

---

**TODO zz a**

*Release TODO zz a*

**unknown**

**May 31, 2021**



# CONTENTS

<b>1</b>	<b>License</b>	<b>3</b>
<b>2</b>	<b>Up-To-Date?</b>	<b>5</b>
<b>3</b>	<b>Dependencies</b>	<b>7</b>
<b>4</b>	<b>User Manual</b>	<b>9</b>
4.1	Installation . . . . .	9
4.2	First Steps . . . . .	9
4.3	Project Configuration . . . . .	9
4.4	Graphical User Interface . . . . .	9
4.5	The Annize Programming Interface . . . . .	10
<b>5</b>	<b>Command Line Interface Reference</b>	<b>11</b>
5.1	Positional Arguments . . . . .	11
5.2	Named Arguments . . . . .	11
5.3	Sub-commands: . . . . .	11
<b>6</b>	<b>API Reference</b>	<b>13</b>
6.1	annize namespace . . . . .	13
	<b>Python Module Index</b>	<b>99</b>
	<b>Index</b>	<b>101</b>



TODO Anise is a Python-based execution engine for automation tasks. Automation tasks exist in software development, and probably all kinds of other sectors. They typically require the execution of different smaller and larger tools. Complex tasks often need a sequence of many steps to execute, with some steps having dependencies to each other. Manually triggering all these steps in the graphical interfaces of all the involved tools is possible in theory, but will generate errors and frustration after some cycles. The automation interfaces of those tools are sometimes easier, but sometimes they are error-prone. Some tasks may also need to ask the user for some information in an interactive way. Some smaller parts might also be machine-specific (e.g. filesystem paths or the code how to access a password vault), while the entire task must be runnable on some different machines. In some situations, this can lead to a rather intransparent forest of different tools, with unique oddnesses and special conventions. As the number of different project increases, you will see more and more different tools, often doing a similar job, but for different platforms or frameworks and, of course, with different usage conventions. Spontaneously written glue scripts help in the beginning, but will explode as the complexity exceeds some threshold. Typical tasks in software development could be:

- Generating documentation- Testing- Automatic code generation- Creating packages- Creating a homepage, automatically built from the available version information, the packages, the documentation and so on- Deploying this homepage to a web server- Handling version information - e.g. print it in the manual- and many more
- The Anise framework allows you to implement all those tasks in a structured but generic way in a combination of XML and Python code. Once you have created this stuff at a defined place in your project, Anise lets you easily execute your tasks from command line (or from any editor if you embed it somehow). This gives you a common and easy interface to all your 'tool glue' code. The Anise engine executes arbitrary Python source code and provides some additional services like logging, parameter passing from command line, basic graphical user interface support, a plugin interface, a flexible event system, injecting code and data from other place, dependencies between code fragments, and more. On top of this engine, Anise comes with a bunch of implementations that fulfill tasks (or parts of them) of software development. There is a testing module, a documentation- and homepage-generator, some package building methods and a lot more. The implementations use the event system in many places in order to allow customization in a somewhat technical but very flexible way. Even so, those implementations are rather specific and it depends on the particular case, if, and how many of those implementations are useful.



## **LICENSE**

Annize is distributed under the terms of the AGPL 3 license. This also affects all included files without a license header (non-source files like images), unless they are explicitly mentioned as third-party content. Read the ‘Dependencies’ section for included third-party stuff.





## **UP-TO-DATE?**

Are you currently reading from another source than the homepage? If you are in doubt whether your package is up-to-date, you should visit the project homepage and check that. You are currently reading the documentation for version 5.0.6379.



## DEPENDENCIES

Annize makes use of some third-party parts.



Required: **Python 3.7**

Required: **GTK 3.24**



Recommended: **GNU/Linux**



Included: **banner image** (License: <http://creativecommons.org/licenses/by-sa/3.0>; from [here](#))



Included: **font ‘Inconsolata’** (for websites; by Raph Levien; License: OFL; from [here](#))



Included: **font ‘Khula’** (for websites; by Erin McLaughlin; License: OFL; from [here](#))



Included: **font ‘Symbola’** (for logo symbol; License: free for use; from [here](#))



Included: **icon set ‘Oxygen’** (some icons; from [here](#))



Included: **third-party project logos** (from [here](#))



## 4.1 Installation

TODO zz

## 4.2 First Steps

TODO zz TODO zz cli TODO zz file format TODO zz gui

## 4.3 Project Configuration

TODO zz TODO zz classes TODO zz features/xml namespaces TODO zz names, references and append\_to

### 4.3.1 Model

TODO zz

### 4.3.2 File Format

TODO zz

## 4.4 Graphical User Interface

TODO zz

## 4.5 The Annize Programming Interface

TODO zz TODO zz i18n TODO zz fs

### 4.5.1 Features

TODO zz

## COMMAND LINE INTERFACE REFERENCE

```
usage: annize [-h] [--project PROJECT] {do} ...
```

### 5.1 Positional Arguments

<b>command</b>	Possible choices: do The command to execute.
----------------	---

### 5.2 Named Arguments

<b>--project</b>	Project directory or Annize configuration file (otherwise: the current working directory). Annize will automatically try to find it in parent directories as well.
------------------	--

### 5.3 Sub-commands:

#### 5.3.1 do

Execute a task.

```
annize do [-h] [--with-answers-from-json-file WITH_ANSWERS_FROM_JSON_FILE]
          [--with-answers-from-json-string WITH_ANSWERS_FROM_JSON_STRING]
          [--with-answer WITH_ANSWER WITH_ANSWER]
          [taskname]
```

#### Positional Arguments

<b>taskname</b>	Task name. Default: “taskchooser”
-----------------	--------------------------------------

## Named Arguments

- with-answers-from-json-file** `an_Cli_withAnswersFromJsonFile`  
Default: []
- with-answers-from-json-string** `an_Cli_withAnswersFromJsonString`  
Default: []
- with-answer** `an_Cli_withAnswer`  
Default: []



## API REFERENCE

### 6.1 annize namespace

#### 6.1.1 Subpackages

`annize.aux` package

Submodules

`annize.aux._project_infos` module

`annize.callconv` package

**exception** `annize.callconv.MultipleValuesForSingleArgumentError` (*argname*)

Bases: `TypeError`

Parameters **argname** (*str*) –

**class** `annize.callconv.ParametersFromChildNodesInfos` (*parameter\_name*: *str*,  
*multiple*: *Union[bool, NoneType]*, *if\_passing*:  
*Union[Callable[[Any], bool], NoneType]*, *if\_typehint\_matches*:  
*bool*)

Bases: `object`

Parameters

- **parameter\_name** (*str*) –
- **multiple** (*Optional[bool]*) –
- **if\_passing** (*Optional[Callable[[Any], bool]]*) –
- **if\_typehint\_matches** (*bool*) –

Return type `None`

**if\_passing**: `Optional[Callable[[Any], bool]]`

**if\_typehint\_matches**: `bool`

**multiple**: `Optional[bool]`

**parameter\_name**: `str`

```

class annize.callconv._CallHelper
    Bases: object

    static _CallHelper__convert_kwargs_from_string (argumentinfos, args, kwargs)
    static _CallHelper__shift_args_to_kwargs (calltype, argumentinfos, args, kwargs)
    classmethod call (calltype, args, kwargs)

    Parameters
        • args (Iterable) –
        • kwargs (Dict) –

annize.callconv.always_true (__)

    Parameters _ (Any) –
    Return type bool

annize.callconv.call (calltype, args, kwargs)

    Parameters
        • args (Iterable) –
        • kwargs (Dict) –

annize.callconv.get_parameters_from_childnodes_infos (calltype)

    Parameters calltype (Type) –
    Return type List[annize.callconv.ParametersFromChildNodesInfos]

annize.callconv.parameter_from_childnodes (paramname, multiple=None, if_passing=None,
                                          if_typehint_matches=True)

    Parameters
        • paramname (str) –
        • multiple (Optional[bool]) –
        • if_passing (Optional[Callable[[Any], bool]]) –
        • if_typehint_matches (bool) –

```

## Submodules

### annize.callconv.typehints module

```

class annize.callconv.typehints.ListTypeInfo (name, innertypeinfo)
    Bases: annize.callconv.typehints.TypeInfo

    Parameters name (str) –

    property inner

class annize.callconv.typehints.TypeInfo (name, resolvedtype, constructfromstringfct)
    Bases: object

    Parameters
        • name (str) –
        • resolvedtype (Type) –

```

- **constructfromstringfct** (*Optional*[*Callable*[[*str*], *Any*]]) –

**construct\_from\_string** (*strvalue*)

Parameters *strvalue* (*str*) –

Return type *Any*

**property is\_constructable\_from\_string**

**property name**

**property resolved\_type**

**annize.callconv.typehints.get\_arguments\_infos** (\*, *for\_callable*)

Parameters **for\_callable** (*Callable*) –

Return type *Dict*[*str*, *annize.callconv.typehints.TypeInfo*]

**annize.callconv.typehints.get\_typeinfo** (*for\_type*)

Return type *annize.callconv.typehints.TypeInfo*

## annize.context package

**class** *annize.context.Context*

Bases: *object*

**\_METADATA\_IS\_TOPLEVEL\_OBJECT** = *'annize..is\_toplevel\_object'*

**\_\_object\_metadata\_dict** (*obj*)

Parameters *obj* (*Any*) –

Return type *Dict*[*str*, *Optional*[*Any*]]

**\_\_object\_raw\_name** (*obj*)

Parameters *obj* (*Any*) –

Return type *str*

**\_\_put\_object** (*name*, *obj*)

Parameters

- **name** (*str*) –
- **obj** (*Any*) –

Return type *None*

**add\_object** (*obj*)

Parameters *obj* (*Any*) –

Return type *str*

**get\_object\_metadata** (*obj*, *key*, *defaultvalue=None*)

Parameters

- **obj** (*Any*) –
- **key** (*str*) –
- **defaultvalue** (*Optional*[*Any*]) –

**Return type** Optional[Any]

**is\_friendly\_name** (*name*)

**Parameters** **name** (*str*) –

**Return type** bool

**is\_toplevel\_object** (*obj*)

**Parameters** **obj** (*Any*) –

**Return type** bool

**mark\_object\_as\_toplevel** (*obj*)

**Parameters** **obj** (*Any*) –

**Return type** None

**object\_by\_name** (*name*, *defaultvalue=None*, \*, *create\_nonexistent=False*)

**Parameters**

- **name** (*str*) –
- **defaultvalue** (*Optional[Any]*) –
- **create\_nonexistent** (*bool*) –

**Return type** Optional[Any]

**object\_name** (*obj*)

**Parameters** **obj** (*Any*) –

**Return type** str

**object\_names** (*obj*)

**Parameters** **obj** (*Any*) –

**Return type** List[str]

**objects\_by\_type** (*objtype*, *toplevel\_only=True*)

**Parameters**

- **objtype** (*Type[\_T]*) –
- **toplevel\_only** (*bool*) –

**Return type** List[\_T]

**set\_object\_metadata** (*obj*, *key*, *value=None*)

**Parameters**

- **obj** (*Any*) –
- **key** (*str*) –
- **value** (*Optional[Any]*) –

**Return type** None

**set\_object\_name** (*obj*, *name*)

**Parameters**

- **obj** (*Any*) –

- **name** (*str*) –

**Return type** None

**exception** `annize.context.OutOfContextError`

Bases: `TypeError`

`annize.context.add_object` (*obj*)

**Parameters** *obj* (*Any*) –

**Return type** `str`

`annize.context.current` ()

**Return type** `annize.context.Context`

`annize.context.get_object_metadata` (*obj*, *key*, *defaultvalue=None*)

**Parameters**

- **obj** (*Any*) –
- **key** (*str*) –
- **defaultvalue** (*Optional* [*Any*]) –

**Return type** `Optional` [*Any*]

`annize.context.is_friendly_name` (*name*)

**Parameters** *name* (*str*) –

**Return type** `bool`

`annize.context.is_toplevel_object` (*obj*)

**Parameters** *obj* (*Any*) –

**Return type** `bool`

`annize.context.object_by_name` (*name*, *defaultvalue=None*, \*, *create\_nonexistent=False*)

**Parameters**

- **name** (*str*) –
- **defaultvalue** (*Optional* [*Any*]) –
- **create\_nonexistent** (*bool*) –

**Return type** `Optional` [*Any*]

`annize.context.object_name` (*obj*)

**Parameters** *obj* (*Any*) –

**Return type** `str`

`annize.context.object_names` (*obj*)

**Parameters** *obj* (*Any*) –

**Return type** `List` [*str*]

`annize.context.objects_by_type` (*objtype*, *toplevel\_only=True*)

**Parameters**

- **objtype** (*Type* [*\_T*]) –

- **toplevel\_only** (*bool*) –

**Return type** List[\_T]

`annize.context.set_object_metadata(obj, key, value=None)`

**Parameters**

- **obj** (*Any*) –
- **key** (*str*) –
- **value** (*Optional[Any]*) –

**Return type** None

`annize.context.set_object_name(obj, name)`

**Parameters**

- **obj** (*Any*) –
- **name** (*str*) –

**Return type** None

## annize.data package

### Submodules

#### annize.data.color module

**class** `annize.data.color.Color` (\*, *red, green, blue*)  
 Bases: `object`

**Parameters**

- **red** (*float*) –
- **green** (*float*) –
- **blue** (*float*) –

**property** `blue`

**property** `green`

**property** `html_color_spec`

**property** `red`

**scalehue** (\*, *brightness, saturation=None*)

**Parameters**

- **brightness** (*float*) –
- **saturation** (*Optional[float]*) –

**Return type** `annize.data.color.Color`

**transformed** (\*, *brightness=None, saturation=None*)

**Parameters**

- **brightness** (*Optional[float]*) –

- **saturation** (*Optional[float]*) –

Return type *annize.data.color.Color*

## annize.data.container module

```
class annize.data.container.Basket(*a,**b)
    Bases: list
```

## annize.data.unique module

```
class annize.data.unique.UniqueId(localid=None)
    Bases: object

    Parameters localid (Optional[str]) –

    __long_str (txt)

    Parameters txt (str) –

    Return type str

    __uniqueid_counter = 0
    __uniqueid_lock = <unlocked _thread.lock object>
    property long_str
    property processonly_long_str
    property processonly_short_str
    property short_str
```

## annize.data.version module

```
class annize.data.version.AbstractVersionPatternSegment
    Bases: object

    _name_anon (namep)

    Parameters namep (str) –

    Return type None

    _str_to_val (txt)

    Parameters txt (str) –

    Return type object

    regexp_string()

    Return type str

    property segment_names

    segments_tuples_to_text (segments_tuples)

    Parameters segments_tuples (TODO) –

    Return type str
```

```

class annize.data.version.ConcatenatedVersionPatternSegment (*, segments)
    Bases: annize.data.version.AbstractVersionPatternSegment

        Parameters segments (Iterable[annize.data.version.
            AbstractVersionPatternSegment]) –

    _name_anon (namep)

    regexp_string()

        Return type str

    property segment_names

    segments_tuples_to_text (segments_tuples)

        Parameters segments_tuples (TODO) –

        Return type str

class annize.data.version.NumericVersionPatternSegment (*, partname=None)
    Bases: annize.data.version.AbstractVersionPatternSegment

        Parameters partname (Optional[str]) –

    _name_anon (namep)

    _str_to_val (txt)

        Parameters txt (str) –

        Return type object

    property partname

    regexp_string()

        Return type str

    property segment_names

    segments_tuples_to_text (segments_tuples)

        Parameters segments_tuples (TODO) –

        Return type str

class annize.data.version.OptionalVersionPatternSegment (*, segments)
    Bases: annize.data.version.AbstractVersionPatternSegment

        Parameters segments (Iterable[annize.data.version.
            AbstractVersionPatternSegment]) –

    _name_anon (namep)

    regexp_string()

        Return type str

    property segment_names

    segments_tuples_to_text (segments_tuples)

        Parameters segments_tuples (TODO) –

        Return type str

class annize.data.version.SeparatorVersionPatternSegment (*, text)
    Bases: annize.data.version.AbstractVersionPatternSegment

