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

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

  * [`dist`](../dist/index.html)
  * [`enums`](../enums/index.html)
  * [`leased_path_entry`](../leased_path_entry/index.html)
  * [`match_spec`](../match_spec/index.html)
  * [`package_info`](../package_info/index.html)
  * [`prefix_graph`](../prefix_graph/index.html)
  * `records`
  * [`version`](../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)

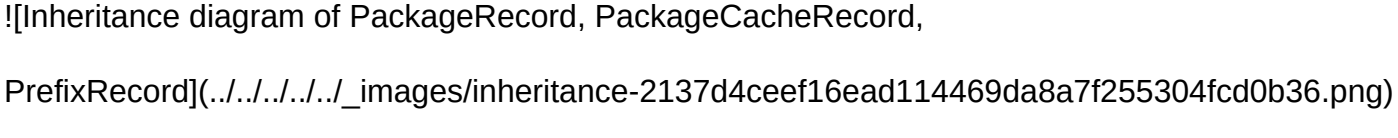
  * [__](../../../../../index.html)
  * [Developer guide](../../../../index.html)
  * __
  * [` models`](../index.html)
  *

`records`#

Implements the data model for conda packages.

A PackageRecord is the record of a package present in a channel. A
PackageCache is the record of a downloaded and cached package. A PrefixRecord
is the record of a package installed into a conda environment.

Object inheritance:



Classes#

`LinkTypeField`	Fields are doing something very similar to boxing and unboxing
`NoarchField`	Fields are doing something very similar to boxing and unboxing

`TimestampField` | Fields are doing something very similar to boxing and unboxing

`Link` |

`_FeaturesField` | Fields are doing something very similar to boxing and unboxing

`ChannelField` | Fields are doing something very similar to boxing and unboxing

`SubdirField` | Fields are doing something very similar to boxing and unboxing

`FilenameField` | Fields are doing something very similar to boxing and unboxing

`PackageTypeField` | Fields are doing something very similar to boxing and unboxing

`PathData` |

`PathDataV1` |

`PathsData` |

`PackageRecord` | Representation of a concrete package archive (tarball or .conda file).

`Md5Field` | Fields are doing something very similar to boxing and unboxing

`PackageCacheRecord` | Representation of a package that has been downloaded or unpacked in the local package cache.

`PrefixRecord` | Representation of a package that is installed in a local conda environmnet.

Attributes#

`EMPTY_LINK` |

---|---

_class _LinkTypeField(_enum_class_ , _default =NULL_, _required =True_,

validation =None, _in_dump =True_, _default_in_dump =True_, _nullable

=False_, _immutable =False_, _aliases =()_)#

Bases:

[`conda.auxlib.entity.EnumField`](../../auxlib/entity/index.html#conda.auxlib.entity.EnumField "conda.auxlib.entity.EnumField")

Fields are doing something very similar to boxing and unboxing of c#/java primitives. __set__ should take a "primitive" or "raw" value and create a "boxed" or "programmatically usable" value of it. While __get__ should return the boxed value, dump in turn should unbox the value into a primitive or raw value.

Parameters:

* **types** (_primitive literal_ _or_[_type_](https://docs.python.org/3/library/functions.html#type "\(in Python v3.13\)") _or_ _sequence_ _of_ _types_)

* **default** (_any_ _,__callable_ _,__optional_) -- If default is callable, it's guaranteed to return a valid value at the time of Entity creation.

* **required** (_boolean_ _,__optional_)

* **validation** (_callable_ _,__optional_)

* **dump** (_boolean_ _,__optional_)

box(_instance_ , _instance_type_ , _val_)#

_class _NoarchField(_enum_class_ , _default =NULL_, _required =True_,

validation =None, _in_dump =True_, _default_in_dump =True_, _nullable

=False_, _immutable =False_, _aliases =()_)#

Bases:

[`conda.auxlib.entity.EnumField`](../../auxlib/entity/index.html#conda.auxlib.entity.EnumField

"conda.auxlib.entity.EnumField")

Fields are doing something very similar to boxing and unboxing of c#/java

primitives. __set__ should take a "primitive" or "raw" value and create a

"boxed" or "programmatically usable" value of it. While __get__ should return

the boxed value, dump in turn should unbox the value into a primitive or raw

value.

