and minor changes across minor releases.

Gives the final, solved state of the environment.

Parameters:

* **deps_modifier** ([_DepsModifier_](../base/constants/index.html#conda.base.constants.DepsModifier "conda.base.constants.DepsModifier")) -- An optional flag indicating special solver handling for dependencies. The default solver behavior is to be as conservative as possible with dependency updates (in the case the dependency already exists in the environment), while still ensuring all dependencies are satisfied. Options include * NO_DEPS * ONLY_DEPS * UPDATE_DEPS * UPDATE_DEPS_ONLY_DEPS * FREEZE_INSTALLED

* **prune** ([_bool_](https://docs.python.org/3/library/functions.html#bool "(in Python v3.13)")) -- If `True`, the solution will not contain packages that were previously brought into the environment as dependencies but are no longer required as dependencies and are not user-requested.

* **ignore_pinned** ([_bool_](https://docs.python.org/3/library/functions.html#bool "(in Python v3.13)")) -- If `True`, the solution will ignore pinned package configuration for the prefix.

* **force_remove** ([_bool_](https://docs.python.org/3/library/functions.html#bool "(in Python v3.13)")) -- Forces removal of a package without removing packages that depend on it.

Returns:

In sorted dependency order from roots to leaves, the package references for the solved state of the environment.

Return type:

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

solve_for_diff(_update_modifier =NULL_, _deps_modifier =NULL_, _prune =NULL_, _ignore_pinned =NULL_, _force_remove =NULL_, _force_reinstall =False_)#

Beta While in beta, expect both major and minor changes across minor releases.

Gives the package references to remove from an environment, followed by the package references to add to an environment.

Parameters:

* **deps_modifier**

([_DepsModifier_](../base/constants/index.html#conda.base.constants.DepsModifier "conda.base.constants.DepsModifier")) -- See `solve_final_state()`.

* **prune** ([_bool_](https://docs.python.org/3/library/functions.html#bool "(in Python v3.13)")) -- See `solve_final_state()`.

* **ignore_pinned** ([_bool_](https://docs.python.org/3/library/functions.html#bool "(in Python v3.13)")) -- See `solve_final_state()`.

* **force_remove** ([_bool_](https://docs.python.org/3/library/functions.html#bool "(in Python v3.13)")) -- See `solve_final_state()`.

* **force_reinstall** ([_bool_](https://docs.python.org/3/library/functions.html#bool "(in Python v3.13)")) -- For requested specs_to_add that are already satisfied in the environment, instructs the solver to remove the package and spec from the environment, and then add it back--possibly with the exact package instance modified, depending on the spec exactness.

Returns:

A two-tuple of PackageRef sequences. The first is the group of packages to
remove from the environment, in sorted dependency order from leaves to roots.
The second is the group of packages to add to the environment, in sorted
dependency order from roots to leaves.

Return type:

[tuple](https://docs.python.org/3/library/stdtypes.html#tuple "\(in Python v3.13\)")[PackageRef],
[tuple](https://docs.python.org/3/library/stdtypes.html#tuple "\(in Python v3.13\)")[PackageRef]

solve_for_transaction(_update_modifier =NULL_, _deps_modifier =NULL_, _prune =NULL_, _ignore_pinned =NULL_, _force_remove =NULL_, _force_reinstall =False_)#

Beta While in beta, expect both major and minor changes across minor releases.

Gives an UnlinkLinkTransaction instance that can be used to execute the solution on an environment.

Parameters:

* **deps_modifier** ([_DepsModifier_](../base/constants/index.html#conda.base.constants.DepsModifier "conda.base.constants.DepsModifier")) -- See `solve_final_state()`.

* **prune** ([_bool_](https://docs.python.org/3/library/functions.html#bool "\(in Python v3.13\)")) -- See `solve_final_state()`.

* **ignore_pinned** ([_bool_](https://docs.python.org/3/library/functions.html#bool "\(in Python v3.13\)")) -- See `solve_final_state()`.

* **force_remove** ([_bool_](https://docs.python.org/3/library/functions.html#bool "\(in Python v3.13\)")) -- See `solve_final_state()`.

* **force_reinstall** ([_bool_](https://docs.python.org/3/library/functions.html#bool "\(in Python v3.13\)")) -- See `solve_for_diff()`.

Return type:

[UnlinkLinkTransaction](../core/link/index.html#conda.core.link.UnlinkLinkTransaction "conda.core.link.UnlinkLinkTransaction")

_class _SubdirData(_channel_)#

Beta While in beta, expect both major and minor changes across minor releases.

High-level management and usage of repodata.json for subdirs.

Beta While in beta, expect both major and minor changes across minor releases.