```



```

    Parameters text (str) –
regexp_string()
    Return type str
segments_tuples_to_text (segments_tuples)
    Parameters segments_tuples (TODO) –
    Return type str
class annize.data.version.Version (*, text=None, pattern=None, **segment_values)
    Bases: object
    Parameters
        • text (str) –
        • pattern (annize.data.version.VersionPattern) –
    property pattern
    property segments_tuples
    property segments_values
    property text
class annize.data.version.VersionPattern (*, segments)
    Bases: object
    Parameters segments (Iterable[annize.data.version.AbstractVersionPatternSegment]) –
    property segment_names
    property segments
segments_tuples_to_text (segments_tuples)
    Parameters segments_tuples (TODO) –
    Return type str
text_to_segments_tuples (text)
    Parameters text (str) –
    Return type TODO

annize.features namespace

Subpackages

annize.features.changelog namespace

Submodules

annize.features.changelog.common module

class annize.features.changelog.common.ByVersionControlSystemCommitMessagesChangelog
    Bases: annize.features.changelog.common.Changelog

```

```

    _S_CHANGE = '##CHANGE:'
    _S_LABEL = '##LABEL:'

    property entries

class annize.features.changelog.common.Changelog(*, entries=())
    Bases: object

        Parameters entries (Iterable[annize.features.changelog.common.Entry]) –

    property entries

class annize.features.changelog.common.Entry(*, version=None, time=None, items)
    Bases: object

        Parameters

            • version (annize.data.version.Version) –

            • time (datetime.datetime) –

            • items (Iterable[annize.features.changelog.common.Item]) –

    property items
    property time
    property version

class annize.features.changelog.common.Item(*, text)
    Bases: object

        Parameters text (annize.i18n.TrStr) –

    property text

annize.features.changelog.common.__items_matcher(o)
annize.features.changelog.common.default_changelog()

    Return type annize.features.changelog.common.Changelog

```

## annize.features.datainjections namespace

### Submodules

#### annize.features.datainjections.common module

```

class annize.features.datainjections.common.Inject(*, injection, destination)
    Bases: object

        Parameters

            • injection (annize.features.datainjections.common.Injection) –

            • destination (annize.fs.FsEntry) –

class annize.features.datainjections.common.Injection
    Bases: annize.fs.AdHocFsEntry

    _abc_impl = <_abc_data object>

    _initialize()

```

`inject(destination)`

Parameters `destination` (`annize.fs.FsEntry`) –

## annize.features.datainjections.python module

```
class annize.features.datainjections.python.Injection(*,
                                                    file-
                                                    name='_project_infos.py',
                                                    version=None)
Bases: annize.features.datainjections.common.Injection
```

### Parameters

- `filename` (`str`) –
- `version` (`annize.data.version.Version`) –

`_abc_impl` = `<_abc_data object>`

`inject(destination)`

Parameters `destination` (`annize.fs.FsEntry`) –

## annize.features.dependencies namespace

### Submodules

## annize.features.dependencies.common module

```
class annize.features.dependencies.common.Artwork(*, kind=None, label, origin=None,
                                                    comment=None, license=None)
Bases: annize.features.dependencies.common.Dependency
```

### Parameters

- `origin` (`str`) –
- `comment` (`str`) –
- `license` (`str`) –

```
class annize.features.dependencies.common.Dependency(*, kind=None, label=None,
                                                    comment=None, icon=None,
                                                    importance=0)
```

Bases: `object`

### Parameters

- `kind` (`annize.features.dependencies.common.Kind`) –
- `label` (`str`) –
- `comment` (`str`) –
- `icon` (`str`) –
- `importance` (`int`) –

property `comment`

property `icon`

```

    property importance
    property kind
    property label
class annize.features.dependencies.common.GnuLinux(*, kind=None, comment=")
    Bases: annize.features.dependencies.common.Dependency
class annize.features.dependencies.common.Included(**b)
    Bases: annize.features.dependencies.common.Kind
class annize.features.dependencies.common.Kind(*, label, importance, flags=None)
    Bases: object

    Parameters
        • label (str) –
        • importance (int) –
        • flags (List[object]) –

    property flags
    property importance
    property label
class annize.features.dependencies.common.Recommended(**b)
    Bases: annize.features.dependencies.common.Kind
class annize.features.dependencies.common.Required(**b)
    Bases: annize.features.dependencies.common.Kind
annize.features.dependencies.common.dependencies_to_rst_text(dependencies)

    Parameters dependencies (List[annize.features.dependencies.common.
        Dependency]) –

    Return type str

```

## annize.features.dependencies.python module

```

class annize.features.dependencies.python.Python(*, version, kind=None)
    Bases: annize.features.dependencies.common.Dependency

    Parameters
        • version (str) –
        • kind (annize.features.dependencies.common.Kind) –

```

## annize.features.distributables namespace

### Subpackages

## annize.features.distributables.store namespace

### Submodules

## annize.features.distributables.store.pypi module

```
class annize.features.distributables.store.pypi.Connection(*, token)
    Bases: object

    Parameters token (str) –

    property token

class annize.features.distributables.store.pypi.Upload(*, source, connection)
    Bases: object

    Parameters

    • source (annize.fs.FsEntry) –

    • connection (annize.features.distributables.store.pypi.
      Connection) –
```

### Submodules

## annize.features.distributables.common module

```
class annize.features.distributables.common.File
    Bases: annize.fs.FsEntry

    __abc_impl = <_abc_data object>

class annize.features.distributables.common.Group(*, title, files, description="", pack-
    agestore=None)
    Bases: object

    Parameters

    • title (str) –

    • files (List[annize.fs.FsEntry]) –

    • description (str) –

    • packagestore (annize.features.distributables.common.
      PackageStore) –

    __get_files_from_packagestore()

    Return type Iterable[annize.fs.FsEntry]

    __getname()

    Return type str

    __store_files_to_packagestore()
```

Return type None

property description

get\_files()

Return type List[*annize.fs.FsEntry*]

property packagestore

property title

**class** *annize.features.distributables.common.PackageStore*

Bases: object

get\_package(\*, name)

Parameters name (str) –

Return type Optional[Iterable[*annize.fs.FsEntry*]]

get\_package\_history(\*, name, limit=3)

Parameters

- name (str) –
- limit (int) –

Return type Iterable[Iterable[*annize.fs.FsEntry*]]

store\_package(fsentries, \*, name)

Parameters

- fsentries (Iterable[*annize.fs.FsEntry*]) –
- name (str) –

Return type None

## ***annize.features.distributables.debian* module**

*annize.features.distributables.debian.AdministrationUtilitiesSection*

alias of *annize.features.distributables.debian.\_debian\_section.<locals>.ASection*

*annize.features.distributables.debian.ApplicationsAccessibilityCategory*

alias of *annize.features.distributables.debian.\_debian\_category.<locals>.ACategory*

*annize.features.distributables.debian.ApplicationsAmateurradioCategory*

alias of *annize.features.distributables.debian.\_debian\_category.<locals>.ACategory*

*annize.features.distributables.debian.ApplicationsDatamanagementCategory*

alias of *annize.features.distributables.debian.\_debian\_category.<locals>.ACategory*

*annize.features.distributables.debian.ApplicationsEditorsCategory*

alias of *annize.features.distributables.debian.\_debian\_category.<locals>.ACategory*

```

annize.features.distributables.debian.ApplicationsEducationCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsEmulatorsCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsFilemanagementCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsGraphicsCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsMobiledevicesCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsNetworkCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsNetworkCommunicationCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsNetworkFiletransferCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsNetworkMonitoringCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsNetworkWebbrowsingCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsNetworkWebnewsCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsOfficeCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsProgrammingCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsProjectmanagementCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsScienceAstronomyCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

```

```

annize.features.distributables.debian.ApplicationsScienceBiologyCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsScienceCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsScienceChemistryCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsScienceDataanalysisCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsScienceElectronicsCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsScienceEngineeringCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsScienceGeoscienceCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsScienceMathematicsCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsScienceMedicineCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsSciencePhysicsCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsScienceSocialCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsShellsCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsSoundsCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsSystemAdministrationCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsSystemCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

```



```

annize.features.distributables.debian.ApplicationsSystemHardwareCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsSystemLanguageenvironmentCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsSystemMonitoringCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsSystemPackagemanagementCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsSystemSecurityCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsTerminalemulatorsCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsTextCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsTvandradiocategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsVideoCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsViewersCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ApplicationsWebdevelopmentCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

class annize.features.distributables.debian.Category(*, debian_name, freedesk-
                                                    top_name)
    Bases: object

    Parameters

    • debian_name (str) –

    • freedesktop_name (str) –

    property debian_name

    property freedesktop_name

annize.features.distributables.debian.CommunicationProgramsSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

```

```

annize.features.distributables.debian.DatabasesSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.DebianInstallerUdebPackagesSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.DebugPackagesSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.DevelopmentSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.DocumentationSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.EditorsSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.EducationSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.ElectronicsSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.EmbeddedSoftwareSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

class annize.features.distributables.debian.ExecutableLink (*, path, name=None)
    Bases: object

        Parameters

            • path (str) –
            • name (str) –

        property name
        property path

annize.features.distributables.debian.FontsSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.GamesActionCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.GamesAdventureCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

```

```

annize.features.distributables.debian.GamesBlocksCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.GamesBoardCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.GamesCardCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.GamesPuzzlesCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.GamesSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.GamesSimulationCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.GamesStrategyCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.GamesToolsCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.GamesToysCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.GnomeSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.GnuRSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.GnustepSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.GraphicsSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.HamRadioSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.HaskellSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

```

```

annize.features.distributables.debian.HelpCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.InterpretersSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.IntrospectionSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.JavaSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.JavascriptSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.KdeSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.KernelsSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.LanguagePacksSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.LibrariesSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.LibraryDevelopmentSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.LispSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.MailSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.MathematicsSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

class annize.features.distributables.debian.MenuEntry (*, name, title, category, com-
                                                         mand, is_gui, icon=None)

    Bases: object

    Parameters

    • name (str) –

    • title (str) –

    • category (annize.features.distributables.debian.Category) –

```

- **command**(*str*) –
- **is\_gui**(*bool*) –
- **icon**(*annize.fs.FsEntry*) –

**property category**

**property command**

**property icon**

**property is\_gui**

**property name**

**property title**

**annize.features.distributables.debian.MetaPackagesSection**

alias of `annize.features.distributables.debian._debian_section.<locals>.ASection`

**annize.features.distributables.debian.MiscellaneousSection**

alias of `annize.features.distributables.debian._debian_section.<locals>.ASection`

**annize.features.distributables.debian.MonoCliSection**

alias of `annize.features.distributables.debian._debian_section.<locals>.ASection`

**annize.features.distributables.debian.NetworkSection**

alias of `annize.features.distributables.debian._debian_section.<locals>.ASection`

**annize.features.distributables.debian.NewsgroupsSection**

alias of `annize.features.distributables.debian._debian_section.<locals>.ASection`

**annize.features.distributables.debian.OcamlSection**

alias of `annize.features.distributables.debian._debian_section.<locals>.ASection`

**annize.features.distributables.debian.OldLibrariesSection**

alias of `annize.features.distributables.debian._debian_section.<locals>.ASection`

**annize.features.distributables.debian.OtherOSsAndFSsSection**

alias of `annize.features.distributables.debian._debian_section.<locals>.ASection`

**class** `annize.features.distributables.debian.Package`(\*, *source*, *menuentries*, *executable\_links*, *package\_name*=None, *description*=None, *summary*=None, *section*=None, *homepage\_url*=None, *version*=None, *documentation*=None, *author*=None, *prerm*="", *postinst*="", *architecture*='all')

Bases: `annize.fs.AdHocFsEntry`

#### Parameters

- **source**(*annize.fs.FsEntry*) –

- **menuentries** (*List[annize.features.distributables.debian.MenuEntry]*) –
- **executable\_links** (*List[annize.features.distributables.debian.ExecutableLink]*) –
- **packagename** (*Optional[str]*) –
- **description** (*Optional[str]*) –
- **summary** (*Optional[str]*) –
- **section** (*annize.features.distributables.debian.Section*) –
- **homepage\_url** (*Optional[str]*) –
- **version** (*annize.data.version.Version*) –
- **documentation** (*annize.fs.FsEntry*) –
- **author** (*annize.features.authors.Author*) –
- **prerm** (*str*) –
- **postinst** (*str*) –
- **architecture** (*str*) –

```
class _BuildInfo (source: annize.fs.FsEntry, executable_links:
    List[annize.features.distributables.debian.ExecutableLink], menuen-
    tries: List[annize.features.distributables.debian.MenuEntry], services:
    List[annize.features.distributables.debian.ServiceDescription], descrip-
    tion: str, name: str, version: annize.data.version.Version, homepage:
    str, author: annize.features.authors.Author, licensename: str, section:
    Union[annize.features.distributables.debian.Section, NoneType], summary:
    str, documentation_source: Union[annize.fs.FsEntry, NoneType], prerm: str,
    postinst: str, architecture: str, authorstring: str = None, pkgrootpath: str
    = None, pkgpath_debian: str = None, pkgpath_documentation: str = None,
    pkgpath_pixmap: str = None, pkgpath_usrbin: str = None, config_files:
    List[str] = None, pkgsize: int = None, result: annize.fs.FsEntry = None)
```

Bases: object

```
_BuildInfo(source: annize.fs.FsEntry, executable_links: List[annize.features.distributables.debian.ExecutableLink],
menuentries: List[annize.features.distributables.debian.MenuEntry], services:
List[annize.features.distributables.debian.ServiceDescription], description: str, name: str, version:
annize.data.version.Version, homepage: str, author: annize.features.authors.Author, licensename: str,
section: Union[annize.features.distributables.debian.Section, NoneType], summary: str, documenta-
tion_source: Union[annize.fs.FsEntry, NoneType], prerm: str, postinst: str, architecture: str, authorstring:
str = None, pkgrootpath: str = None, pkgpath_debian: str = None, pkgpath_documentation: str = None,
pkgpath_pixmap: str = None, pkgpath_usrbin: str = None, config_files: List[str] = None, pkgsize: int =
None, result: annize.fs.FsEntry = None)
```

#### Parameters

- **source** (*annize.fs.FsEntry*) –
- **executable\_links** (*List[annize.features.distributables.debian.ExecutableLink]*) –
- **menuentries** (*List[annize.features.distributables.debian.MenuEntry]*) –
- **services** (*List[annize.features.distributables.debian.ServiceDescription]*) –