Parameters:

* **types** (_primitive literal_ _or_[_type_](https://docs.python.org/3/library/functions.html#type

"\(in Python v3.13\)") _or_ _sequence_ _of_ _types_)

* **default** (_any_ _,__callable_ _,__optional_) -- If default is callable, it's guaranteed to return a

valid value at the time of Entity creation.

  * **required** (_boolean_ _,__optional_)

  * **validation** (_callable_ _,__optional_)

  * **dump** (_boolean_ _,__optional_)

box(_instance_ , _instance_type_ , _val_)#

_class _TimestampField#

Bases:
[`conda.auxlib.entity.NumberField`](../../auxlib/entity/index.html#conda.auxlib.entity.NumberField "conda.auxlib.entity.NumberField")

Fields are doing something very similar to boxing and unboxing of c#/java primitives. __set__ should take a "primitive" or "raw" value and create a "boxed" or "programmatically usable" value of it. While __get__ should return the boxed value, dump in turn should unbox the value into a primitive or raw value.

Parameters:

* **types** (_primitive literal_ _or_[_type_](https://docs.python.org/3/library/functions.html#type "\(in Python v3.13\)") _or_ _sequence_ _of_ _types_)

* **default** (_any_ _,__callable_ _,__optional_) -- If default is callable, it's guaranteed to return a valid value at the time of Entity creation.

* **required** (_boolean_ _,__optional_)

* **validation** (_callable_ _,__optional_)

* **dump** (_boolean_ _,__optional_)

_static __make_seconds(_val_)#

_static __make_milliseconds(_val_)#

box(_instance_ , _instance_type_ , _val_)#

dump(_instance_ , _instance_type_ , _val_)#

__get__(_instance_ , _instance_type_)#

_class _Link(_** kwargs_)#

Bases: `conda.auxlib.entity.DictSafeMixin`,

[`conda.auxlib.entity.Entity`](../../auxlib/entity/index.html#conda.auxlib.entity.Entity

"conda.auxlib.entity.Entity")

source#

type#

EMPTY_LINK#

_class __FeaturesField(_** kwargs_)#

Bases: [`conda.auxlib.entity.ListField`](../../auxlib/entity/index.html#conda.auxlib.entity.ListField "conda.auxlib.entity.ListField")

Fields are doing something very similar to boxing and unboxing of c#/java primitives. __set__ should take a "primitive" or "raw" value and create a "boxed" or "programmatically usable" value of it. While __get__ should return the boxed value, dump in turn should unbox the value into a primitive or raw value.

Parameters:

* **types** (_primitive literal_ _or_ _type_](https://docs.python.org/3/library/functions.html#type "\(in Python v3.13\)") _or_ _sequence_ _of_ _types_)

* **default** (_any_ _,__callable_ _,__optional_) -- If default is callable, it's guaranteed to return a valid value at the time of Entity creation.

* **required** (_boolean_ _,__optional_)

* **validation** (_callable_ _,__optional_)

* **dump** (_boolean_ _,__optional_)

box(_instance_ , _instance_type_ , _val_)#

dump(_instance_ , _instance_type_ , _val_)#

_class _ChannelField(_aliases =()_)#

Bases:
[`conda.auxlib.entity.ComposableField`](../../auxlib/entity/index.html#conda.auxlib.entity.ComposableField
"conda.auxlib.entity.ComposableField")

Fields are doing something very similar to boxing and unboxing of c#/java primitives. __set__ should take a "primitive" or "raw" value and create a "boxed" or "programmatically usable" value of it. While __get__ should return the boxed value, dump in turn should unbox the value into a primitive or raw value.

Parameters:

* **types** (_primitive literal_ _or_[_type_](https://docs.python.org/3/library/functions.html#type "\(in Python v3.13\)") _or_ _sequence_ _of_ _types_)

* **default** (_any_ _,__callable_ _,__optional_) -- If default is callable, it's guaranteed to return a valid value at the time of Entity creation.

* **required** (_boolean_ _,__optional_)

* **validation** (_callable_ _,__optional_)

* **dump** (_boolean_ _,__optional_)

dump(_instance_ , _instance_type_ , _val_)#

__get__(_instance_ , _instance_type_)#

_class _SubdirField#

Bases:

[`conda.auxlib.entity.StringField`](../../auxlib/entity/index.html#conda.auxlib.entity.StringField "conda.auxlib.entity.StringField")

Fields are doing something very similar to boxing and unboxing of c#/java primitives. __set__ should take a "primitive" or "raw" value and create a "boxed" or "programmatically usable" value of it. While __get__ should return the boxed value, dump in turn should unbox the value into a primitive or raw value.

