

[Skip to main content](#)

[__Back to top](#)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Navigation

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

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

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

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

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

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

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

- * [link`](../../core/link/index.html)
- * [package_cache_data`](../../core/package_cache_data/index.html)
- * [path_actions`](../../core/path_actions/index.html)
- * [portability`](../../core/portability/index.html)
- * [prefix_data`](../../core/prefix_data/index.html)
- * [solve`](../../core/solve/index.html)
- * [subdir_data`](../../core/subdir_data/index.html)
- * [deprecations`](../../deprecations/index.html)
- * [env`](../../env/index.html) ____
- * [env`](../../env/env/index.html)
- * [installers`](../../env/installers/index.html) ____
- * [base`](../../env/installers/base/index.html)
- * [conda`](../../env/installers/conda/index.html)
- * [pip`](../../env/installers/pip/index.html)
- * [pip_util`](../../env/pip_util/index.html)
- * [specs`](../../env/specs/index.html) ____
- * [binstar`](../../env/specs/binstar/index.html)
- * [requirements`](../../env/specs/requirements/index.html)
- * [yaml_file`](../../env/specs/yaml_file/index.html)
- * [exception_handler`](../../exception_handler/index.html)
- * [exceptions`](../../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``](../index.html) __
- * [`hookspec``](../hookspec/index.html)
- * [`manager``](../manager/index.html)
- * [`post_solves``](../post_solves/index.html) __
 - * [`signature_verification``](../post_solves/signature_verification/index.html)
- * [`reporter_backends``](../reporter_backends/index.html) __
 - * [`console``](../reporter_backends/console/index.html)
 - * [`json``](../reporter_backends/json/index.html)
- * [`solvers``](../solvers/index.html)
- * [`subcommands``](../subcommands/index.html) __
 - * [`doctor``](../subcommands/doctor/index.html) __
 - * [`health_checks``](../subcommands/doctor/health_checks/index.html)

- * `types`
- * [`virtual_packages`](../virtual_packages/index.html) __
- * [`archspeg`](../virtual_packages/archspec/index.html)
- * [`conda`](../virtual_packages/conda/index.html)
- * [`cuda`](../virtual_packages/cuda/index.html)
- * [`freebsd`](../virtual_packages/freebsd/index.html)
- * [`linux`](../virtual_packages/linux/index.html)
- * [`osx`](../virtual_packages/osx/index.html)
- * [`windows`](../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)

* `__`

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

*

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

Definition of specific return types for use when defining a conda plugin hook.

Each type corresponds to the plugin hook for which it is used.

Classes#

[`CondaSubcommand``](../../plugins/subcommands.html#conda.plugins.types.CondaSubcommand "conda.plugins.types.CondaSubcommand") | Return type to use when defining a conda subcommand plugin hook.

---|---

[`CondaVirtualPackage``](../../plugins/virtual_packages.html#conda.plugins.types.CondaVirtualPackage "conda.plugins.types.CondaVirtualPackage") | Return type to use when defining a conda virtual package plugin hook.

[`CondaSolver``](../../plugins/solvers.html#conda.plugins.types.CondaSolver "conda.plugins.types.CondaSolver") | Return type to use when defining a conda solver plugin hook.

[`CondaPreCommand`](../../plugins/pre_commands.html#conda.plugins.types.CondaPreCommand "conda.plugins.types.CondaPreCommand") | Return type to use when defining a conda pre-command plugin hook.

[`CondaPostCommand`](../../plugins/post_commands.html#conda.plugins.types.CondaPostCommand "conda.plugins.types.CondaPostCommand") | Return type to use when defining a conda post-command plugin hook.

`ChannelNameMixin` | Class mixin to make all plugin implementations compatible, e.g. when they

`ChannelAuthBase` | Base class that we require all plugin implementations to use to be compatible.

[`CondaAuthHandler`](../../plugins/auth_handlers.html#conda.plugins.types.CondaAuthHandler "conda.plugins.types.CondaAuthHandler") | Return type to use when the defining the conda auth handlers hook.

[`CondaHealthCheck`](../../plugins/health_checks.html#conda.plugins.types.CondaHealthCheck "conda.plugins.types.CondaHealthCheck") | Return type to use when defining conda health checks plugin hook.

`CondaPreSolve` | Return type to use when defining a conda pre-solve plugin hook.