Parameters:

channel ([_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") _or_[_Channel_](../gateways/repodata/index.html#conda.gateways.repodata.Channel "conda.gateways.repodata.Channel")) --

The target subdir for the instance. Must either be a url that includes a subdir or a `Channel` that includes a subdir. e.g.:

> * '<https://repo.anaconda.com/pkgs/main/linux-64>'
>
> * Channel('<https://repo.anaconda.com/pkgs/main/linux-64>')
>
> * Channel('conda-forge/osx-64')
>
>

query(_package_ref_or_match_spec_)#

Beta While in beta, expect both major and minor changes across minor

releases.

Run a query against this specific instance of repodata.

Parameters:

package_ref_or_match_spec (_PackageRef_ _or_[

MatchSpec](../models/match_spec/index.html#conda.models.match_spec.MatchSpec

"conda.models.match_spec.MatchSpec") _or_[

str](https://docs.python.org/3/library/stdtypes.html#str "\(in Python

v3.13\")) -- Either an exact `PackageRef` to match against, or a `MatchSpec`

query object. A [`str`](https://docs.python.org/3/library/stdtypes.html#str

"\(in Python v3.13\") will be turned into a `MatchSpec` automatically.

Returns:

tuple[PackageRecord]

_static _query_all(_package_ref_or_match_spec_ , _channels =None_, _subdirs

=None_)#

Beta While in beta, expect both major and minor changes across minor releases.

Run a query against all repodata instances in channel/subdir matrix.

Parameters:

* **package_ref_or_match_spec** (_PackageRef_ _or_[_MatchSpec_](../models/match_spec/index.html#conda.models.match_spec.MatchSpec "conda.models.match_spec.MatchSpec") _or_[_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)")) -- Either an exact `PackageRef` to match against, or a `MatchSpec` query object. A [`str`](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") will be turned into a `MatchSpec` automatically.

* **channels** (_Iterable_ _[_[_Channel_](../gateways/repodata/index.html#conda.gateways.repodata.Channel "conda.gateways.repodata.Channel") _or_[_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") _] or_ _None_) -- An iterable of urls for channels or `Channel` objects. If None, will fall back to context.channels.

* **subdirs** (_Iterable_ _[_[_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python

v3.13\)") _] or_ _None_) -- If None, will fall back to context.subdirs.

Returns:

tuple[PackageRecord]

iter_records()#

Beta While in beta, expect both major and minor changes across minor

releases.

Returns:

A generator over all records contained in the repodata.json

instance. Warning: this is a generator that is exhausted on first use.

Return type:

Iterable[[PackageRecord](../models/records/index.html#conda.models.records.PackageRecord "conda.models.records.PackageRecord")]

reload()#

Beta While in beta, expect both major and minor changes across minor releases.

Update the instance with new information. Backing information (i.e. repodata.json) is lazily downloaded/loaded on first use by the other methods of this class. You should only use this method if you are _sure_ you have outdated data.

Returns:

SubdirData

_class _PackageCacheData(_pkgs_dir_)#

Beta While in beta, expect both major and minor changes across minor releases.

High-level management and usage of package caches.

Beta While in beta, expect both major and minor changes across minor releases.

Parameters:

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

_property _is_writable#

Beta While in beta, expect both major and minor changes across minor releases.

Indicates if the package cache location is writable or read-only.

Returns:

bool

get(_package_ref_ , _default =NULL_)#

Beta While in beta, expect both major and minor changes across minor

releases.

Parameters:

* **package_ref** (_PackageRef_) -- A `PackageRef` instance representing the key for the `PackageCacheRecord` being sought.

* **default** \-- The default value to return if the record does not exist. If not specified and no record exists, [`KeyError`](https://docs.python.org/3/library/exceptions.html#KeyError "\(in Python v3.13\)") is raised.

Returns:

PackageCacheRecord

query(_package_ref_or_match_spec_)#

Beta While in beta, expect both major and minor changes across minor releases.

Run a query against this specific package cache instance.

Parameters:

package_ref_or_match_spec (_PackageRef_ _or_[_MatchSpec_](../models/match_spec/index.html#conda.models.match_spec.MatchSpec "conda.models.match_spec.MatchSpec") _or_[_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)")) -- Either an exact `PackageRef` to match against, or a `MatchSpec` query object. A [`str`](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") will be turned into a `MatchSpec` automatically.

Returns:

tuple[PackageCacheRecord]

_static _query_all(_package_ref_or_match_spec_ , _pkgs_dirs =None_)#

Beta While in beta, expect both major and minor changes across minor releases.

Run a query against all package caches.

Parameters:

* **package_ref_or_match_spec** (_PackageRef_ _or_[_MatchSpec_](../models/match_spec/index.html#conda.models.match_spec.MatchSpec "conda.models.match_spec.MatchSpec") _or_[_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)")) -- Either an exact `PackageRef` to match against, or a `MatchSpec` query object. A [`str`](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") will be turned into a `MatchSpec` automatically.