Parameters:

* **types** (_primitive literal_ _or_[_type_](https://docs.python.org/3/library/functions.html#type "\(in Python v3.13\)") _or_ _sequence_ _of_ _types_)

* **default** (_any_ _,__callable_ _,__optional_) -- If default is callable, it's guaranteed to return a valid value at the time of Entity creation.

* **required** (_boolean_ _,__optional_)

* **validation** (_callable_ _,__optional_)

* **dump** (_boolean_ _,__optional_)

__get__(_instance_ , _instance_type_)#

_class _FilenameField(_aliases =()_)#

Bases:
[`conda.auxlib.entity.StringField`](../../auxlib/entity/index.html#conda.auxlib.entity.StringField "conda.auxlib.entity.StringField")

Fields are doing something very similar to boxing and unboxing of c#/java primitives. __set__ should take a "primitive" or "raw" value and create a "boxed" or "programmatically usable" value of it. While __get__ should return the boxed value, dump in turn should unbox the value into a primitive or raw value.

Parameters:

* **types** (_primitive literal_ _or_[_type_](https://docs.python.org/3/library/functions.html#type "\(in Python v3.13\)") _or_ _sequence_ _of_ _types_)

* **default** (_any_ _,__callable_ _,__optional_) -- If default is callable, it's guaranteed to return a valid value at the time of Entity creation.

* **required** (_boolean_ _,__optional_)

* **validation** (_callable_ _,__optional_)

* **dump** (_boolean_ _,__optional_)

__get__(_instance_ , _instance_type_)#

_class _PackageTypeField#

Bases:
[`conda.auxlib.entity.EnumField`](../../auxlib/entity/index.html#conda.auxlib.entity.EnumField "conda.auxlib.entity.EnumField")

Fields are doing something very similar to boxing and unboxing of c#/java primitives. __set__ should take a "primitive" or "raw" value and create a "boxed" or "programmatically usable" value of it. While __get__ should return the boxed value, dump in turn should unbox the value into a primitive or raw value.

Parameters:

* **types** (_primitive literal_ _or_[_type_](https://docs.python.org/3/library/functions.html#type "\(in Python v3.13\)") _or_ _sequence_ _of_ _types_)

* **default** (_any_ _,__callable_ _,__optional_) -- If default is callable, it's guaranteed to return a valid value at the time of Entity creation.

* **required** (_boolean_ _,__optional_)

* **validation** (_callable_ _,__optional_)

* **dump** (_boolean_ _,__optional_)

__get__(_instance_ , _instance_type_)#

_class _PathData(_** kwargs_)#

Bases: [`conda.auxlib.entity.Entity`](../../auxlib/entity/index.html#conda.auxlib.entity.Entity "conda.auxlib.entity.Entity")

_property _path#

_path#

prefix_placeholder#

file_mode#

no_link#

path_type#

_class _PathDataV1(_** kwargs_)#

Bases: `PathData`

sha256#

size_in_bytes#

inode_paths#

sha256_in_prefix#

_class _PathsData(_** kwargs_)#

Bases:

[`conda.auxlib.entity.Entity`](../../auxlib/entity/index.html#conda.auxlib.entity.Entity

"conda.auxlib.entity.Entity")

paths_version#

paths#

_class _PackageRecord(_* args_, _** kwargs_)#

Bases: `conda.auxlib.entity.DictSafeMixin`,
[`conda.auxlib.entity.Entity`](../../auxlib/entity/index.html#conda.auxlib.entity.Entity
"conda.auxlib.entity.Entity")

Representation of a concrete package archive (tarball or .conda file).

It captures all the relevant information about a given package archive,
including its source, in the following attributes.

Note that there are two subclasses, `PrefixRecord` and `PackageCacheRecord`.
These capture the same information, but are augmented with additional
information relevant for these two sources of packages.

Further note that `PackageRecord` makes use of its `_pkey` for comparison and
hash generation. This means that for common operations, like comparisons
between `PackageRecord` s and reference of `PackageRecord` s in mappings,

different objects appear identical. The fields taken into account are marked

in the following list of attributes. The subclasses do not add further

attributes to the `_pkey`.