- **description** (*str*) –
- **name** (*str*) –
- **version** (*annize.data.version.Version*) –
- **homepage** (*str*) –
- **author** (*annize.features.authors.Author*) –
- **licensename** (*str*) –
- **section** (*Optional[annize.features.distributables.debian.Section]*) –
- **summary** (*str*) –
- **documentation\_source** (*Optional[annize.fs.FsEntry]*) –
- **prerm** (*str*) –
- **postinst** (*str*) –
- **architecture** (*str*) –
- **authorstring** (*str*) –
- **pkgrootpath** (*str*) –
- **pkgpath\_debian** (*str*) –
- **pkgpath\_documentation** (*str*) –
- **pkgpath\_pixmap** (*str*) –
- **pkgpath\_usrbin** (*str*) –
- **config\_files** (*List[str]*) –
- **pkgsize** (*int*) –
- **result** (*annize.fs.FsEntry*) –

Return type *None*

```
architecture: str
author: annize.features.authors.Author
authorstring: str = None
config_files: List[str] = None
description: str
documentation_source: Optional[annize.fs.FsEntry]
executable_links: List[annize.features.distributables.debian.ExecutableLink]
homepage: str
licensename: str
menuentries: List[annize.features.distributables.debian.MenuEntry]
name: str
pkgpath_debian: str = None
pkgpath_documentation: str = None
```

```

    pkgpath_pixmap: str = None
    pkgpath_usrbin: str = None
    pkgrootpath: str = None
    pkgsize: int = None
    postinst: str
    prerm: str
    result: annize.fs.FsEntry = None
    section: Optional[annize.features.distributables.debian.Section]
    services: List[annize.features.distributables.debian.ServiceDescription]
    source: annize.fs.FsEntry
    summary: str
    version: annize.data.version.Version
_abc_impl = <_abc_data object>
_initialize()
classmethod _mkpackage (build)
    Parameters build (annize.features.distributables.debian.Package.
        _BuildInfo)-
    Return type annize.fs.FsEntry
classmethod _mkpackage_correctbuildsourcepermissions (build)
    Parameters build (annize.features.distributables.debian.Package.
        _BuildInfo)-
    Return type None
classmethod _mkpackage_determinesize (build)
    Parameters build (annize.features.distributables.debian.Package.
        _BuildInfo)-
    Return type None
classmethod _mkpackage_dpkg (build)
    Parameters build (annize.features.distributables.debian.Package.
        _BuildInfo)-
    Return type None
classmethod _mkpackage_mkchangelog (build)
    Parameters build (annize.features.distributables.debian.Package.
        _BuildInfo)-
    Return type None
classmethod _mkpackage_mkcopyright (build)
    Parameters build (annize.features.distributables.debian.Package.
        _BuildInfo)-
    Return type None

```



```

classmethod _mkpackage_mkdebianconffilesfile (build)

    Parameters build      (annize.features.distributables.debian.Package.
                           _BuildInfo)-

    Return type None

classmethod _mkpackage_mkdebiancontrolfile (build)

    Parameters build      (annize.features.distributables.debian.Package.
                           _BuildInfo)-

    Return type None

classmethod _mkpackage_mkexeclinks (build)

    Parameters build      (annize.features.distributables.debian.Package.
                           _BuildInfo)-

    Return type None

classmethod _mkpackage_mkmenuentries (build)

    Parameters build      (annize.features.distributables.debian.Package.
                           _BuildInfo)-

    Return type None

classmethod _mkpackage_mkprepostcmds (build)

    Parameters build      (annize.features.distributables.debian.Package.
                           _BuildInfo)-

    Return type None

classmethod _mkpackage_mkservices (build)

    Parameters build      (annize.features.distributables.debian.Package.
                           _BuildInfo)-

    Return type None

classmethod _mkpackage_prepareinfos (build, tmpdir)

    Parameters

        • build      (annize.features.distributables.debian.Package.
                     _BuildInfo)-

        • tmpdir (str)-

    Return type None

annize.features.distributables.debian.PerlSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.PhpSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.PythonSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

```

```

annize.features.distributables.debian.RubySection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.RustSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.ScienceSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.ScreenLockingCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

annize.features.distributables.debian.ScreenSavingCategory
    alias of annize.features.distributables.debian._debian_category.<locals>.
    ACategory

class annize.features.distributables.debian.Section(*name)
    Bases: object

    property name

class annize.features.distributables.debian.ServiceDescription(name,      com-
                                                                mand)
    Bases: object

    Description for Debian services to be included in a package.

    Parameters

    • name (str) – The display name.

    • command (str) – The command to be executed.

annize.features.distributables.debian.ShellsSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.SoundSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.TasksSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.TexSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.TextProcessingSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.UtilitiesSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

```

```

annize.features.distributables.debian.VersionControlSystemsSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.VideoSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.VirtualPackagesSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.WebServersSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.WebSoftwareSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.XWindowSystemSoftwareSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.XfceSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian.ZopePloneFrameworkSection
    alias of annize.features.distributables.debian._debian_section.<locals>.
    ASection

annize.features.distributables.debian._debian_category (debian_name, freedesk-
                                                         top_name)

```

#### Parameters

- **debian\_name** (*str*) –
- **freedesktop\_name** (*str*) –

**Return type** Type[*annize.features.distributables.debian.Category*]

```

annize.features.distributables.debian._debian_section (name)

```

**Parameters** **name** (*str*) –

**Return type** Type[*annize.features.distributables.debian.Section*]

### annize.features.distributables.flatpak module

```

class annize.features.distributables.flatpak.EnvironmentVariable (**b)
    Bases: object

class annize.features.distributables.flatpak.Filesystem (**b)
    Bases: object

class annize.features.distributables.flatpak.FlatpakImage (*, source, pack-
                                                         age_name, sock-
                                                         ets='x11', filesys-
                                                         tems='home',
                                                         shares='network')

```

Bases: `annize.fs.AdHocFsEntry`

#### Parameters

- **source** (`annize.fs.FsEntry`) –
- **package\_name** (`str`) –
- **sockets** (`Iterable[str]`) –
- **filesystems** (`Iterable[str]`) –
- **shares** (`Iterable[str]`) –

```
class _BuildInfo(source: annize.fs.FsEntry, name: str, sdk: str, platform: str, kitversion: Union[str, NoneType], command: Union[str, NoneType], sockets: Iterable[str], filesystems: Iterable[str], shares: Iterable[str], menu_entries: Iterable[annize.features.distributables.flatpak.MenuEntry], environment: Dict[str, str], pkgrootpath: str = None, pkgpath_share: str = None, pkgpath_share_applications: str = None, pkgpath_share_icons: str = None, result: annize.fs.FsEntry = None)
```

Bases: `object`

`_BuildInfo`(source: `annize.fs.FsEntry`, name: `str`, sdk: `str`, platform: `str`, kitversion: `Union[str, NoneType]`, command: `Union[str, NoneType]`, sockets: `Iterable[str]`, filesystems: `Iterable[str]`, shares: `Iterable[str]`, menu\_entries: `Iterable[annize.features.distributables.flatpak.MenuEntry]`, environment: `Dict[str, str]`, pkgrootpath: `str = None`, pkgpath\_share: `str = None`, pkgpath\_share\_applications: `str = None`, pkgpath\_share\_icons: `str = None`, result: `annize.fs.FsEntry = None`)

#### Parameters

- **source** (`annize.fs.FsEntry`) –
- **name** (`str`) –
- **sdk** (`str`) –
- **platform** (`str`) –
- **kitversion** (`Optional[str]`) –
- **command** (`Optional[str]`) –
- **sockets** (`Iterable[str]`) –
- **filesystems** (`Iterable[str]`) –
- **shares** (`Iterable[str]`) –
- **menu\_entries** (`Iterable[annize.features.distributables.flatpak.MenuEntry]`) –
- **environment** (`Dict[str, str]`) –
- **pkgrootpath** (`str`) –
- **pkgpath\_share** (`str`) –
- **pkgpath\_share\_applications** (`str`) –
- **pkgpath\_share\_icons** (`str`) –
- **result** (`annize.fs.FsEntry`) –

Return type `None`

**command:** `Optional[str]`

```

environment: Dict[str, str]
filesystems: Iterable[str]
kitversion: Optional[str]
menu_entries: Iterable[annize.features.distributables.flatpak.MenuEntry]
name: str
pkgpath_share: str = None
pkgpath_share_applications: str = None
pkgpath_share_icons: str = None
pkgrootpath: str = None
platform: str
result: annize.fs.FsEntry = None
sdk: str
shares: Iterable[str]
sockets: Iterable[str]
source: annize.fs.FsEntry
_abc_impl = <_abc_data object>
_initialize()
classmethod _mkpackage (build)
    Parameters build (annize.features.distributables.flatpak.
        FlatpakImage._BuildInfo) -
    Return type annize.fs.FsEntry
classmethod _mkpackage_applysource (build)
    Parameters build (annize.features.distributables.flatpak.
        FlatpakImage._BuildInfo) -
    Return type None
classmethod _mkpackage_flatpak_build_export (build)
    Parameters build (annize.features.distributables.flatpak.
        FlatpakImage._BuildInfo) -
    Return type None
classmethod _mkpackage_flatpak_build_finish (build)
    Parameters build (annize.features.distributables.flatpak.
        FlatpakImage._BuildInfo) -
    Return type None
classmethod _mkpackage_flatpak_build_init (build)
    Parameters build (annize.features.distributables.flatpak.
        FlatpakImage._BuildInfo) -
    Return type None
classmethod _mkpackage_prepareinfos (build, tmpdir)

```

#### Parameters

- **build** (`annize.features.distributables.flatpak.FlatpakImage._BuildInfo`) –
- **tmpdir** (`str`) –

Return type: `None`

**classmethod** `_mkpackage_share` (`build`)

**Parameters** `build` (`annize.features.distributables.flatpak.FlatpakImage._BuildInfo`) –

Return type: `None`

```
class annize.features.distributables.flatpak.FlatpakrefFile(*,  refname=None,
                                                         package_name,
                                                         title=None,
                                                         branch='master',
                                                         run-
                                                         time_repository_url=None,
                                                         gpgkey=None,
                                                         repository)
```

Bases: `annize.fs.AdHocFsEntry`

#### Parameters

- **refname** (`str`) –
- **package\_name** (`str`) –
- **title** (`str`) –
- **branch** (`str`) –
- **runtime\_repository\_url** (`str`) –
- **gpgkey** (`bytes`) –
- **repository** (`annize.features.distributables.flatpak.Repository`) –

`_abc_impl = <_abc_data object>`

`_initialize()`

```
class annize.features.distributables.flatpak.GpgFile(*,  refname=None,
                                                         gpgkey=None)
```

Bases: `annize.fs.AdHocFsEntry`

#### Parameters

- **refname** (`str`) –
- **gpgkey** (`bytes`) –

`_abc_impl = <_abc_data object>`

`_initialize()`

```
class annize.features.distributables.flatpak.Group(*,  source,  title,  repos-
                                                         itory,  package_name,
                                                         project_short_hint_name=None,
                                                         package_short_name=None)
```

Bases: `annize.features.distributables.common.Group`

#### Parameters

- **source** (`annize.fs.FsEntry`) –
- **title** (`str`) –
- **repository** (`annize.features.distributables.flatpak.Repository`) –
- **package\_name** (`str`) –
- **project\_short\_hint\_name** (`str`) –
- **package\_short\_name** (`str`) –

#### property description

`get_files()`

```
class annize.features.distributables.flatpak.LocalRepository(*, public_url,
                                                             friendly_name_suggestion=None,
                                                             up-
                                                             load_path=None,
                                                             up-
                                                             load_fsentry=None)
```

Bases: `annize.features.distributables.flatpak.Repository`

#### Parameters

- **public\_url** (`str`) –
- **friendly\_name\_suggestion** (`str`) –
- **upload\_path** (`str`) –
- **upload\_fsentry** (`annize.fs.FsEntry`) –

`upload(source)`

Parameters **source** (`annize.fs.FsEntry`) –

```
class annize.features.distributables.flatpak.MenuEntry(*, name, title, category,
                                                         command, gui, icon)
```

Bases: `object`

#### Parameters

- **name** (`str`) –
- **title** (`str`) –
- **category** (`Tuple`) –
- **command** (`str`) –
- **gui** (`bool`) –
- **icon** (`annize.fs.FsEntry`) –

```
class annize.features.distributables.flatpak.Repository(*, public_url,
                                                         friendly_name_suggestion=None)
```

Bases: `object`

#### Parameters

- **public\_url** (`str`) –
- **friendly\_name\_suggestion** (`str`) –

**property** `friendly_name_suggestion`

**property** `public_url`

**upload** (*source*)

**Parameters** `source` (`annize.fs.FsEntry`) –

**class** `annize.features.distributables.flatpak.Share` (\*\*b)

Bases: `object`

## `annize.features.distributables.pythonwheel` module

**class** `annize.features.distributables.pythonwheel.ExecutableLink` (\*, *linkname*,  
*modulename*,  
*methodname*,  
*is\_gui*)

Bases: `object`

### **Parameters**

- **linkname** (*str*) –
- **modulename** (*str*) –
- **methodname** (*str*) –
- **is\_gui** (*bool*) –

**property** `is_gui`

**property** `linkname`

**property** `methodname`

**property** `modulename`

**class** `annize.features.distributables.pythonwheel.Package` (\*, *source*, *executable\_links*, *packagename*=None,  
*description*=None,  
*homepage\_url*=None,  
*long\_description*=None,  
*version*=None, *keywords*=None, *license*=None,  
*author*=None)

Bases: `annize.fs.AdHocFsEntry`

### **Parameters**

- **source** (`annize.fs.FsEntry`) –
- **executable\_links** (*List* [`annize.features.distributables.pythonwheel.ExecutableLink`]) –
- **packagename** (*Optional* [*str*]) –
- **description** (*Optional* [*str*]) –
- **homepage\_url** (*Optional* [*str*]) –
- **long\_description** (*Optional* [*str*]) –



- **version** (`annize.data.version.Version`) –
- **keywords** (`annize.features.base.Keywords`) –
- **license** (`annize.features.licensing.License`) –
- **author** (`annize.features.authors.Author`) –

```
class _BuildInfo (source: annize.fs.FsEntry, description: str, long_description: str, keywords:
    annize.features.base.Keywords, name: str, version: annize.data.version.Version,
    homepage: str, author: annize.features.authors.Author, li-
    cense: annize.features.licensing.License, executable_links:
    List[annize.features.distributables.pythonwheel.ExecutableLink], pkgroot-
    path: str = None, pkgpath_setuppy: str = None, setuppy_conf: Dict[str,
    Union[Any, NoneType]] = None, result: annize.fs.FsEntry = None)
```

Bases: `object`

`_BuildInfo`(source: `annize.fs.FsEntry`, description: `str`, long\_description: `str`, keywords: `annize.features.base.Keywords`, name: `str`, version: `annize.data.version.Version`, homepage: `str`, author: `annize.features.authors.Author`, license: `annize.features.licensing.License`, executable\_links: `List[annize.features.distributables.pythonwheel.ExecutableLink]`, pkgrootpath: `str = None`, pkgpath\_setuppy: `str = None`, setuppy\_conf: `Dict[str, Union[Any, NoneType]] = None`, result: `annize.fs.FsEntry = None`)