`CondaPostSolve` | Return type to use when defining a conda post-solve plugin hook.

[`CondaSetting`](../../plugins/settings.html#conda.plugins.types.CondaSetting "conda.plugins.types.CondaSetting") | Return type to use when defining a conda setting plugin hook.

`ProgressBarBase` | Helper class that provides a standard way to create an ABC using

`SpinnerBase` | Helper class that provides a standard way to create an ABC using

`ReporterRendererBase` | Base class for all reporter renderers.

`CondaReporterBackend` | Return type to use when defining a conda reporter backend plugin hook.

[`CondaRequestHeader`](../../plugins/request_headers.html#conda.plugins.types.CondaRequestHeader "conda.plugins.types.CondaRequestHeader") | Define vendor specific headers to include

HTTP requests

`_class _CondaSubcommand#`

Return type to use when defining a conda subcommand plugin hook.

For details on how this is used, see

```
[`conda_subcommands()`](../../plugins/subcommands.html#conda.plugins.hooks.spec.CondaSpecs.conda_subcommands  
"conda.plugins.hooks.spec.CondaSpecs.conda_subcommands").
```

Parameters:

`***name**` `--` Subcommand name (e.g., ``conda my-subcommand-name``).

`***summary**` `--` Subcommand summary, will be shown in ``conda --help``.

`***action**` `--` Callable that will be run when the subcommand is invoked.

`***configure_parser**` `--` Callable that will be run when the subcommand parser is initialized.

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

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

action _:
Callable[[argparse.Namespace](https://docs.python.org/3/library/argparse.html#argparse.Namespace "(in Python v3.13\)) | [tuple](https://docs.python.org/3/library/stdtypes.html#tuple "(in Python v3.13\))][str](https://docs.python.org/3/library/stdtypes.html#str "(in Python v3.13\))]],
[int](https://docs.python.org/3/library/functions.html#int "(in Python v3.13\)) |
[None](https://docs.python.org/3/library/constants.html#None "(in Python v3.13\))_#

configure_parser _:
Callable[[argparse.ArgumentParser](https://docs.python.org/3/library/argparse.html#argparse.ArgumentParser "(in Python v3.13\))], [None](https://docs.python.org/3/library/constants.html#None "(in Python v3.13\))] | [None](https://docs.python.org/3/library/constants.html#None "(in Python v3.13\))_#

_class _CondaVirtualPackage#

Bases: ``NamedTuple``

Return type to use when defining a conda virtual package plugin hook.

For details on how this is used, see

```
[`conda_virtual_packages()`](../../plugins/virtual_packages.html#conda.plugins.hooks.spec.Con-  
daSpecs.conda_virtual_packages  
"conda.plugins.hooks.spec.Con-  
daSpecs.conda_virtual_packages").
```

Parameters:

`* **name**` `--` Virtual package name (e.g., ``my_custom_os``).

`* **version**` `--` Virtual package version (e.g., ``1.2.3``).

`* **build**` `--` Virtual package build string (e.g., ``x86_64``).

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

`version _:` `[str](https://docs.python.org/3/library/stdtypes.html#str "(in Python v3.13\)) |
[None](https://docs.python.org/3/library/constants.html#None "(in Python v3.13\))_#`

```
build _: [str](https://docs.python.org/3/library/stdtypes.html#str "(in Python v3.13\)) |
[None](https://docs.python.org/3/library/constants.html#None "(in Python v3.13\))_#
```

to_virtual_package() ->

```
[conda.models.records.PackageRecord](../../models/records/index.html#conda.models.records.PackageRecord
```

```
"conda.models.records.PackageRecord")#
```

```
_class _CondaSolver#
```

Bases: `NamedTuple`

Return type to use when defining a conda solver plugin hook.

For details on how this is used, see

```
[`conda_solvers()`](../../plugins/solvers.html#conda.plugins.hookspec.CondaSpecs.conda_solvers
```

```
"conda.plugins.hookspec.CondaSpecs.conda_solvers").
```

Parameters:

`* **name** -- Solver name (e.g., `custom-solver`).`

`* **backend** -- Type that will be instantiated as the solver backend.`

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

`backend _: [type](https://docs.python.org/3/library/functions.html#type "(in
Python
v3.13\))"[[conda.core.solve.Solver](../core/solve/index.html#conda.core.solve.Solver
"conda.core.solve.Solver")])_#`

`_class _CondaPreCommand#`

Bases: ``NamedTuple``

Return type to use when defining a conda pre-command plugin hook.