_property _schannel#

The canonical name of the channel of this package.

Part of the `_pkey`.

Type:

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

v3.13\)")

_property __pkey#

The components of the PackageRecord that are used for comparison and hashing.

The `_pkey` is a tuple made up of the following fields of the `PackageRecord`.

Two `PackageRecord` s test equal if their respective `_pkey` s are equal. The

hash of the `PackageRecord` (important for dictionary access) is the hash of the `_pkey`.

The included fields are:

* `schannel`

* `subdir`

* `name`

* `version`

* `build_number`

* `build`

* `fn` only if [separate_format_cache](../../../../../configuration.html#auto-config-reference) is set to true (default: false)

Type:

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

_property _is_unmanageable#

_property _combined_depends#

_property _namekey#

name#

version#

build#

build_number#

channel#

subdir#

fn#

md5#

legacy_bz2_md5#

legacy_bz2_size#

url#

sha256#

arch#

platform#

depends#

constrains#

track_features#

features#

noarch#

preferred_env#

python_site_packages_path#

license#

license_family#

package_type#

timestamp#

date#

size#

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

__hash__()#

Return hash(self).

__eq__(_other_)#

Return self==value.

dist_str()#

dist_fields_dump()#

__str__()#

Return str(self).

to_match_spec()#

to_simple_match_spec()#

record_id()#

_classmethod _feature(_feature_name_) -> PackageRecord#

_classmethod _virtual_package(_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 "\(in Python v3.13\)") = None_, _build_string : [str](https://docs.python.org/3/library/stdtypes.html#str "\(in Python v3.13\)") | [None](https://docs.python.org/3/library/constants.html#None "\(in Python v3.13\)") = None_) -> PackageRecord#

Create a virtual package record.

Parameters:

* **name** \-- The name of the virtual package.

* **version** \-- The version of the virtual package, defaults to "0".

* **build_string** \-- The build string of the virtual package, defaults to "0".

Returns:

A PackageRecord representing the virtual package.

_class _Md5Field#

Bases:

[`conda.auxlib.entity.StringField`](../../auxlib/entity/index.html#conda.auxlib.entity.StringField "conda.auxlib.entity.StringField")

Fields are doing something very similar to boxing and unboxing of c#/java primitives. __set__ should take a "primitive" or "raw" value and create a "boxed" or "programmatically usable" value of it. While __get__ should return the boxed value, dump in turn should unbox the value into a primitive or raw value.

Parameters:

* **types** (_primitive literal_ _or_[_type_](https://docs.python.org/3/library/functions.html#type "\(in Python v3.13\)") _or_ _sequence_ _of_ _types_)

* **default** (_any_ _,__callable_ _,__optional_) -- If default is callable, it's guaranteed to return a

valid value at the time of Entity creation.

* **required** (_boolean_ _,__optional_)

* **validation** (_callable_ _,__optional_)

* **dump** (_boolean_ _,__optional_)

__get__(_instance_ , _instance_type_)#

_class _PackageCacheRecord(_* args_, _** kwargs_)#

Bases: `PackageRecord`

Representation of a package that has been downloaded or unpacked in the local package cache.

Specialization of `PackageRecord` that adds information for packages that exist in the local package cache, either as the downloaded package file, or unpacked in its own package dir, or both.

Note that this class does not add new fields to the `PackageRecord._pkey` so that a pure `PackageRecord` object that has the same `_pkey` fields as a

different `PackageCacheRecord` object (or, indeed, a `PrefixRecord` object)

will be considered equal and will produce the same hash.

_property _is_fetched#

Whether the package file exists locally.

Type:

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

v3.13\)")

_property _is_extracted#

Whether the package has been extracted locally.

Type:

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

v3.13\)")

_property _tarball_basename#

The basename of the local package file.

Type:

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

v3.13\)")

package_tarball_full_path#

extracted_package_dir#

md5#

_calculate_md5sum()#

_class _PrefixRecord(_* args_, _** kwargs_)#

Bases: `PackageRecord`

Representation of a package that is installed in a local conda environmnet.

Specialization of `PackageRecord` that adds information for packages that are installed in a local conda environment or prefix.