#### Parameters

- **source** (`annize.fs.FsEntry`) –
- **description** (`str`) –
- **long\_description** (`str`) –
- **keywords** (`annize.features.base.Keywords`) –
- **name** (`str`) –
- **version** (`annize.data.version.Version`) –
- **homepage** (`str`) –
- **author** (`annize.features.authors.Author`) –
- **license** (`annize.features.licensing.License`) –
- **executable\_links** (`List[annize.features.distributables.pythonwheel.ExecutableLink]`) –
- **pkgrootpath** (`str`) –
- **pkgpath\_setuppy** (`str`) –
- **setuppy\_conf** (`Dict[str, Optional[Any]]`) –
- **result** (`annize.fs.FsEntry`) –

Return type `None`

**author:** `annize.features.authors.Author`

**description:** `str`

**executable\_links:** `List[annize.features.distributables.pythonwheel.ExecutableLink]`

**homepage:** `str`

**keywords:** `annize.features.base.Keywords`

**license:** `annize.features.licensing.License`

```

    long_description: str
    name: str
    pkgpath_setuppy: str = None
    pkgrootpath: str = None
    result: annize.fs.FsEntry = None
    setuppy_conf: Dict[str, Optional[Any]] = None
    source: annize.fs.FsEntry
    version: annize.data.version.Version

_abc_impl = <_abc_data object>
_initialize()

classmethod _mkpackage (build)
    Parameters build (annize.features.distributables.pythonwheel.
        Package._BuildInfo)-
    Return type annize.fs.FsEntry

classmethod _mkpackage_bdist_wheel (build)
    Parameters build (annize.features.distributables.pythonwheel.
        Package._BuildInfo)-
    Return type None

classmethod _mkpackage_mkexeclinks (build)
    Parameters build (annize.features.distributables.pythonwheel.
        Package._BuildInfo)-
    Return type None

classmethod _mkpackage_mkmanifestin (build)
    Parameters build (annize.features.distributables.pythonwheel.
        Package._BuildInfo)-
    Return type None

classmethod _mkpackage_mksetuppyconf (build)
    Parameters build (annize.features.distributables.pythonwheel.
        Package._BuildInfo)-
    Return type None

classmethod _mkpackage_prepareinfos (build, tmpdir)
    Parameters
        • build (annize.features.distributables.pythonwheel.Package.
            _BuildInfo)-
        • tmpdir (str)-
    Return type None

classmethod _mkpackage_setuppyconf_classifiers (build)

```

**Parameters** **build** (annize.features.distributables.pythonwheel.  
Package.\_BuildInfo)–

**Return type** None

**classmethod** **\_mkpackage\_setuptoolsconf\_install\_requires** (build)

**Parameters** **build** (annize.features.distributables.pythonwheel.  
Package.\_BuildInfo)–

**Return type** None

**classmethod** **\_mkpackage\_setuptoolsconf\_prepare** (build)

**Parameters** **build** (annize.features.distributables.pythonwheel.  
Package.\_BuildInfo)–

**Return type** None

## annize.features.distributables.tar module

**class** annize.features.distributables.tar.**Package** (\*, packagename=None, pack-  
agenamepostfix=None, source,  
version=None, license=None, docu-  
mentation=None)

Bases: *annize.fs.AdHocFsEntry*

### Parameters

- **packagename** (*Optional[str]*)–
- **packagenamepostfix** (*Optional[str]*)–
- **source** (*annize.fs.FsEntry*)–
- **version** (*annize.data.version.Version*)–
- **license** (*annize.features.licensing.License*)–
- **documentation** (*annize.fs.FsEntry*)–

**\_abc\_impl** = <\_abc\_data object>

**\_initialize** ()

## annize.features.documentation namespace

### Subpackages

### annize.features.documentation.sphinx namespace

### Subpackages

### annize.features.documentation.sphinx.output namespace

### Submodules

## annize.features.documentation.sphinx.output.common module

**class** `annize.features.documentation.sphinx.output.common.OutputGenerator` (*outputspec*)  
 Bases: `abc.ABC`

Base class for documentation output specifications. See `render()`.

`_abc_impl = <_abc_data object>`

**abstract** `formatname()`

Returns the Sphinx format name.

**Return type** `str`

**abstract classmethod** `is_compatible_for` (*outputspec*)

**Parameters** `outputspec` (`annize.features.documentation.common.OutputSpec`) –

**Return type** `bool`

**multilanguage\_frame** (*document*)

**Parameters** `document` (`documentationsphinx.Document`) –

**Return type** `annize.features.documentation.common._DocumentGenerateAllLanguagesResult`

**property** `outputspec`

**postproc** (*preresult*)

**Parameters** `preresult` (`annize.fs.FsEntry`) –

**Return type** `annize.fs.FsEntry`

**prepare\_generate** (*geninfo*)

**Parameters** `geninfo` (`documentationsphinx.Document.GenerateInfo`) –

**Return type** `None`

`annize.features.documentation.sphinx.output.common.find_output_generator_for_outputspec` (*outputspec*)

`annize.features.documentation.sphinx.output.common.register_output_generator` (*outputgenerator*)

## annize.features.documentation.sphinx.output.html module

**class** `annize.features.documentation.sphinx.output.html.HtmlOutputGenerator` (*outputspec*)  
 Bases: `annize.features.documentation.sphinx.output.common.OutputGenerator`

HTML documentation output.

`_abc_impl = <_abc_data object>`

**formatname** ()

Returns the Sphinx format name.

**classmethod** `is_compatible_for` (*outputspec*)

**multilanguage\_frame** (*document*)

**prepare\_generate** (*geninfo*)

```
class annize.features.documentation.sphinx.output.html.HtmlOutputSpec(*,
                                                                    is_homepage=False,
                                                                    ti-
                                                                    tle=None,
                                                                    short_title=None,
                                                                    short_desc=None,
                                                                    theme=None,
                                                                    mas-
                                                                    ter-
                                                                    link=None,
                                                                    logo_image=None,
                                                                    back-
                                                                    ground_image=None)
```

Bases: *annize.features.documentation.common.HtmlOutputSpec*

#### Parameters

- **path** – The output path.
- **theme** (*Optional[str]*) – The sphinx html theme name.
- **homagemode** – If to render output for a homepage with slight different stylings and behaviors.
- **title** (*Optional[str]*) – The html title.
- **short\_title** (*Optional[str]*) – The short html title.
- **short\_desc** (*Optional[str]*) – The short description. Ignored by most themes.
- **masterlink** (*Optional[str]*) – Url that overrides the target of the main heading (which is also a link).
- **is\_homepage** (*bool*) –
- **logo\_image** (*Optional[annize.fs.FsEntry]*) –
- **background\_image** (*Optional[annize.fs.FsEntry]*) –

**\_abc\_impl** = *<\_abc\_data object>*

**property background\_image**

**property logo\_image**

**property masterlink**

**property short\_desc**

**property short\_title**

**property theme**

### annize.features.documentation.sphinx.output.pdf module

```
class annize.features.documentation.sphinx.output.pdf.PdfOutputGenerator (outputspec)
    Bases: annize.features.documentation.sphinx.output.common.OutputGenerator
    PDF documentation output.
    _abc_impl = <_abc_data object>
    formatname ()
        Returns the Sphinx format name.
    classmethod is_compatible_for (outputspec)
    postproc (preresult)
    resultpath (projectname, buildpath)
```

### annize.features.documentation.sphinx.output.plaintext module

```
class annize.features.documentation.sphinx.output.plaintext.PlaintextOutputGenerator (outputspec)
    Bases: annize.features.documentation.sphinx.output.common.OutputGenerator
    Plaintext documentation output.
    _abc_impl = <_abc_data object>
    formatname ()
        Returns the Sphinx format name.
    classmethod is_compatible_for (outputspec)
```

## Submodules

### annize.features.documentation.sphinx.\_utils module

```
annize.features.documentation.sphinx._utils.serialize_for_confpy (obj)
    Parameters obj (Optional[Any]) –
    Return type str
```

## annize.features.documentation.sphinx.common module

```
class annize.features.documentation.sphinx.common.AboutProjectDocument (*,
                                                                    pro-
                                                                    ject-
                                                                    name=None,
                                                                    au-
                                                                    thors=None,
                                                                    ver-
                                                                    sion=None,
                                                                    re-
                                                                    lease=None,
                                                                    de-
                                                                    pen-
                                                                    den-
                                                                    cies,
                                                                    project_version=None)
```

Bases: *annize.features.documentation.sphinx.common.Document*

### Parameters

- **projectname** (*Optional[str]*) –
- **authors** (*Optional[str]*) –
- **version** (*Optional[str]*) –
- **release** (*Optional[str]*) –
- **dependencies** (*List[annize.features.dependencies.common.Dependency]*) –
- **project\_version** (*annize.data.version.Version*) –

**\_abc\_impl** = *<\_abc\_data object>*

**\_generate\_master\_doc** (*geninfo*)

**available\_languages** ()

**Return type** *List[str]*

```
class annize.features.documentation.sphinx.common.ApiReferenceDocument (*,
                                                                    lan-
                                                                    guage,
                                                                    head-
                                                                    ing=None,
                                                                    source=None)
```

Bases: *annize.features.documentation.sphinx.common.Document*

An api reference.

### Parameters

- **language** (*annize.features.documentation.sphinx.common.ApiReferenceLanguage*) –
- **heading** (*Optional[str]*) –
- **source** (*annize.fs.FsEntry*) –

**\_\_get\_refgeninfo** (*geninfo*)

```

    Parameters geninfo (annize.features.documentation.sphinx.common.
        Document.GenerateInfo)–

    _abc_impl = <_abc_data object>

    _generate_master_doc (geninfo)

    _prepare_generate (geninfo)

    available_languages ()

    Return type List[str]

```

**class** annize.features.documentation.sphinx.common.**ApiReferenceLanguage**

Bases: abc.ABC

Base class for a programming language in api references. See ApiReferencePiece.

**class** **ApiReferenceGenerateInfo** (geninfo, source, heading)

Bases: *annize.features.documentation.sphinx.common.Document.GenerateInfo*

**Parameters**

- **geninfo** (annize.features.documentation.sphinx.common. Document.GenerateInfo)–
- **source** (annize.fs.FsEntry)–
- **heading** (str)–

**property** heading

**property** source

\_abc\_impl = <\_abc\_data object>

**abstract** generate\_master\_doc (rgeninfo)

**Parameters** **rgeninfo** (annize.features.documentation.sphinx.common. ApiReferenceLanguage.ApiReferenceGenerateInfo)–

**Return type** str

**abstract** prepare\_generate (rgeninfo)

**Parameters** **rgeninfo** (annize.features.documentation.sphinx.common. ApiReferenceLanguage.ApiReferenceGenerateInfo)–

**Return type** None

**class** annize.features.documentation.sphinx.common.**ArgparseCommandLineInterfaceDocument** (\*,

*parse*  
*fac-*  
*tory,*  
*pro-*  
*gram*  
*name*  
*head*  
*ing=*  
*source*

Bases: *annize.features.documentation.sphinx.common.Document*

**Parameters**

- **parserfactory** (str)–
- **programname** (str)–



```

    • heading (Optional[str]) –
    • source (annize.fs.FsEntry) –
    _abc_impl = <_abc_data object>
    _generate_master_doc (geninfo)
    _prepare_generate (geninfo)
    available_languages ()

    Return type List[str]

class annize.features.documentation.sphinx.common.CompositeDocument (*,
                                                                    project-
                                                                    name=None,
                                                                    au-
                                                                    thors=None,
                                                                    ver-
                                                                    sion=None,
                                                                    re-
                                                                    lease=None,
                                                                    docu-
                                                                    ments)

Bases: annize.features.documentation.sphinx.common.Document

    Parameters

    • projectname (Optional[str]) –
    • authors (Optional[str]) –
    • version (Optional[str]) –
    • release (Optional[str]) –
    • documents (Iterable[annize.features.documentation.sphinx.
                   common.Document]) –

    __get_inner_generateinfo (geninfo, innerdoc)

    Parameters

    • geninfo (annize.features.documentation.sphinx.common.
               Document.GenerateInfo) –
    • innerdoc (annize.features.documentation.sphinx.common.
                 Document) –

    _abc_impl = <_abc_data object>
    _generate_master_doc (geninfo)
    _prepare_generate (geninfo)

class annize.features.documentation.sphinx.common.Document (*, projectname=None,
                                                                    authors=None,
                                                                    version=None,    re-
                                                                    lease=None)

Bases: annize.features.documentation.common.Document

    Parameters

    • projectname (Optional[str]) –

```

```

    • authors (Optional[str]) –
    • version (Optional[str]) –
    • release (Optional[str]) –
class GenerateInfo (intstruct, outdir, confdir, language, *, configvalues=None, configlines=None,
                      entry_path="")
    Bases: object

    Parameters

    • intstruct (annize.fs.FsEntry) –
    • outdir (annize.fs.FsEntry) –
    • confdir (annize.fs.FsEntry) –
    • language (str) –
    • configvalues (Optional[Dict]) –
    • configlines (Optional[List]) –
    • entry_path (str) –

    _to_dict()
    property confdir
    property configlines
    property configvalues
    property entry_path
    property intstruct
    property language
    property outdir

    _abc_impl = <_abc_data object>
    property _authors
    abstract _generate_master_doc (geninfo)

        Parameters geninfo      (annize.features.documentation.sphinx.common.
                                Document.GenerateInfo) –

        Return type str

    _prepare_generate (geninfo)

        Parameters geninfo      (annize.features.documentation.sphinx.common.
                                Document.GenerateInfo) –

        Return type None

    property _projectname
    property _release
    property _version
    generate (outputspec, *, language='?')
    generate_all_languages (outputspec)

```

```
class annize.features.documentation.sphinx.common.ReadmeDocument (*, project-
                                                                    name=None,
                                                                    au-
                                                                    thors=None,
                                                                    ver-
                                                                    sion=None,
                                                                    re-
                                                                    lease=None,
                                                                    documents,
                                                                    title=None,
                                                                    depen-
                                                                    dencies,
                                                                    project_version=None)
```

Bases: *annize.features.documentation.sphinx.common.CompositeDocument*

A reStructuredText formatted file or a directory of such files.

#### Parameters

- **projectname** (*Optional[str]*) –
- **authors** (*Optional[str]*) –
- **version** (*Optional[str]*) –
- **release** (*Optional[str]*) –
- **documents** (*Iterable[annize.features.documentation.sphinx.common.Document]*) –
- **title** (*Optional[str]*) –
- **dependencies** (*List[annize.features.dependencies.common.Dependency]*) –
- **project\_version** (*annize.data.version.Version*) –

**\_abc\_impl** = *<\_abc\_data object>*

**available\_languages** ()

Return type *List[str]*

property **title**

```
class annize.features.documentation.sphinx.common.RstDocument (*, variants,
                                                                **kwargs)
```

Bases: *annize.features.documentation.sphinx.common.Document*

A reStructuredText formatted file or a directory of such files.