For details on how this is used, see

```
[`conda_pre_commands()`](../../plugins/pre_commands.html#conda.plugins.hookspec.CondaSpecs.conda_pre_commands  
ecs.conda_pre_commands  
"conda.plugins.hookspec.CondaSpecs.conda_pre_commands").
```

Parameters:

name -- Pre-command name (e.g., `custom_plugin_pre_commands`).

action -- Callable which contains the code to be run.

run_for -- Represents the command(s) this will be run on (e.g. `install` or `create`).

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

action `_`: `Callable[[[str](https://docs.python.org/3/library/stdtypes.html#str "\ (in Python v3.13\))],
[None](https://docs.python.org/3/library/constants.html#None "\ (in Python v3.13\))]]_#`

run_for `_`: `[set](https://docs.python.org/3/library/stdtypes.html#set "\ (in`

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

"\ (in Python v3.13\))"]_#

_class _CondaPostCommand#

Bases: `NamedTuple`

Return type to use when defining a conda post-command plugin hook.

For details on how this is used, see

[`conda_post_commands()`](../.../plugins/post_commands.html#conda.plugins.hooks.spec.CondaSpecs.conda_post_commands

"conda.plugins.hooks.spec.CondaSpecs.conda_post_commands").

Parameters:

* **name** \-- Post-command name (e.g., `custom_plugin_post_commands`).

* **action** \-- Callable which contains the code to be run.

* **run_for** \-- Represents the command(s) this will be run on (e.g. `install` or `create`).

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

action _ : Callable[[[str](<https://docs.python.org/3/library/stdtypes.html#str> "(in Python v3.13\)")],
[None](<https://docs.python.org/3/library/constants.html#None> "(in Python v3.13\)")]_#

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

_class _ChannelNameMixin(_channel_name :
[str](<https://docs.python.org/3/library/stdtypes.html#str> "(in Python v3.13\)")_, _* args_, _** kwargs_)#

Class mixin to make all plugin implementations compatible, e.g. when they use an existing (e.g. 3rd party) requests authentication handler.

Please use the concrete `ChannelAuthBase` in case you're creating an own implementation.

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

Bases: `ChannelNameMixin`, `requests.auth.AuthBase`

Base class that we require all plugin implementations to use to be compatible.

Authentication is tightly coupled with individual channels. Therefore, an additional `channel_name` property must be set on the `requests.auth.AuthBase` based class.

```
_class _CondaAuthHandler#
```

Bases: `NamedTuple`

Return type to use when the defining the conda auth handlers hook.

Parameters:

* **name** -- Name (e.g., `basic-auth`). This name should be unique and only one may be registered at a time.

* **handler** -- Type that will be used as the authentication handler during network requests.

name _: [str](<https://docs.python.org/3/library/stdtypes.html#str>) (in Python v3.13)

handler _: [type](<https://docs.python.org/3/library/functions.html#type>) (in Python v3.13)

_class _CondaHealthCheck#

Bases: `NamedTuple`

Return type to use when defining conda health checks plugin hook.

name _: [str](<https://docs.python.org/3/library/stdtypes.html#str>) (in Python

v3.13\")_#

action _ : Callable[[[str](<https://docs.python.org/3/library/stdtypes.html#str>
"\(in Python v3.13\)"),
[bool](<https://docs.python.org/3/library/functions.html#bool> "\(in Python
v3.13\)"), [None](<https://docs.python.org/3/library/constants.html#None> "\(in
Python v3.13\)")]_#

_class _CondaPreSolve#

Return type to use when defining a conda pre-solve plugin hook.

For details on how this is used, see

[`conda_pre_solves()`]([../hookspec/index.html#conda.plugins.hooks.specs.CondaSpecs.conda_pre_sol](https://docs.pydata.org/en/stable/conda.plugins.hooks.html#conda.plugins.hooks.specs.CondaSpecs.conda_pre_solves)
ves
"conda.plugins.hooks.specs.CondaSpecs.conda_pre_solves").