Note that this class does not add new fields to the `PackageRecord._pkey` so that a pure `PackageRecord` object that has the same `_pkey` fields as a different `PrefixRecord` object (or, indeed, a `PackageCacheRecord` object) will be considered equal and will produce the same hash.

Objects of this class are generally constructed from metadata in json files inside $prefix/conda-meta.

package_tarball_full_path#

extracted_package_dir#

files#

paths_data#

link#

requested_spec#

auth#

__On this page

* Classes

* Attributes
  * `LinkTypeField`
    * `LinkTypeField.box()`
  * `NoarchField`
    * `NoarchField.box()`
  * `TimestampField`
    * `TimestampField._make_seconds()`
    * `TimestampField._make_milliseconds()`
    * `TimestampField.box()`
    * `TimestampField.dump()`
    * `TimestampField.__get__()`
  * `Link`
    * `Link.source`
    * `Link.type`
  * `EMPTY_LINK`
  * `_FeaturesField`
    * `_FeaturesField.box()`
    * `_FeaturesField.dump()`
  * `ChannelField`
    * `ChannelField.dump()`
    * `ChannelField.__get__()`
  * `SubdirField`
    * `SubdirField.__get__()`
  * `FilenameField`
    * `FilenameField.__get__()`
  * `PackageTypeField`
    * `PackageTypeField.__get__()`

* `PathData`
  * `PathData.path`
  * `PathData._path`
  * `PathData.prefix_placeholder`
  * `PathData.file_mode`
  * `PathData.no_link`
  * `PathData.path_type`
* `PathDataV1`
  * `PathDataV1.sha256`
  * `PathDataV1.size_in_bytes`
  * `PathDataV1.inode_paths`
  * `PathDataV1.sha256_in_prefix`
* `PathsData`
  * `PathsData.paths_version`
  * `PathsData.paths`
* `PackageRecord`
  * `PackageRecord.schannel`
  * `PackageRecord._pkey`
  * `PackageRecord.is_unmanageable`
  * `PackageRecord.combined_depends`
  * `PackageRecord.namekey`
  * `PackageRecord.name`
  * `PackageRecord.version`
  * `PackageRecord.build`
  * `PackageRecord.build_number`
  * `PackageRecord.channel`
  * `PackageRecord.subdir`

* `PackageRecord.fn`
* `PackageRecord.md5`
* `PackageRecord.legacy_bz2_md5`
* `PackageRecord.legacy_bz2_size`
* `PackageRecord.url`
* `PackageRecord.sha256`
* `PackageRecord.arch`
* `PackageRecord.platform`
* `PackageRecord.depends`
* `PackageRecord.constrains`
* `PackageRecord.track_features`
* `PackageRecord.features`
* `PackageRecord.noarch`
* `PackageRecord.preferred_env`
* `PackageRecord.python_site_packages_path`
* `PackageRecord.license`
* `PackageRecord.license_family`
* `PackageRecord.package_type`
* `PackageRecord.timestamp`
* `PackageRecord.date`
* `PackageRecord.size`
* `PackageRecord.metadata`
* `PackageRecord.__hash__()`
* `PackageRecord.__eq__()`
* `PackageRecord.dist_str()`
* `PackageRecord.dist_fields_dump()`
* `PackageRecord.__str__()`

    * `PackageRecord.to_match_spec()`
    * `PackageRecord.to_simple_match_spec()`
    * `PackageRecord.record_id()`
    * `PackageRecord.feature()`
    * `PackageRecord.virtual_package()`
  * `Md5Field`
    * `Md5Field.__get__()`
  * `PackageCacheRecord`
    * `PackageCacheRecord.is_fetched`
    * `PackageCacheRecord.is_extracted`
    * `PackageCacheRecord.tarball_basename`
    * `PackageCacheRecord.package_tarball_full_path`
    * `PackageCacheRecord.extracted_package_dir`
    * `PackageCacheRecord.md5`
    * `PackageCacheRecord._calculate_md5sum()`
  * `PrefixRecord`
    * `PrefixRecord.package_tarball_full_path`
    * `PrefixRecord.extracted_package_dir`
    * `PrefixRecord.files`
    * `PrefixRecord.paths_data`
    * `PrefixRecord.link`
    * `PrefixRecord.requested_spec`
    * `PrefixRecord.auth`

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

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

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.