

[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``](../index.html) __
- * [`envs_manager``](../envs_manager/index.html)
- * [`index``](../index/index.html)
- * [`initialize``](../initialize/index.html)

- * [`link``](../link/index.html)
- * [`package_cache_data``](../package_cache_data/index.html)
- * [`path_actions``](../path_actions/index.html)
- * [`portability``](../portability/index.html)
- * [`prefix_data``](../prefix_data/index.html)
- * ``solve``
- * [`subdir_data``](../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)
* __
* [ `core`](../index.html)
*
```

`solve`#

The classic solver implementation.

Classes#

`Solver` | A high-level API to conda's solving logic. Three public methods are provided to access a

---|---

`SolverStateContainer` |

Functions#

`get_pinned_specs`(prefix) | Find pinned specs from file and return a tuple of MatchSpec.

---|---

`diff_for_unlink_link_precs`([force_reinstall]) |

_class _Solver(_prefix :

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

v3.13\))_ , _channels :

[collections.abc.Iterable](https://docs.python.org/3/library/collections.abc.html#collections.abc.Iterable

"\ (in Python

v3.13\)))[conda.models.channel.Channel](../models/channel/index.html#conda.models.channel.Channel

"conda.models.channel.Channel")]_ , _subdirs :

[collections.abc.Iterable](https://docs.python.org/3/library/collections.abc.html#collections.abc.Iterable

"\ (in Python

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

Python v3.13\)))] = ()_ , _specs_to_add :

[collections.abc.Iterable](https://docs.python.org/3/library/collections.abc.html#collections.abc.Iterable

"\ (in Python

v3.13\)))[conda.models.match_spec.MatchSpec](../models/match_spec/index.html#conda.models

"conda.models.match_spec.MatchSpec")] = ()_ , _specs_to_remove :

[collections.abc.Iterable](https://docs.python.org/3/library/collections.abc.html#collections.abc.Iterable

"\ (in Python

v3.13\)))[conda.models.match_spec.MatchSpec](../models/match_spec/index.html#conda.models

"conda.models.match_spec.MatchSpec")] = ()_ , _reodata_fn :

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

v3.13\)) = REPODATA_FN_ , _command =NULL_)#

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()`  
>  
>
```

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_for_transaction(_update_modifier=NULL_, _deps_modifier=NULL_, _prune
=NULL_, _ignore_pinned=NULL_, _force_remove=NULL_, _force_reinstall=NULL_,
_should_retry_solve=False_)#
```

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()`.

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

Return type:

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

solve_for_diff(_update_modifier =NULL_, _deps_modifier =NULL_, _prune =NULL_,
_ignore_pinned =NULL_, _force_remove =NULL_, _force_reinstall =NULL_,
_should_retry_solve =False_) ->

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

v3.13\`))[[conda.models.records.PackageRecord](../models/records/index.html#conda.models.records.PackageRecord

"conda.models.records.PackageRecord"), Ellipsis],

[tuple](https://docs.python.org/3/library/stdtypes.html#tuple "\in Python v3.13\`))[[conda.models.records.PackageRecord](../models/records/index.html#conda.models.records.PackageRecord

"conda.models.records.PackageRecord"), Ellipsis]]#

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.

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

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_final_state(_update_modifier =NULL_, _deps_modifier =NULL_, _prune
=FALSE_)#
```

Gives the final, solved state of the environment.

Parameters:

```
*
**update_modifier**
([_UpdateModifier_](../../base/constants/index.html#conda.base.constants.UpdateModifier
"conda.base.constants.UpdateModifier")) -- An optional flag directing how updates are handled
regarding packages already existing in the environment.
```

```
*
**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.

* **should_retry_solve** ([_bool_](https://docs.python.org/3/library/functions.html#bool "(in Python v3.13\))) -- Indicates whether this solve will be retried. This allows us to control whether to call `find_conflicts (slow)` in `ssc.r.solve`

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]

determine_constricting_specs(_spec_ , _solution_precs_)#

get_request_package_in_solution(_solution_precs_ , _specs_map_)#

get_constrained_packages(_pre_packages_ , _post_packages_ , _index_keys_)#

_collect_all_metadata(_ssc_)#

_remove_specs(_ssc_)#

_find_inconsistent_packages(_ssc_)#

_package_has_updates(_ssc_ , _spec_ , _installed_pool_)#

`_should_freeze(_ssc_ , _target_prec_ , _conflict_specs_ , _explicit_pool_ ,
_installed_pool_)#`

`_add_specs(_ssc_)#`

`_run_sat(_ssc_)#`

`_post_sat_handling(_ssc_)#`

`_notify_conda_outdated(_link_precs_)#`

`_prepare(_prepared_specs_)#`

`_class _SolverStateContainer(_prefix_ , _update_modifier_ , _deps_modifier_ ,`

`_prune_ , _ignore_pinned_ , _force_remove_ , _should_retry_solve_)#`

`prefix_data()#`

`specs_from_history_map()#`

`track_features_specs()#`

`pinned_specs()#`

`set_repository_metadata(_index_ , _r_)#`

`_init_solution_precs()#`

`working_state_reset()#`

`get_pinned_specs(_prefix_)#`

Find pinned specs from file and return a tuple of MatchSpec.

`diff_for_unlink_link_precs(_prefix_ , _final_precs_ , _specs_to_add =(),
_force_reinstall =NULL_) ->`

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

`v3.13\)))[conda.models.records.PackageRecord](../models/records/index.html#conda.models.rec
ords.PackageRecord`

`"conda.models.records.PackageRecord"), Ellipsis],`

`[tuple](https://docs.python.org/3/library/stdtypes.html#tuple "(in Python
v3.13\)))[conda.models.records.PackageRecord](../models/records/index.html#conda.models.rec
ords.PackageRecord`

`"conda.models.records.PackageRecord"), Ellipsis]]#`

__On this page

- * Classes

- * Functions

- * `Solver``

- * `Solver.solve_for_transaction()``

- * `Solver.solve_for_diff()``

- * `Solver.solve_final_state()``

- * `Solver.determine_constraining_specs()``

- * `Solver.get_request_package_in_solution()``

- * `Solver.get_constrained_packages()``

- * `Solver._collect_all_metadata()``

- * `Solver._remove_specs()``

- * `Solver._find_inconsistent_packages()``

- * `Solver._package_has_updates()``

- * `Solver._should_freeze()``

- * `Solver._add_specs()``

- * `Solver._run_sat()``

- * `Solver._post_sat_handling()``

- * `Solver._notify_conda_outdated()``

- * `Solver._prepare()``

- * `SolverStateContainer``

- * `SolverStateContainer.prefix_data()``

- * `SolverStateContainer.specs_from_history_map()``

- * `SolverStateContainer.track_features_specs()``

- * `SolverStateContainer.pinned_specs()``

- * `SolverStateContainer.set_repository_metadata()``

- * `SolverStateContainer._init_solution_precs()``

- * ``SolverStateContainer.working_state_reset()``
- * ``get_pinned_specs()``
- * ``diff_for_unlink_link_precs()``

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

[[__Show Source](https://github.com/conda/conda/edit/main/docs/source/dev-guide/api/conda/core/solve/index.rst.txt)](../../../../../_sources/dev-guide/api/conda/core/solve/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)](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.