**Parameters variants** (*List[annize.features.documentation.sphinx.common.RstDocumentVariant]*) –

**\_\_get\_variant** (*language*)

**Parameters language** (*str*) –

Return type *Optional[annize.features.documentation.sphinx.common.RstDocumentVariant]*

**\_abc\_impl** = *<\_abc\_data object>*

**\_generate\_master\_doc** (*geninfo*)

**available\_languages** ()

```
class annize.features.documentation.sphinx.common.RstDocumentVariant (*, lan-
                                                                    guage='?',
                                                                    source)
```

Bases: object

**Parameters**

- **language** (*str*) –
- **source** (*annize.fs.FsEntry*) –

**property language**

**property source**

**annize.features.documentation.sphinx.cpp module**

```
class annize.features.documentation.sphinx.cpp.CppApiReferenceLanguage (**kwargs)
    Bases: annize.features.documentation.sphinx.doxygencompat.
           DoxygenSupportedApiReferenceLanguage
```

C++ language support for api references.

```
_abc_impl = <_abc_data object>
```

## annize.features.documentation.sphinx.doxygencompat module

**class** annize.features.documentation.sphinx.doxygencompat.DoxxygenSupportedApiReferenceLanguage

Bases: *annize.features.documentation.sphinx.common.ApiReferenceLanguage*

Language support for api references powered by Doxygen.

### Parameters

- **\_doxygenopts** (*Optional[Dict[str, str]]*) –
- **extract\_all** (*bool*) –
- **extract\_private** (*bool*) –
- **extract\_package** (*bool*) –
- **extract\_static** (*bool*) –
- **extract\_local\_classes** (*bool*) –
- **extract\_local\_methods** (*bool*) –
- **extract\_anon\_nspaces** (*bool*) –
- **extract\_priv\_virtual** (*bool*) –
- **file\_patterns** (*str*) –
- **inline\_inherited\_memb** (*bool*) –
- **inherit\_docs** (*bool*) –

- **hide\_undoc\_members** (*bool*) –
- **hide\_undoc\_classes** (*bool*) –
- **exclude\_patterns** (*str*) –
- **predefined** (*Optional[List[str]]*) –

**\_\_info** (*outpath*)

**\_abc\_impl** = *<\_abc\_data object>*

**\_generate\_master\_doc** (*srcpath, intstructpath, confdirpath, heading*)

**prepare\_generate** (*rootpath, outpath*)

### annize.features.documentation.sphinx.javascript module

**class** `annize.features.documentation.sphinx.javascript.JavaScriptApiReferenceLanguage`  
 Bases: `annize.features.documentation.sphinx.common.ApiReferenceLanguage`

JavaScript language support for api references.

**\_\_jsfiles** (*dirf*)

Parameters **dirf** (*str*) –

Return type `List[str]`

**\_\_scanjsfile** (*f*)

Parameters **f** (*str*) –

Return type `str`

**\_abc\_impl** = *<\_abc\_data object>*

**generate\_master\_doc** (*rgeninfo*)

**prepare\_generate** (*rgeninfo*)

### annize.features.documentation.sphinx.python module

**class** `annize.features.documentation.sphinx.python.Python3ApiReferenceLanguage` (\*,

*show\_undoc\_members*  
*show\_protected\_members*  
*exclude\_patterns*  
*predefined*  
*include=None*)

Bases: `annize.features.documentation.sphinx.common.ApiReferenceLanguage`

Python 3 language support for api references.

Parameters

- **show\_undoc\_members** (*bool*) –
- **show\_protected\_members** (*bool*) –
- **exclude** (*Optional[List[str]]*) –

**\_\_patch\_property\_types\_in\_docstrings** (*pdir*)

Parameters **pdir** (*str*) –

Return type `None`

```
_abc_impl = <_abc_data object>
generate_master_doc (rgeninfo)
prepare_generate (rgeninfo)
```

## annize.features.documentation.sphinx.rst module

```
class annize.features.documentation.sphinx.rst.RstGenerator
    Bases: object
```

Generator for some documentation source parts.

```
static heading (text, *, variant='=', sub=False, anchor=None)
    Generates documentation source for a section heading.
```

### Parameters

- **text** (*str*) –
- **variant** (*str*) –
- **sub** (*bool*) –
- **anchor** (*Optional[str]*) –

**Return type** *str*

## Submodules

### annize.features.documentation.common module

```
class annize.features.documentation.common.Document
    Bases: abc.ABC
```

```
_abc_impl = <_abc_data object>
```

```
abstract available_languages ()
```

**Return type** *List[str]*

```
abstract generate (outputspec, *, language='?')
```

**Parameters** **language** (*str*) –

**Return type** *annize.features.documentation.common.\_DocumentGenerateResult*

```
abstract generate_all_languages (outputspec)
```

**Return type** *annize.features.documentation.common.\_DocumentGenerateAllLanguagesResult*

```
class annize.features.documentation.common.GeneratedDocument (*, document,
                                                                outputspec, language=None,
                                                                filename=None)
```

Bases: *annize.fs.AdHocFsEntry*

### Parameters

- **document** (*annize.features.documentation.common.Document*) –
- **outputspec** (*annize.features.documentation.common.OutputSpec*) –

```

        • language(str) –
        • filename(str) –
    _abc_impl = <_abc_data object>
    _initialize()

class annize.features.documentation.common.HtmlOutputSpec(*, is_homepage=False,
                                                           title=None)
    Bases: annize.features.documentation.common.OutputSpec
    HTML documentation output.

    Parameters
        • is_homepage(bool) – If to render output for a homepage with slight different stylings
          and behavior.
        • title(Optional[str]) – The html title. TODO higher level thing here?!

    _abc_impl = <_abc_data object>
    property is_homepage
    property title

class annize.features.documentation.common.OutputSpec
    Bases: abc.ABC

    Base class for documentation output specifications. See render().

    _abc_impl = <_abc_data object>

class annize.features.documentation.common.PdfOutputSpec
    Bases: annize.features.documentation.common.OutputSpec
    PDF documentation output.

    _abc_impl = <_abc_data object>

class annize.features.documentation.common.PlaintextOutputSpec
    Bases: annize.features.documentation.common.OutputSpec
    Plaintext documentation output.

    _abc_impl = <_abc_data object>

class annize.features.documentation.common._DocumentGenerateAllLanguagesResult(file,
                                                                                   en-
                                                                                   try_path,
                                                                                   en-
                                                                                   try_paths_for_la)
    Bases: annize.features.documentation.common._DocumentGenerateResult

    Parameters
        • file(annize.fs.FsEntry) –
        • entry_path(str) –
        • entry_paths_for_languages(Dict[str, str]) –
    entry_path_for_language(language)
    Parameters language(str) –
    Return type str

```



## property languages

```
class annize.features.documentation.common._DocumentGenerateResult (file, entry_path)
```

Bases: object

### Parameters

- **file** (annize.fs.FsEntry) –
- **entry\_path** (str) –

```
_set_entry_path (entry_path)
```

Parameters **entry\_path** (str) –

Return type None

property **entry\_path**

property **file**

## annize.features.files namespace

### Subpackages

### annize.features.files.transfer namespace

### Submodules

### annize.features.files.transfer.common module

```
class annize.features.files.transfer.common.Endpoint
```

Bases: object

```
access_filesystem (rootpath)
```

Parameters **rootpath** (str) –

Return type AbstractContextManager[annize.fs.FsEntry]

```
class annize.features.files.transfer.common.FsEndpoint (*, fsentry=None, path=None)
```

Bases: annize.features.files.transfer.common.Endpoint

### Parameters

- **fsentry** (annize.fs.FsEntry) –
- **path** (str) –

```
access_filesystem (rootpath)
```

Parameters **rootpath** (str) –

Return type AbstractContextManager[annize.fs.FsEntry]

```
class annize.features.files.transfer.common.Upload (*, source, destination_endpoint, destination_path)
```

Bases: object

### Parameters

- **source**(`annize.fs.FsEntry`) –
- **destination\_endpoint** (`annize.features.files.transfer.common.Endpoint`) –
- **destination\_path**(`str`) –

## annize.features.files.transfer.ssh module

**class** `annize.features.files.transfer.ssh.Endpoint` (\*, `host`, `port=22`, `username`, `identity_file=None`, `has_shell_access=False`)

Bases: `annize.features.files.transfer.common.Endpoint`

### Parameters

- **host**(`str`) –
- **port**(`int`) –
- **username**(`str`) –
- **identity\_file**(`Optional[str]`) –
- **has\_shell\_access**(`bool`) –

**access\_filesystem**(`rootpath`)

**exec**(`cmdline`)

Parameters **cmdline**(`str`) –

Return type *TODO*

property **has\_shell\_access**

property **host**

property **identity\_file**

locationstring(`path=None`)

Parameters **path**(`Optional[str]`) –

Return type `str`

property **port**

property **username**

## Submodules

### annize.features.files.common module

**class** `annize.features.files.common.Dir` (\*, `excludes`, `parts`, `root=None`, `path=None`, `name=None`)

Bases: `annize.features.files.common.FsEntry`

### Parameters

- **excludes**(`Iterable[annize.features.files.common.Exclude]`) –
- **parts**(`Iterable[annize.features.files.common.DirPart]`) –

```

    • root (annize.fs.FsEntry) –
    • path (str) –
    • name (str) –

classmethod __transfer_predicate_and (*predicates)
    Parameters predicates (Callable[[str, FsEntry, FsEntry], bool]) –
    Return type Callable[[str, annize.features.files.common.FsEntry, annize.features.files.common.FsEntry], bool]

classmethod __transfer_predicate_for_exclude (exclude)
    Parameters exclude (annize.features.files.common.Exclude) –
    Return type Callable[[str, annize.features.files.common.FsEntry, annize.features.files.common.FsEntry], bool]

classmethod __transfer_predicate_for_excludes (excludes)
    Parameters excludes (Iterable[annize.features.files.common.Exclude]) –
    Return type Callable[[str, annize.features.files.common.FsEntry, annize.features.files.common.FsEntry], bool]

_abc_impl = <_abc_data object>
_initialize()

class annize.features.files.common.DirPart (*, excludes=None, root=None,
source_path=None, destination_path=None,
path=None, destination_is_parent=False)

Bases: object

    Parameters
        • excludes (Iterable[annize.features.files.common.Exclude]) –
        • root (annize.fs.FsEntry) –
        • source_path (str) –
        • destination_path (str) –
        • path (str) –
        • destination_is_parent (bool) –

    property destination_is_parent
    property destination_path
    property excludes
    property path
    property root

class annize.features.files.common.Exclude (*, by_path_pattern=None, by_path=None,
by_name_pattern=None, by_name=None)

Bases: object

    Parameters
        • by_path_pattern (str) –

```

```

    • by_path(str) –
    • by_name_pattern(str) –
    • by_name(str) –
__does_exclude(by_text, by_pattern)

Parameters
    • text(str) –
    • by_text(str) –
    • by_pattern(re.Pattern) –

Return type bool

does_exclude(relative_path, source, destination)

Parameters
    • relative_path(str) –
    • source(annize.fs.FsEntry) –
    • destination(annize.fs.FsEntry) –

Return type bool

class annize.features.files.common.File(*, root=None, path=None)
    Bases: annize.features.files.common.FsEntry

    Parameters
        • root(annize.fs.FsEntry) –
        • path(str) –

    _abc_impl = <_abc_data object>

class annize.features.files.common.FsEntry(*, root=None, path=None)
    Bases: annize.fs.AdHocFsEntry

    Parameters
        • root(annize.fs.FsEntry) –
        • path(str) –

    _abc_impl = <_abc_data object>
    _initialize()
    property relative_path
    property root

class annize.features.files.common.MachineRootDirectory
    Bases: annize.fs.AdHocFsEntry

    _abc_impl = <_abc_data object>
    _initialize()

class annize.features.files.common.ProjectDir
    Bases: annize.fs.AdHocFsEntry

    _abc_impl = <_abc_data object>

```

```
_initialize()
```

**annize.features.homepage namespace**

**Subpackages**

**annize.features.homepage.sections namespace**

**Submodules**

**annize.features.homepage.sections.about module**

```
class annize.features.homepage.sections.about.Section(*,
                                                    head=<annize.i18n.ProvidedTrStr
                                                    object>, sortindex=10000)
    Bases: annize.features.homepage.common.HomepageSection
    _generate(info)
```

**annize.features.homepage.sections.changelog module**

```
class annize.features.homepage.sections.changelog.Section(*, changelog=None,
                                                         head=<annize.i18n.ProvidedTrStr
                                                         object>, sortin-
                                                         dex=60000)
    Bases: annize.features.homepage.common.HomepageSection
    Parameters changelog (annize.features.changelog.common.Changelog) –
    _generate(info)
```

**annize.features.homepage.sections.documentation module**

```
class annize.features.homepage.sections.documentation.Section(*, documentation,
                                                             head=<annize.i18n.ProvidedTrStr
                                                             object>, sortin-
                                                             dex=30000)
    Bases: annize.features.homepage.common.HomepageSection
    Parameters documentation (List[annize.features.documentation.common.
    Document]) –
    _generate(info)
    preproc_generate(info)
```

## annize.features.homepage.sections.download module

```
class annize.features.homepage.sections.download.Section(*,  
                                                         distributa-  
                                                         bles,      dependencies,  
                                                         head=<annize.i18n.ProvidedTrStr  
                                                         object>,    sortin-  
                                                         dex=40000)
```

Bases: *annize.features.homepage.common.HomepageSection*

### Parameters

- **distributables** (*List[annize.features.distributables.common.Group]*) –
- **dependencies** (*List[annize.features.dependencies.common.Dependency]*) –

**\_\_generate\_packagelist** (*info*)

**Parameters info** (*annize.features.homepage.common.HomepageSection.\_GenerateInfo*) –

**\_generate** (*info*)

**postproc\_generate** (*info*)

**preproc\_generate** (*info*)

annize.features.homepage.sections.download.**filehash** (*filepath*)

**Parameters filepath** (*str*) –

**Return type** *str*

annize.features.homepage.sections.download.**friendly\_filesize** (*isize*)

**Parameters isize** (*int*) –

**Return type** *str*

## annize.features.homepage.sections.gallery module

```
class annize.features.homepage.sections.gallery.Section(*,  
                                                         head=<annize.i18n.ProvidedTrStr  
                                                         object>, sortindex=50000,  
                                                         mediagalleries)
```

Bases: *annize.features.homepage.common.HomepageSection*

**Parameters mediagalleries** (*List[annize.features.mediagalleries.Gallery]*) –

**\_generate** (*info*)

**preproc\_generate** (*info*)

## annize.features.homepage.sections.imprint module

```
class annize.features.homepage.sections.imprint.Section(*, imprint,
                                                         head=<annize.i18n.ProvidedTrStr
                                                         object>,          sortin-
                                                         dex=70000)

Bases: annize.features.homepage.common.HomepageSection

Parameters imprint (str)–

_generate (info)
```

## annize.features.homepage.sections.license module

```
class annize.features.homepage.sections.license.Section(*,
                                                         head=<annize.i18n.ProvidedTrStr
                                                         object>,          sortin-
                                                         dex=20000)

Bases: annize.features.homepage.common.HomepageSection

_generate (info)
```

## Submodules

### annize.features.homepage.common module

```
class annize.features.homepage.common.GeneratedHomepage(*, homepage)
    Bases: annize.fs.AdHocFsEntry

    Parameters homepage (annize.features.homepage.common.Homepage)–

    _abc_impl = <_abc_data object>

    _initialize()

class annize.features.homepage.common.Homepage(*, title=None, short_desc=None, sec-
                                                         tions, languages='')

Bases: object

Parameters

    • title (Optional[str])–

    • short_desc (Optional[str])–

    • sections (List[annize.features.homepage.common.
                  HomepageSection])–

    • languages (str)–

__generate_postpreproc(*, is_pre, document_root_directory, document_root_url)

Parameters

    • customargs (Dict)–

    • is_pre (bool)–

    • document_root_directory (annize.fs.FsEntry)–

    • document_root_url (str)–
```