Parameters:

`* **name** \-- Pre-solve name (e.g., `custom_plugin_pre_solve`).`

`* **action** \-- Callable which contains the code to be run.`

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

```
action _:
```

```
Callable[[frozenset](https://docs.python.org/3/library/stdtypes.html#frozenset
```

```
"(in Python
```

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

```
"conda.models.match_spec.MatchSpec")],
```

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

```
Python
```

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

```
"conda.models.match_spec.MatchSpec")]]],
```

```
[None](https://docs.python.org/3/library/constants.html#None "(in Python
```

```
v3.13\))")_#
```

```
_class _CondaPostSolve#
```

Return type to use when defining a conda post-solve plugin hook.

For details on how this is used, see

```
[`conda_post_solves()`](../hookspec/index.html#conda.plugins.hookspec.CondaSpecs.conda_post_solves  
solves  
"conda.plugins.hookspec.CondaSpecs.conda_post_solves").
```

Parameters:

name -- Post-solve name (e.g., `custom_plugin_post_solve`).

action -- Callable which contains the code to be run.

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

action `_`: `Callable[[[str](https://docs.python.org/3/library/stdtypes.html#str
"(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]],
[None](https://docs.python.org/3/library/constants.html#None "(in Python
v3.13\))"]_#
```

```
_class _CondaSetting#
```

Return type to use when defining a conda setting plugin hook.

For details on how this is used, see

```
[`conda_settings()`](../../plugins/settings.html#conda.plugins.hooks.spec.CondaSpecs.conda_setti
ngs
```

```
"conda.plugins.hooks.spec.CondaSpecs.conda_settings").
```

Parameters:

```
* **name** -- name of the setting (e.g., `config_param`)
```

* **description** \-- description of the setting that should be targeted towards users of the plugin

* **parameter** \-- Parameter instance containing the setting definition

* **aliases** \-- alternative names of the setting

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

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

parameter _:

[conda.common.configuration.Parameter](../common/configuration/index.html#conda.common.configuration.Parameter
"conda.common.configuration.Parameter")_#

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

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

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

Helper class that provides a standard way to create an ABC using inheritance.

```
_abstract _update_to(_fraction_) ->  
[None](https://docs.python.org/3/library/constants.html#None "(in Python  
v3.13\))">#
```

```
_abstract _refresh() ->  
[None](https://docs.python.org/3/library/constants.html#None "(in Python  
v3.13\))">#
```

```
_abstract _close() ->  
[None](https://docs.python.org/3/library/constants.html#None "(in Python
```

v3.13\)"#

finish()#

_classmethod _get_lock()#

_class _SpinnerBase(_message :

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

v3.13\)"_ , _fail_message :

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

v3.13\)" = 'failed\n'_)#

Bases: [`abc.ABC`](https://docs.python.org/3/library/abc.html#abc.ABC "(in

Python v3.13\)"

Helper class that provides a standard way to create an ABC using inheritance.

_abstract ____enter__()#

```
_abstract __exit__(_exc_type_ , _exc_val_ , _exc_tb_)#
```

```
_class _ReporterRendererBase#
```

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

Base class for all reporter renderers.

```
render(_data : Any_, ** kwargs_) ->
```

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

```
_abstract _detail_view(_data : [dict](https://docs.python.org/3/library/stdtypes.html#dict "\(in Python v3.13\)")[[str](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)")],  
[str](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") |  
[int](https://docs.python.org/3/library/functions.html#int "\(in Python v3.13\)") |  
[bool](https://docs.python.org/3/library/functions.html#bool "\(in Python v3.13\)")], ** kwargs_) ->  
[str](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)")#
```


Render the output in a "tabular" format.

```
_abstract _envs_list(_data_ , ** kwargs_) ->
```

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

Render a list of environments

```
_abstract _progress_bar(_description :
```

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

Return a `ProgressBarBase` object to use as a progress bar

```
_classmethod _progress_bar_context_manager() ->
```

```
[contextlib.AbstractContextManager](https://docs.python.org/3/library/contextlib.html#contextlib.Abst  
ractContextManager  
"(in Python v3.13\)"#
```

Returns a null context by default but allows plugins to define their own if necessary

```
_abstract_spinner(_message_ , _failed_message_) -> SpinnerBase#
```

Return a `SpinnerBase` object to use as a spinner (i.e. loading dialog)

```
_abstract_prompt(_message :
```

```
[str](https://docs.python.org/3/library/stdtypes.html#str "(in Python  
v3.13\))" = 'Proceed'_, _choices =('yes', 'no')_, _default :
```

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

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

Allows for defining an implementation of a "yes/no" confirmation function