* **pkgs_dirs** (_Iterable_ _[_[_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") _] or_ _None_) -- If None, will fall back to context.pkgs_dirs.

Returns:

tuple[PackageCacheRecord]

iter_records()#

Beta While in beta, expect both major and minor changes across minor

releases.

Returns:

A generator over all records contained in the package

cache instance. Warning: this is a generator that is exhausted on first use.

Return type:

Iterable[[PackageCacheRecord](../models/records/index.html#conda.models.records.PackageCach

eRecord

"conda.models.records.PackageCacheRecord")]

_static _first_writable(_pkgs_dirs =None_)#

Beta While in beta, expect both major and minor changes across minor releases.

Get an instance object for the first writable package cache.

Parameters:

pkgs_dirs (_Iterable_ _[_[_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") _]_) -- If None, will fall back to context.pkgs_dirs.

Returns:

An instance for the first writable package cache.

Return type:

PackageCacheData

reload()#

Beta While in beta, expect both major and minor changes across minor releases.

Update the instance with new information. Backing information (i.e. contents of the pkgs_dir) is lazily loaded on first use by the other methods of this class. You should only use this method if you are _sure_ you have outdated data.

Returns:

PackageCacheData

_class _PrefixData(_prefix_path_)#

Beta While in beta, expect both major and minor changes across minor releases.

High-level management and usage of conda environment prefixes.

Beta While in beta, expect both major and minor changes across minor releases.

Parameters:

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

_property _is_writable#

Beta While in beta, expect both major and minor changes across minor releases.

Indicates if the prefix is writable or read-only.

Returns:

True if the prefix is writable. False if read-only. None if the prefix does

not exist as a conda environment.

Return type:

[bool](https://docs.python.org/3/library/functions.html#bool "\(in Python v3.13\)") or None

get(_package_ref_ , _default =NULL_)#

Beta While in beta, expect both major and minor changes across minor releases.

Parameters:

* **package_ref** (_PackageRef_) -- A `PackageRef` instance representing the key for the `PrefixRecord` being sought.

* **default** \-- The default value to return if the record does not exist. If not specified and no record exists, [`KeyError`](https://docs.python.org/3/library/exceptions.html#KeyError "\(in Python v3.13\)") is raised.

Returns:

PrefixRecord

query(_package_ref_or_match_spec_)#

Beta While in beta, expect both major and minor changes across minor releases.

Run a query against this specific prefix instance.

Parameters:

package_ref_or_match_spec (_PackageRef_ _or_[_MatchSpec_](../models/match_spec/index.html#conda.models.match_spec.MatchSpec "conda.models.match_spec.MatchSpec") _or_[_str_](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)")) -- Either an exact `PackageRef` to match against, or a `MatchSpec` query object. A [`str`](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") will be turned into a `MatchSpec` automatically.

Returns:

tuple[PrefixRecord]

iter_records()#

Beta While in beta, expect both major and minor changes across minor releases.

Returns:

A generator over all records contained in the prefix.

Warning: this is a generator that is exhausted on first use.

Return type:

Iterable[[PrefixRecord](../models/records/index.html#conda.models.records.PrefixRecord "conda.models.records.PrefixRecord")]

reload()#

Beta While in beta, expect both major and minor changes across minor releases.

Update the instance with new information. Backing information (i.e. contents of the conda-meta directory) is lazily loaded on first use by the other methods of this class. You should only use this method if you are _sure_ you have outdated data.

Returns:

PrefixData

__On this page

* Classes
* Attributes
  * `DepsModifier`
  * `UpdateModifier`

    * `Solver`
      * `Solver.solve_final_state()`
      * `Solver.solve_for_diff()`
      * `Solver.solve_for_transaction()`
    * `SubdirData`
      * `SubdirData.query()`
      * `SubdirData.query_all()`
      * `SubdirData.iter_records()`
      * `SubdirData.reload()`
    * `PackageCacheData`
      * `PackageCacheData.is_writable`
      * `PackageCacheData.get()`
      * `PackageCacheData.query()`
      * `PackageCacheData.query_all()`
      * `PackageCacheData.iter_records()`
      * `PackageCacheData.first_writable()`
      * `PackageCacheData.reload()`
    * `PrefixData`
      * `PrefixData.is_writable`
      * `PrefixData.get()`
      * `PrefixData.query()`
      * `PrefixData.iter_records()`
      * `PrefixData.reload()`

[__Edit on GitHub](https://github.com/conda/conda/edit/main/docs/source/dev-guide/api/conda/api/index.rst)

[__Show Source](../../../../_sources/dev-guide/api/conda/api/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 "Analytics Dashboard")

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