**\_\_generate\_section** (\*, *customargs*, *culture*, *document\_root\_directory*, *document\_root\_url*, *document\_variant\_directory*)

Parameters

- **customargs** (*Dict*) –
- **culture** (*annize.i18n.Culture*) –
- **document\_root\_directory** (*annize.fs.FsEntry*) –
- **document\_root\_url** (*str*) –
- **document\_variant\_directory** (*annize.fs.FsEntry*) –

**\_\_append\_section** (*section*)

Parameters **section** (*annize.features.homepage.common.HomepageSection*) –

Return type *None*

**generate** ()

property **languages**

property **sections**

property **short\_desc**

property **title**

**class** *annize.features.homepage.common.HomepageSection* (\*, *head*, *sortindex=0*)

Bases: *object*

Parameters

- **head** (*str*) –
- **sortindex** (*int*) –

**class** *Product* (\*, *rst\_text=""*, *media\_files=()*)

Bases: *object*

Parameters

- **rst\_text** (*str*) –
- **media\_files** (*List[annize.fs.FsEntry]*) –

**append\_rst** (*rst\_text*)

Return type *None*

**attach\_media\_file** (*file*)

Parameters **file** (*annize.fs.FsEntry*) –

Return type *None*

property **media\_files**

property **rst\_text**

**class** *\_GenerateInfo* (*culture: annize.i18n.Culture*, *customarg: Union[object, NoneType]*, *document\_root\_url: str*, *document\_variant\_directory: annize.fs.FsEntry*, *document\_variant\_url: str*)

Bases: *object*

*\_GenerateInfo*(*culture: annize.i18n.Culture*, *customarg: Union[object, NoneType]*, *document\_root\_url: str*, *document\_variant\_directory: annize.fs.FsEntry*, *document\_variant\_url: str*)



#### Parameters

- **culture** (*annize.i18n.Culture*) –
- **customarg** (*Optional[object]*) –
- **document\_root\_url** (*str*) –
- **document\_variant\_directory** (*annize.fs.FsEntry*) –
- **document\_variant\_url** (*str*) –

Return type *None*

**culture:** *annize.i18n.Culture*

**customarg:** *Optional[object]*

**document\_root\_url:** *str*

**document\_variant\_directory:** *annize.fs.FsEntry*

**document\_variant\_url:** *str*

**class** *\_PrePostprocGenerateInfo* (*document\_root\_directory: annize.fs.FsEntry, document\_root\_url: str, customarg: Union[object, NoneType]*)

Bases: *object*

*\_PrePostprocGenerateInfo*(*document\_root\_directory: annize.fs.FsEntry, document\_root\_url: str, customarg: Union[object, NoneType]*)

#### Parameters

- **document\_root\_directory** (*annize.fs.FsEntry*) –
- **document\_root\_url** (*str*) –
- **customarg** (*Optional[object]*) –

Return type *None*

**customarg:** *Optional[object]*

**document\_root\_directory:** *annize.fs.FsEntry*

**document\_root\_url:** *str*

**\_generate** (*info*)

**Parameters** **info** (*annize.features.homepage.common.HomepageSection.\_GenerateInfo*) –

**Return type** *Union[annize.features.homepage.common.HomepageSection.Product, str]*

**generate** (*info*)

**Parameters** **info** (*annize.features.homepage.common.HomepageSection.\_GenerateInfo*) –

**Return type** *Optional[annize.features.homepage.common.HomepageSection.Product]*

**property** **head**

**postproc\_generate** (*info*)

**Parameters** **info** (*annize.features.homepage.common.HomepageSection.\_PrePostprocGenerateInfo*) –

Return type *None*

```

preproc_generate (info)
    Parameters info    (annize.features.homepage.common.HomepageSection.
                        _PrePostprocGenerateInfo) –
    Return type None

property sortindex

class annize.features.homepage.common.PinoHomepageAuxInfo
    Bases: annize.features.homepage.common.HomepageSection
    _generate (info)
    preproc_generate (info)

class annize.features.homepage.common.SimpleProjectHomepage (*, changelog=None,
                                                             dependencies,
                                                             distributables,
                                                             documentation,
                                                             imprint=None,
                                                             mediagalleries,
                                                             **kwargs)

    Bases: annize.features.homepage.common.Homepage

    Parameters
        • changelog (annize.features.changelog.common.Changelog) –
        • dependencies      (List [annize.features.dependencies.common.
                             Dependency]) –
        • distributables    (List [annize.features.distributables.common.
                             Group]) –
        • documentation    (List [annize.features.documentation.common.
                             Document]) –
        • imprint (str) –
        • mediagalleries (List [annize.features.mediagalleries.Gallery]) –

```

## annize.features.i18n namespace

### Submodules

#### annize.features.i18n.common module

```

class annize.features.i18n.common.ProjectDefinedTranslationProvider
    Bases: annize.i18n.TranslationProvider
    __translations_for_stringname (stringname)
        Parameters stringname (str) –
        Return type Dict[str, str]
    add_translations (stringname, variants)
        Parameters
            • stringname (str) –

```

- **variants** (*Dict[str, str]*) –

Return type None

**translate** (*stringname*, \*, *culture*)

**class** `annize.features.i18n.common.String` (\*, *stringname=None*, *stringtr=None*, \*\**variants*)

Bases: `annize.i18n.ProvidedTrStr`

Parameters

- **stringname** (*Optional[str]*) –
- **stringtr** (*Optional[str]*) –
- **variants** (*str*) –

## annize.features.i18n.gettext module

**class** `annize.features.i18n.gettext.GenerateMOs` (\*, *po\_directory*, *mo\_directory*)

Bases: `object`

Parameters

- **po\_directory** (`annize.fs.FsEntry`) –
- **mo\_directory** (`annize.fs.FsEntry`) –

**class** `annize.features.i18n.gettext.UpdatePOs` (\*, *po\_directory*)

Bases: `object`

Parameters **po\_directory** (`annize.fs.FsEntry`) –

## annize.features.testing namespace

### Submodules

### annize.features.testing.common module

**class** `annize.features.testing.common.RunTests` (\*\**b*)

Bases: `object`

**class** `annize.features.testing.common.Test`

Bases: `object`

**run** ()

**class** `annize.features.testing.common.TestGroup` (\*, *tests*)

Bases: `annize.features.testing.common.Test`

Parameters **tests** (*List[annize.features.testing.common.Test]*) –

**run** ()

### annize.features.testing.pylint module

```
class annize.features.testing.pylint.Test (*, src)
    Bases: annize.features.testing.common.Test

    Parameters src (annize.fs.FsEntry) –

    run ()
```

### annize.features.testing.pytest module

```
class annize.features.testing.pytest.Test (*, testdir, src)
    Bases: annize.features.testing.common.Test

    Parameters

    • testdir (str) –

    • src (str) –

    run ()
```

### annize.features.testing.pyunit module

```
class annize.features.testing.pyunit.Test (*, src)
    Bases: annize.features.testing.common.Test

    Parameters src (str) –

    run ()
```

### annize.features.versioncontrol namespace

#### Submodules

#### annize.features.versioncontrol.common module

```
class annize.features.versioncontrol.common.BuildVersion (*, base_version, vcs)
    Bases: annize.data.version.Version

    Parameters

    • base_version (annize.data.version.Version) –

    • vcs (annize.features.versioncontrol.common.
      VersionControlSystem) –

    __effversion ()

    property segments_tuples

    property text

class annize.features.versioncontrol.common.VersionControlSystem
    Bases: object

    get_commit_message (revision)
```

```

    Parameters revision (str) –
    Return type str

get_commit_time (revision)
    Parameters revision (str) –
    Return type datetime.datetime

get_current_revision ()
    Return type str

get_revision_list ()
    Return type List[str]

get_revision_number (revision)
    Parameters revision (str) –
    Return type int

annize.features.versioncontrol.common.default_version_control_system ()
    Return type Optional[annize.features.versioncontrol.common.VersionControlSystem]

```

## annize.features.versioncontrol.git module

```

class annize.features.versioncontrol.git.ExcludeByGitIgnores
    Bases: annize.features.files.common.Exclude
    does_exclude (relative_path, source, destination)

class annize.features.versioncontrol.git.VersionControlSystem (*, path="")
    Bases: annize.features.versioncontrol.common.VersionControlSystem
    Parameters path (str) –
    __call_git (cmdline)
        Parameters cmdline (List[str]) –
        Return type str

    get_commit_message (revision)
    get_commit_time (revision)
    get_current_revision ()
    get_revision_list ()
    get_revision_number (revision)
    property path

```

## Submodules

### annize.features.authors module

**class** annize.features.authors.**Author** (\*, *fullname*, *email*=None)

Bases: object

#### Parameters

- **fullname** (*str*) –
- **email** (*str*) –

**property** email

**property** fullname

annize.features.authors.**default\_project\_authors** ()

**Return type** List[annize.features.authors.Author]

### annize.features.base module

**class** annize.features.base.**Basket** (\*, *objects*)

Bases: annize.data.container.Basket

**Parameters** **objects** (List[object]) –

**class** annize.features.base.**BrandColor** (\*, *red*, *green*, *blue*)

Bases: annize.data.color.Color

#### Parameters

- **red** (*float*) –
- **green** (*float*) –
- **blue** (*float*) –

**class** annize.features.base.**Data** (\*, *project\_name*=None, *pretty\_project\_name*=None, *summary*=None, *long\_description*=None, *homepage\_url*=None, *imprint*=None, *project\_directory*=None)

Bases: object

#### Parameters

- **project\_name** (Union[*str*, annize.i18n.TrStr]) –
- **pretty\_project\_name** (Union[*str*, annize.i18n.TrStr]) –
- **summary** (Union[*str*, annize.i18n.TrStr]) –
- **long\_description** (Union[*str*, annize.i18n.TrStr]) –
- **homepage\_url** (Union[*str*, annize.i18n.TrStr]) –
- **imprint** (Union[*str*, annize.i18n.TrStr]) –
- **project\_directory** (*str*) –

**property** homepage\_url

**property** imprint

**property** long\_description

```

    property pretty_project_name
    property project_directory
    property project_name
    property summary
class annize.features.base.DateTime(*, iso=None)
    Bases: datetime.datetime

    Parameters iso(str) –
class annize.features.base.FirstOf(*, objects)
    Bases: annize.data.container.Basket

    Parameters objects(List[object]) –
class annize.features.base.Keyword(text)
    Bases: annize.features.base.Keywords

    Parameters text(str) –
class annize.features.base.Keywords(*, from_string="", split_by=',', keywords=())
    Bases: object

    Parameters
        • from_string(str) –
        • split_by(str) –
        • keywords(List[str]) –

    property keywords
annize.features.base._get_data(key, defaultvalue)

    Parameters
        • key(str) –
        • defaultvalue(Optional[Any]) –

    Return type Optional[Any]
annize.features.base.brand_color(*, none_on_undefined=False)

    Parameters none_on_undefined(bool) –

    Return type annize.data.color.Color
annize.features.base.homepage_url()

    Return type str
annize.features.base.imprint()

    Return type str
annize.features.base.long_description()

    Return type str
annize.features.base.pretty_project_name()

    Return type str
annize.features.base.project_directory()

```

**Return type** str

`annize.features.base.project_keywords()`

**Return type** *annize.features.base.Keywords*

`annize.features.base.project_name()`

**Return type** str

`annize.features.base.summary()`

**Return type** str

## annize.features.licensing module

`annize.features.licensing.AFLv3`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.AGPLv3`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.Apache2`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.Artistic1`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.BSD2clause`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.BSD3clause`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.Cc0v1`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.CcBy3`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.CcByNc3`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.CcByNcNd3`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.CcByNcSa3`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.CcByNd3`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.CcBySa3`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.GPLv3`

alias of `annize.features.licensing._license.<locals>.ALicense`

`annize.features.licensing.LGPLv3`

alias of `annize.features.licensing._license.<locals>.ALicense`

**class** `annize.features.licensing.License(*, name, text=None, **additional_info)`

Bases: object



#### Parameters

- **name** (*str*) –
- **text** (*str*) –

**additional\_info** (*key*, \*, *default=None*)

#### Parameters

- **key** (*str*) –
- **default** (*Optional[Any]*) –

**Return type** Any

**property name**

**property text**

```
annize.features.licensing.MIT
    alias of annize.features.licensing._license.<locals>.ALicense
annize.features.licensing.MPLv11
    alias of annize.features.licensing._license.<locals>.ALicense
annize.features.licensing.MPLv2
    alias of annize.features.licensing._license.<locals>.ALicense
annize.features.licensing.PublicDomain
    alias of annize.features.licensing._license.<locals>.ALicense
annize.features.licensing._license (name)
```

**Parameters** **name** (*str*) –

**Return type** Type[*annize.features.licensing.License*]

```
annize.features.licensing.project_licenses()
```

**Return type** List[*annize.features.licensing.License*]

### annize.features.medialogalleries module

```
class annize.features.medialogalleries.Gallery (*, source, title="")
    Bases: object
```

#### Parameters

- **source** (*annize.fs.FsEntry*) –
- **title** (*str*) –

```
class Item (file, description)
```

Bases: object

**property description**

**property file**

```
_description_for_mediafile (itemfile)
```

**Parameters** **itemfile** (*annize.fs.FsEntry*) –

**Return type** str

**property items**

**property title**

## annize.features.task module

```
class annize.features.task.Task(*, innertasks)
    Bases: object

    Parameters innertasks (List[object]) –
```

## annize.features.version module

```
class annize.features.version.CommonVersionPattern
    Bases: annize.data.version.VersionPattern

class annize.features.version.Line(*, version)
    Bases: object

    Parameters version (annize.data.version.Version) –

    property version

class annize.features.version.Version(*, text=None, pattern=None, **segment_values)
    Bases: annize.data.version.Version

    Parameters

    • text (str) –

    • pattern (annize.data.version.VersionPattern) –

annize.features.version.default_version_pattern()

    Return type annize.data.version.VersionPattern

annize.features.version.project_versions()

    Return type List[annize.data.version.Version]
```

## annize.fs package

```
class annize.fs.AdHocFsEntry
    Bases: annize.fs.FsEntry

    __getinner()

    _abc_impl = <_abc_data object>

    _initialize()

    Return type annize.fs.FsEntry

    readable_path()

exception annize.fs.BadDisowningError(path)
    Bases: RuntimeError

    Parameters path (str) –

class annize.fs.ChDir(path)
    Bases: object

    Temporarily changes the current working directory in a scope of the with keyword.
```

**Parameters** `path` (*str*) – The path to the new working directory.

**class** `annize.fs.FsEntry` (\*, *is\_ownable=False*, *is\_owned=False*)

Bases: `abc.ABC`

**Parameters**

- `is_ownable` (*bool*) –

- `is_owned` (*bool*) –

`_abc_impl` = `<_abc_data object>`

`basename` ()

**Return type** `str`

`child_by_relative_path` (*name*)

**Return type** `annize.fs.FsEntry`

`children` ()

**Return type** `List[annize.fs.FsEntry]`

`copy_to` (*path*, \*, *path\_is\_parent=False*, *merge=False*, *overwrite=False*, *transfer\_predicate=None*)