```
_class _CondaReporterBackend#
```

Return type to use when defining a conda reporter backend plugin hook.

For details on how this is used, see:

```
[`conda_reporter_backends()`](../hookspec/index.html#conda.plugins.hookspec.CondaSpecs.conda_reporter_backends
```

```
"conda.plugins.hookspec.CondaSpecs.conda_reporter_backends").
```

Parameters:

* **name** -- name of the reporter backend (e.g., `email_reporter`) This is how the reporter backend will be referenced in configuration files.

* **description** -- short description of what the reporter handler does

* **renderer** -- implementation of `ReporterRendererBase` that will be used as the reporter renderer

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

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

```
renderer _: [type](https://docs.python.org/3/library/functions.html#type "(in
Python v3.13\")[ReporterRendererBase]_#
```

```
_class _CondaRequestHeader#
```

Define vendor specific headers to include HTTP requests

For details on how this is used, see

```
[`conda_request_headers()`](../../plugins/request_headers.html#conda.plugins.hookspec.Conda
Specs.conda_request_headers
```

"conda.plugins.hookspec.CondaSpecs.conda_request_headers") and

```
[`conda_session_headers()`](../../plugins/request_headers.html#conda.plugins.hookspec.Conda
Specs.conda_session_headers
```

"conda.plugins.hookspec.CondaSpecs.conda_session_headers").

Parameters:

name -- name of the header used in the HTTP request

value -- value of the header used in the HTTP request

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

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

__On this page

* Classes

* `CondaSubcommand`

* `CondaSubcommand.name`

* `CondaSubcommand.summary`

* `CondaSubcommand.action`

* `CondaSubcommand.configure_parser`

* `CondaVirtualPackage`

* `CondaVirtualPackage.name`

* `CondaVirtualPackage.version`

* `CondaVirtualPackage.build`

* `CondaVirtualPackage.to_virtual_package()`

* `CondaSolver`

* `CondaSolver.name`

* `CondaSolver.backend`

* `CondaPreCommand`

- * `CondaPreCommand.name`
- * `CondaPreCommand.action`
- * `CondaPreCommand.run_for`
- * `CondaPostCommand`
 - * `CondaPostCommand.name`
 - * `CondaPostCommand.action`
 - * `CondaPostCommand.run_for`
- * `ChannelNameMixin`
- * `ChannelAuthBase`
- * `CondaAuthHandler`
 - * `CondaAuthHandler.name`
 - * `CondaAuthHandler.handler`
- * `CondaHealthCheck`
 - * `CondaHealthCheck.name`
 - * `CondaHealthCheck.action`
- * `CondaPreSolve`
 - * `CondaPreSolve.name`
 - * `CondaPreSolve.action`
- * `CondaPostSolve`
 - * `CondaPostSolve.name`
 - * `CondaPostSolve.action`
- * `CondaSetting`
 - * `CondaSetting.name`
 - * `CondaSetting.description`
 - * `CondaSetting.parameter`
 - * `CondaSetting.aliases`
- * `ProgressBarBase`

- * ``ProgressBarBase.update_to()``
- * ``ProgressBarBase.refresh()``
- * ``ProgressBarBase.close()``
- * ``ProgressBarBase.finish()``
- * ``ProgressBarBase.get_lock()``
- * ``SpinnerBase``
 - * ``SpinnerBase.__enter__()``
 - * ``SpinnerBase.__exit__()``
- * ``ReporterRendererBase``
 - * ``ReporterRendererBase.render()``
 - * ``ReporterRendererBase.detail_view()``
 - * ``ReporterRendererBase.envs_list()``
 - * ``ReporterRendererBase.progress_bar()``
 - * ``ReporterRendererBase.progress_bar_context_manager()``
 - * ``ReporterRendererBase.spinner()``
 - * ``ReporterRendererBase.prompt()``
- * ``CondaReporterBackend``
 - * ``CondaReporterBackend.name``
 - * ``CondaReporterBackend.description``
 - * ``CondaReporterBackend.renderer``
- * ``CondaRequestHeader``
 - * ``CondaRequestHeader.name``
 - * ``CondaRequestHeader.value``

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

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

© Copyright 2017, Anaconda, Inc.

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

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

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