**Parameters**

- `path` (*str*) –

- `path_is_parent` (*bool*) –

- `merge` (*bool*) –

- `overwrite` (*bool*) –

- `transfer_predicate` (*Optional[Callable[[str, FsEntry, FsEntry], bool]]*) –

**Return type** `annize.fs.FsEntry`

`created_time` ()

**Return type** `datetime.datetime`

`disowned` ()

**Return type** `annize.fs.FsEntry`

`exists` ()

**Return type** `bool`

`filesize` ()

**Return type** `int`

`is_dir` (*follow\_symlinks=True*)

**Parameters** `follow_symlinks` (*bool*) –

**Return type** `bool`

`is_file` (*follow\_symlinks=True*)

**Parameters** `follow_symlinks` (*bool*) –

**Return type** `bool`

`is_link` ()

**Return type** bool

**property is\_ownable**

**property is\_owned**

**mkdir** (\*, *exist\_ok=True*)

**Parameters** **exist\_ok** (*bool*) –

**Return type** None

**modified\_time** ()

**Return type** datetime.datetime

**move\_to** (path, \*, *path\_is\_parent=False, merge=False, overwrite=False, transfer\_predicate=None*)

**Parameters**

- **path** (*str*) –
- **path\_is\_parent** (*bool*) –
- **merge** (*bool*) –
- **overwrite** (*bool*) –
- **transfer\_predicate** (*Optional[Callable[[str, FsEntry, FsEntry], bool]]*) – (Optional[Callable[[str, FsEntry, FsEntry], bool]]) –

**Return type** *annize.fs.FsEntry*

**owned** (*allow\_clone=True*)

**Parameters** **allow\_clone** (*bool*) –

**Return type** *annize.fs.FsEntry*

**parent** ()

**Return type** Optional[*annize.fs.FsEntry*]

**path** ()

**Return type** str

**read\_file** ()

**Return type** bytes

**readable\_path** ()

**Return type** str

**remove** ()

**Return type** None

**temp\_clone** (\*, *temprootpath=None, basename=None*)

**Parameters**

- **temprootpath** (*Optional[str]*) –
- **basename** (*Optional[str]*) –

**Return type** *annize.fs.FsEntry*

**write\_file** (*content*)

```

    Parameters content (AnyStr) –
    Return type None

exception annize.fs.MissingOwnershipError (path)
    Bases: RuntimeError

    Parameters path (str) –

class annize.fs.Mount (src, dst, *, options=(), mounter=None, unmounter=None)
    Bases: object

    Parameters

    • src (str) –
    • dst (str) –
    • options (Iterable[str]) –
    • mounter (Optional[Iterable[str]]) –
    • unmounter (Optional[Iterable[str]]) –

    property destination

exception annize.fs.OriginalOwnershipImpossibleError (path)
    Bases: RuntimeError

    Parameters path (str) –

class annize.fs._SimpleFsEntry (path, *, is_ownable=False, is_owned=False)
    Bases: annize.fs.FsEntry

    Parameters

    • path (str) –
    • is_ownable (bool) –
    • is_owned (bool) –

    _abc_impl = <_abc_data object>

    readable_path()

class annize.fs._TransferHelper
    Bases: object

    _TransferHelper__transfer_dirmerge (action, transfer_predicate)

    Parameters

    • action (Callable) –
    • transfer_predicate (Optional[Callable[[str, FsEntry, FsEntry], bool]]) –

    Return type Callable

    _TransferHelper__transfer_prepare (srcf, dstf, overwrite)

    Parameters

    • srcf (annize.fs.FsEntry) –
    • dstf (annize.fs.FsEntry) –
    • overwrite (bool) –

```

Return type `None`

**transfer\_action\_copy** (*srcf, dstf, overwrite*)

Parameters

- **srcf** (*annize.fs.FsEntry*) –
- **dstf** (*annize.fs.FsEntry*) –
- **overwrite** (*bool*) –

Return type `None`

**transfer\_action\_move** (*srcf, dstf, overwrite*)

Parameters

- **srcf** (*annize.fs.FsEntry*) –
- **dstf** (*annize.fs.FsEntry*) –
- **overwrite** (*bool*) –

Return type `None`

**transfer\_to** (*srcf, path, \*, merge, overwrite, action, transfer\_predicate=None*)

Parameters

- **srcf** (*annize.fs.FsEntry*) –
- **path** (*str*) –
- **merge** (*bool*) –
- **overwrite** (*bool*) –
- **action** (*Callable*) –
- **transfer\_predicate** (*Optional[Callable[[str, FsEntry, FsEntry], bool]]*) –

Return type *annize.fs.FsEntry*

**annize.fs.fsentry\_by\_path** (*path, \*, is\_ownable=False, is\_owned=False*)

Parameters

- **path** (*str*) –
- **is\_ownable** (*bool*) –
- **is\_owned** (*bool*) –

Return type *annize.fs.FsEntry*

## Submodules

### annize.fs.temp module

**class** annize.fs.temp.**TempFileByContent** (\*, content, filename='afile')

Bases: *annize.fs.AdHocFsEntry*

#### Parameters

- **content** (*AnyStr*) –
- **filename** (*Optional[str]*) –

**\_abc\_impl** = <\_abc\_data object>

**\_initialize**()

Return type *annize.fs.FsEntry*

**class** annize.fs.temp.**\_TempDir** (dirname=None, temprootpath=None)

Bases: *annize.fs.FsEntry*

#### Parameters

- **dirname** (*Optional[str]*) –
- **temprootpath** (*Optional[str]*) –

**\_TempDir\_\_cleanup**()

**\_abc\_impl** = <\_abc\_data object>

**readable\_path**()

**restrict\_path**(path)

annize.fs.temp.**fresh\_temp\_dir** (dirname=None, \*, temprootpath=None)

#### Parameters

- **dirname** (*Optional[str]*) –
- **temprootpath** (*Optional[str]*) –

Return type *annize.fs.temp.\_TempDir*

### annize.i18n package

**class** annize.i18n.**Culture** (english\_lang\_name, iso\_639\_1\_lang\_code)

Bases: *object*

#### Parameters

- **english\_lang\_name** (*str*) –
- **iso\_639\_1\_lang\_code** (*str*) –

**property** english\_lang\_name

**static** get\_from\_iso\_639\_1\_lang\_code (iso\_639\_1\_lang\_code)

**property** iso\_639\_1\_lang\_code

**class** annize.i18n.**GettextTranslationProvider** (mopath)

Bases: *annize.i18n.TranslationProvider*

```

translate (stringname, *, culture)

    Parameters
        • stringname (str) –
        • culture (annize.i18n.Culture) –

class annize.i18n.IndependentTrStr (**variants)
    Bases: annize.i18n.TrStr

    Parameters variants (str) –

    translate (culture=None)

    Return type str

exception annize.i18n.NoCurrentCultureError
    Bases: TypeError

class annize.i18n.ProvidedTrStr (stringname)
    Bases: annize.i18n.TrStr

    Parameters stringname (str) –

    translate (culture=None)

    Return type str

class annize.i18n.TODO (txt)
    Bases: annize.i18n.TrStr

    translate (culture=None)

    Parameters culture (Optional[Union[Culture, str]]) –

    Return type str

class annize.i18n.TrStr
    Bases: object

    format (*args, **kwargs)

    Return type annize.i18n.TrStr

    static tr (stringname)

    Parameters stringname (str) –

    Return type annize.i18n.TrStr

    translate (culture=None)

    Parameters culture (Optional[Union[Culture, str]]) –

    Return type str

annize.i18n.TrStrOrStr
    Type annotation for something that can be either a str or a TrStr.
    alias of Union[str, annize.i18n.TrStr]

class annize.i18n.TranslationProvider
    Bases: object

    translate (stringname, *, culture)

    Parameters

```



- **stringname** (*str*) –
- **culture** (*annize.i18n.Culture*) –

**Return type** *str*

**exception** *annize.i18n.TranslationUnavailableError* (*stringname, language*)

Bases: *TypeError*

**Parameters**

- **stringname** (*str*) –
- **language** (*str*) –

*annize.i18n.add\_translation\_provider* (*provider, \*, priority=0*)

**Parameters**

- **provider** (*annize.i18n.TranslationProvider*) –
- **priority** (*int*) –

**Return type** *None*

*annize.i18n.annize\_user\_interaction\_culture* ()

**Return type** *annize.i18n.Culture*

*annize.i18n.current\_culture* ()

**Return type** *annize.i18n.Culture*

*annize.i18n.get\_culture* (*culture*)

**Parameters** **culture** (*Optional[Union[Culture, str]]*) –

**Return type** *Optional[annize.i18n.Culture]*

*annize.i18n.tr* (*stringname, \*, culture=None*)

**Parameters**

- **stringname** (*str*) –
- **culture** (*Optional[Union[Culture, str]]*) –

**Return type** *str*

*annize.i18n.tr\_if\_trstr* (*txt, culture=None*)

**Parameters**

- **txt** (*Union[str, annize.i18n.TrStr]*) –
- **culture** (*Optional[Union[Culture, str]]*) –

**Return type** *str*

*annize.i18n.translation\_providers* ()

**Return type** *List[annize.i18n.TranslationProvider]*

## annize.project package

Definitions of Annize projects.

```

class annize.project.ArgumentNode
    Bases: annize.project.Node

    _str_helper()

    property append_to

    property arg_name

    property name

exception annize.project.BadStructureError(message)
    Bases: ValueError

    Parameters message (str) –

class annize.project.BlockNode
    Bases: annize.project.Node

    classmethod _allowed_child_types()

class annize.project.CallableNode
    Bases: annize.project.ArgumentNode

    classmethod _allowed_child_types()

    _str_helper()

    property callee

    property feature

exception annize.project.FeatureUnavailableError(featurename)
    Bases: ModuleNotFoundError

    Parameters featurename (str) –

class annize.project.FileNode(path)
    Bases: annize.project.Node

    Parameters path (str) –

    classmethod _allowed_child_types()

    _str_helper()

    property path

exception annize.project.MaterializerError(message)
    Bases: TypeError

    Parameters message (str) –

class annize.project.Node
    Bases: object

    __get_description(indent, with_children, multiline)

    Parameters

        • indent (int) –

        • with_children (bool) –
    
```

```

        • multiline (bool) –
            Return type str
classmethod _allowed_child_types ()
            Return type Iterable[Type[annize.project.Node]]
_str_helper ()
append_child (node)
            Parameters node (annize.project.Node) –
property children
clone (with_children=True)
            Parameters with_children (bool) –
            Return type annize.project.Node
get_description (*, with_children=True, multiline=True)
            Parameters
                • with_children (bool) –
                • multiline (bool) –
            Return type str
insert_child (i, node)
            Parameters
                • i (int) –
                • node (annize.project.Node) –
property parent
remove_child (node)
            Parameters node (annize.project.Node) –
class annize.project.OnFeatureUnavailableNode
    Bases: annize.project.ScopableBlockNode
class Action (value)
    Bases: enum.Enum
    An enumeration.
    FAIL = 'fail'
    SKIP_BLOCK = 'skip_block'
    SKIP_NODE = 'skip_node'
_str_helper ()
property do
property feature
exception annize.project.ParserError (message)
    Bases: ValueError
    Parsing error like bad input xml.

```

**Parameters** `message (str)` –

**class** `annize.project.ProjectDefinition (*nodes)`  
 Bases: `object`

A data structure that holds an in-memory representation of the project description. It is not the in-memory representation of the live universe object state. It is like the file structure of the project description file, but not exactly the same. There is a parser and a generator for translating between a project description file and an `ProjectDefinition`.

**Parameters** `nodes (FileNode)` –

**static from\_files** (`path`)

**Parameters** `path (str)` –

**Return type** `annize.project.ProjectDefinition`

**property nodes**

**class** `annize.project.ReferenceNode`  
 Bases: `annize.project.ArgumentNode`

**class** `OnUnresolvableAction (value)`  
 Bases: `enum.Enum`

An enumeration.

**FAIL** = 'fail'

**SKIP** = 'skip'

**classmethod** `_allowed_child_types ()`

**\_str\_helper** ()

**property on\_unresolvable**

**property reference\_key**

**class** `annize.project.ScalarValueNode`  
 Bases: `annize.project.ArgumentNode`

**\_\_shorten** (`maxlen=100`)

**Parameters**

- `obj (Any)` –
- `maxlen (int)` –

**Return type** `str`

**\_str\_helper** ()

**property value**

**class** `annize.project.ScopableBlockNode`  
 Bases: `annize.project.BlockNode`

**class** `Scope (value)`  
 Bases: `enum.Enum`

An enumeration.

**BLOCK** = 'block'

**FILE** = 'file'

```

    PROJECT = 'project'
    _str_helper()
    property scope
exception annize.project.UnresolvableReferenceError(refname)
    Bases: annize.project.MaterializerError
    Parameters refname(str) -

```

## Subpackages

### annize.project.materializer package

Materializing of Annize projects into a working structure.

```
annize.project.materializer.materialize(nodes, *, featureloader=None)
```

#### Parameters

- **nodes** (*Iterable*[*annize.project.FileNode*]) -
- **featureloader** (*Optional*[*annize.project.featureloader.FeatureLoader*]) -

**Return type** List[Optional[Any]]

## Subpackages

### annize.project.materializer.behaviors package

```

class annize.project.materializer.behaviors.Behavior
    Bases: abc.ABC
    _abc_impl = <_abc_data object>
    abstract node_context(nodemat)
        Parameters nodemat (annize.project.materializer.core.
            NodeMaterialization) -
        Return type AbstractContextManager
    stop_materialization_here_and_try_again()
        Return type None

```

## Submodules

### annize.project.materializer.behaviors.argument module

```

class annize.project.materializer.behaviors.argument.ArgumentBehavior(callfct,
                                                                    *,
                                                                    fea-
                                                                    tureloader)
    Bases: annize.project.materializer.behaviors.Behavior

```

```

    Parameters featureloader (annize.project.featureloader.FeatureLoader) –
    _abc_impl = <_abc_data object>
    node_context (nodemat)

```

#### annize.project.materializer.behaviors.basket module

```

class annize.project.materializer.behaviors.basket.BasketBehavior
    Bases: annize.project.materializer.behaviors.Behavior
    _abc_impl = <_abc_data object>
    node_context (nodemat)

```

#### annize.project.materializer.behaviors.block module

```

class annize.project.materializer.behaviors.block.BlockBehavior
    Bases: annize.project.materializer.behaviors.Behavior
    _abc_impl = <_abc_data object>
    node_context (nodemat)

```

#### annize.project.materializer.behaviors.featureunavailable module

```

class annize.project.materializer.behaviors.featureunavailable.FeatureUnavailableBehavior
    Bases: annize.project.materializer.behaviors.Behavior
    __context_catchexceptions (nodemat, featureignorelist)
    __context_skipnode_featureignorelist (node)
    _abc_impl = <_abc_data object>
    node_context (nodemat)

```

#### annize.project.materializer.behaviors.reference module

```

class annize.project.materializer.behaviors.reference.ReferenceBehavior (rootblock)
    Bases: annize.project.materializer.behaviors.Behavior
    _abc_impl = <_abc_data object>
    detect_retry_useful_node_context (nodemat)
    node_context (nodemat)
    resolver_node_context (nodemat)
    retry_node_context (nodemat)

class annize.project.materializer.behaviors.reference.SkipUnresolvableReferenceBehavior (rootblock)
    Bases: annize.project.materializer.behaviors.Behavior
    classmethod __node_is_hopeless (node, rootnodemat)
        Parameters node (annize.project.Node) –

```

Return type `bool`

`__nodes_by_name` (\*, *rootnode*)

Parameters

- `name` (*str*) –
- `rootnode` (*annize.project.Node*) –

Return type `List[annize.project.Node]`

`classmethod` `__reference_is_hopeless` (*refnode*, \*, *rootnodemat*)

Parameters `refnode` (*annize.project.ReferenceNode*) –

Return type `bool`

`__unresolvable_references` ()

Return type `List[annize.project.ReferenceNode]`

`_abc_impl` = `<_abc_data object>`

`node_context` (*nodemat*)

## Submodules

### annize.project.materializer.core module

`class` `annize.project.materializer.core.NodeMaterialization` (*materializer*, *node*, *store*)

Bases: `object`

Parameters

- `materializer` (*ProjectMaterializer*) –
- `node` (*annize.project.Node*) –
- `store` (*Dict*) –

`get_materialized_children` ()

Return type `Iterable[Optional[Any]]`

`get_materialized_children_tuples` ()

property `has_result`

property `node`

property `result`

`set_materialized_result` (*resultlist*)

`try_get_materialization_for_node` (*node*)

Parameters `node` (*annize.project.Node*) –

`class` `annize.project.materializer.core.ProjectMaterializer` (*node*, \*, *behaviors*)

Bases: `object`

Parameters

- `node` (*annize.project.Node*) –

- **behaviors** (*Iterable[annize.project.materializer.behaviors.Behavior]*) –

**\_\_materialization\_for\_node** (*node, store*)

**Parameters**

- **node** (*annize.project.Node*) –
- **store** (*Dict*) –

**Return type** *annize.project.materializer.core.NodeMaterialization*

**\_\_materialize** (*node, store*)

**Parameters**

- **node** (*annize.project.Node*) –
- **store** (*Dict*) –

**Return type** *List[Optional[Any]]*

**\_\_materialize\_hlp\_childobjs** (*node, store*)

**Parameters**

- **node** (*annize.project.Node*) –
- **store** (*Dict*) –

**Return type** *List[Tuple[annize.project.Node, List[Optional[Any]]]]*

**get\_materialized** ()

**Return type** *List[Optional[Any]]*

**exception** *annize.project.materializer.core.TryAgain*  
 Bases: *BaseException*

## annize.project.materializer.preprocessors module

*annize.project.materializer.preprocessors.normalize\_blockscopes* (*nodes*)

**Parameters** **nodes** (*Iterable[annize.project.Node]*) –

**Return type** *Iterable[annize.project.Node]*

*annize.project.materializer.preprocessors.resolve\_appendtonodes* (*nodes*)

**Parameters** **nodes** (*Iterable[annize.project.Node]*) –

**Return type** *Iterable[annize.project.Node]*



## Submodules

### annize.project.featureloader module

Feature module loader.

```
class annize.project.featureloader.DefaultFeatureLoader
    Bases: annize.project.featureloader.FeatureLoader
    _FEATURES_NAMESPACE = 'annize.features'
    _abc_impl = <_abc_data object>
    load_feature(name)

class annize.project.featureloader.FeatureLoader
    Bases: abc.ABC
    _abc_impl = <_abc_data object>
    abstract load_feature(name)
        Parameters name (str) –
        Return type Optional[Any]
```

### annize.project.parser module

Reading, processing and writing of Annize projects.

```
class annize.project.parser._XmlParser
    Bases: object
    ATTRIBUTE_COMMAND_END = '~'
    ATTRIBUTE_COMMAND_START = '~'

    class _Context
        Bases: object
        static _Context__nodeshort(xnode, maxlen=100)
            Parameters xnode (xml.etree.ElementTree.Element) –
        in_file(fpath)
        in_node(xnode)

    class _TagParts(name, namespace="")
        Bases: object
        Parameters namespace (str) –

    classmethod _XmlParser__interpret_attribute_string(txt)
        Parameters txt (str) –
        Return type Tuple[bool, str]

    classmethod _XmlParser__parse_attrib(key, value)
        Parameters
        • key (str) –
        • value (str) –
```

Return type *annize.project.Node*

**\_XmlParser\_\_parse\_child** (*node*, *xnode*)

Parameters

- **node** (*annize.project.Node*) –
- **xnode** (*xml.etree.ElementTree.Element*) –

Return type *Tuple[annize.project.Node, xml.etree.ElementTree.Element]*

**\_XmlParser\_\_parse\_children** (*node*, *xparent*)

Parameters

- **node** (*annize.project.Node*) –
- **xparent** (*xml.etree.ElementTree.Element*) –

**\_XmlParser\_\_parse\_tag** (*node*, *argname*, *callname*, *feature*, *xnode*)

Parameters **node** (*annize.project.Node*) –

**classmethod escape\_attribute\_string** (*txt*)

Parameters **txt** (*str*) –

Return type *str*

**parse\_file** (*fpath*)

Parameters **fpath** (*str*) –

Return type *annize.project.FileNode*

*annize.project.parser.parse* (*path*)

Parameters **path** (*str*) –

Return type *List[annize.project.FileNode]*

## annize.userfeedback package

**class** *annize.userfeedback.NullUserFeedbackController*

Bases: *object*

**choice\_dialog** (\*\_)

**input\_dialog** (\*\_)

**message\_dialog** (\*\_)

**exception** *annize.userfeedback.UnsatisfiableUserFeedbackAttemptError*

Bases: *RuntimeError*

**class** *annize.userfeedback.UserFeedbackController*

Bases: *object*

**choice\_dialog** (*question*, *choices*, *config\_key*)

Parameters

- **question** (*str*) –
- **choices** (*List[str]*) –
- **config\_key** (*Optional[str]*) –

**Return type** Optional[int]

**input\_dialog** (*question, suggested\_answer, config\_key*)

**Parameters**

- **question** (*str*) –
- **suggested\_answer** (*str*) –
- **config\_key** (*Optional[str]*) –

**Return type** Optional[str]

**message\_dialog** (*message, answers, config\_key*)

**Parameters**

- **message** (*str*) –
- **answers** (*List[str]*) –
- **config\_key** (*Optional[str]*) –

**Return type** int

**annize.userfeedback.\_add\_controller\_to\_context** (\*, *controller, context, priority\_index=0*)

**Parameters**

- **controller** (*annize.userfeedback.UserFeedbackController*) –
- **context** (*annize.context.Context*) –
- **priority\_index** (*int*) –

**Return type** None

**annize.userfeedback.\_controllers\_for\_context** (*context*)

**Parameters** **context** (*annize.context.Context*) –

**Return type** List[*annize.userfeedback.UserFeedbackController*]

**annize.userfeedback.\_controllers\_tuples\_for\_context** (*context*)

**Parameters** **context** (*annize.context.Context*) –

**Return type** List[Tuple[int, *annize.userfeedback.UserFeedbackController*]]

**annize.userfeedback.choice\_dialog** (*message, choices, \*, config\_key=None*)

**Parameters**

- **message** (*Union[str, annize.i18n.TrStr]*) –
- **choices** (*Iterable[Union[str, annize.i18n.TrStr]]*) –
- **config\_key** (*Optional[str]*) –

**Return type** Optional[int]

**annize.userfeedback.input\_dialog** (*message, \*, suggested\_answer, config\_key=None*)

**Parameters**

- **message** (*Union[str, annize.i18n.TrStr]*) –
- **suggested\_answer** (*Union[str, annize.i18n.TrStr]*) –
- **config\_key** (*Optional[str]*) –

**Return type** Optional[str]

`annize.userfeedback.message_dialog(message, answers, *, config_key=None)`

**Parameters**

- **message** (Union[str, `annize.i18n.TrStr`]) –
- **answers** (Iterable[Union[str, `annize.i18n.TrStr`]]) –
- **config\_key** (Optional[str]) –

**Return type** int

## Submodules

### annize.userfeedback.static module

**class** `annize.userfeedback.static.StaticUserFeedbackController` (answers)

Bases: `annize.userfeedback.UserFeedbackController`

**Parameters** **answers** (Dict[str, Any]) –

`__get_answer` (config\_key)

**Parameters** **config\_key** (str) –

**Return type** Any

`add_answer` (config\_key, value)

**Parameters**

- **config\_key** (str) –
- **value** (Any) –

**Return type** None

`choice_dialog` (question, choices, config\_key)

`input_dialog` (question, suggested\_answer, config\_key)

`message_dialog` (message, answers, config\_key)

### annize.userfeedback.tty module

**class** `annize.userfeedback.tty.TtyUserFeedbackController`

Bases: `annize.userfeedback.UserFeedbackController`

`__action_line` (num, text)

**Parameters**

- **num** (Any) –
- **text** (str) –

**Return type** str

`__dialog` (message, config\_key, actiontext)

**Parameters**

- **message** (str) –

- **config\_key** (*str*) –
- **actionstext** (*str*) –

**Return type** str

**\_\_dialog\_frame\_actions** (*text*)

**Parameters** **text** (*str*) –

**Return type** str

**\_\_dialog\_frame\_configkey** (*config\_key*)

**Parameters** **config\_key** (*str*) –

**Return type** str

**\_\_dialog\_frame\_message** (*message*)

**Parameters** **message** (*str*) –

**Return type** str

**choice\_dialog** (*question, choices, config\_key*)

**input\_dialog** (*question, suggested\_answer, config\_key*)

**message\_dialog** (*message, answers, config\_key*)

## 6.1.2 Submodules

### 6.1.3 annize.annize\_cli module

```
class annize.annize_cli.Cli
    Bases: object

    classmethod __answers_from_json_files (destination, with_answers_from_json_files)
        Parameters
            • destination (Dict) –
            • with_answers_from_json_files (Iterable[str]) –

    classmethod __answers_from_json_strings (destination, with_answers_from_json_strings)
        Parameters
            • destination (Dict) –
            • with_answers_from_json_strings (Iterable[str]) –

    classmethod __answers_from_single_answers (destination, with_answers)
        Parameters
            • destination (Dict) –
            • with_answers (Iterable[Tuple[str, str]]) –

    classmethod do (project, taskname, with_answers_from_json_file, with_answers_from_json_string,
                    with_answer, **_)
        Parameters
            • project (str) –
```

```

        • taskname (str) –
        • with_answers_from_json_file (Iterable[str]) –
        • with_answers_from_json_string (Iterable[str]) –
        • with_answer (Iterable[Tuple[str, str]]) –

    project_default = '/home/pino/projects/annize/src/annize'
annize.annize_cli.get_parser()

    Return type argparse.ArgumentParser

annize.annize_cli.main()

```

## 6.1.4 annize.engine module

```

class annize.engine.LoadedProject (context, null_culture)
    Bases: object

    Parameters

        • context (annize.context.Context) –
        • null_culture (annize.i18n.Culture) –

    property context

annize.engine._prepare_context (context, *, annize_config_rootpath)

    Parameters

        • context (annize.context.Context) –
        • annize_config_rootpath (str) –

    Return type None

annize.engine.find_project_annize_config_root_file (project_path)

    Parameters project_path (str) –

    Return type Optional[str]

annize.engine.guess_project_root_directory (annize_config_rootpath)

    Parameters annize_config_rootpath (str) –

    Return type str

annize.engine.load_project (project_path)

    Parameters project_path (str) –

    Return type annize.engine.LoadedProject

annize.engine.load_project_definition (projectdefinition, *, annize_config_rootpath)

    Parameters

        • projectdefinition (annize.project.ProjectDefinition) –
        • annize_config_rootpath (str) –

    Return type annize.engine.LoadedProject

```

## PYTHON MODULE INDEX

### a

annize.annize\_cli, 97  
annize.aux, 13  
annize.aux.\_project\_infos, 13  
annize.callconv, 13  
annize.callconv.typehints, 14  
annize.context, 15  
annize.data, 18  
annize.data.color, 18  
annize.data.container, 19  
annize.data.unique, 19  
annize.data.version, 19  
annize.engine, 98  
annize.features.authors, 74  
annize.features.base, 74  
annize.features.changelog.common, 21  
annize.features.datainjections.common, 22  
annize.features.datainjections.python, 23  
annize.features.dependencies.common, 23  
annize.features.dependencies.python, 24  
annize.features.distributables.common, 25  
annize.features.distributables.debian, 26  
annize.features.distributables.flatpak, 39  
annize.features.distributables.pythonwheel, 44  
annize.features.distributables.store.pypi, 25  
annize.features.distributables.tar, 47  
annize.features.documentation.common, 59  
annize.features.documentation.sphinx.\_utils, 50  
annize.features.documentation.sphinx.common, 51  
annize.features.documentation.sphinx.cpp, 56  
annize.features.documentation.sphinx.doxygen\_compat, 57  
annize.features.documentation.sphinx.javascript, 58  
annize.features.documentation.sphinx.output.common, 48  
annize.features.documentation.sphinx.output.html, 48  
annize.features.documentation.sphinx.output.pdf, 50  
annize.features.documentation.sphinx.output.plaintext, 50  
annize.features.documentation.sphinx.python, 58  
annize.features.documentation.sphinx.rst, 59  
annize.features.files.common, 62  
annize.features.files.transfer.common, 61  
annize.features.files.transfer.ssh, 62  
annize.features.homepage.common, 67  
annize.features.homepage.sections.about, 65  
annize.features.homepage.sections.changelog, 65  
annize.features.homepage.sections.documentation, 65  
annize.features.homepage.sections.download, 66  
annize.features.homepage.sections.gallery, 66  
annize.features.homepage.sections.imprint, 67  
annize.features.homepage.sections.license, 67  
annize.features.i18n.common, 70  
annize.features.i18n.gettext, 71  
annize.features.licensing, 76  
annize.features.mediagalleries, 77  
annize.features.task, 78  
annize.features.testing.common, 71  
annize.features.testing.pylint, 72  
annize.features.testing.pytest, 72

`annize.features.testing.pyunit`, [72](#)  
`annize.features.version`, [78](#)  
`annize.features.versioncontrol.common`,  
    [72](#)  
`annize.features.versioncontrol.git`, [73](#)  
`annize.fs`, [78](#)  
`annize.fs.temp`, [83](#)  
`annize.il8n`, [83](#)  
`annize.project`, [86](#)  
`annize.project.featureloader`, [93](#)  
`annize.project.materializer`, [89](#)  
`annize.project.materializer.behaviors`,  
    [89](#)  
`annize.project.materializer.behaviors.argument`,  
    [89](#)  
`annize.project.materializer.behaviors.basket`,  
    [90](#)  
`annize.project.materializer.behaviors.block`,  
    [90](#)  
`annize.project.materializer.behaviors.featureunavailable`,  
    [90](#)  
`annize.project.materializer.behaviors.reference`,  
    [90](#)  
`annize.project.materializer.core`, [91](#)  
`annize.project.materializer.preprocessors`,  
    [92](#)  
`annize.project.parser`, [93](#)  
`annize.userfeedback`, [94](#)  
`annize.userfeedback.static`, [96](#)  
`annize.userfeedback.tty`, [96](#)



## Symbols

\_CallHelper (class in *annize.callconv*), 13  
 \_CallHelper\_\_convert\_kwargs\_from\_string() (annize.callconv.\_CallHelper static method), 14  
 \_CallHelper\_\_shift\_args\_to\_kwargs() (annize.callconv.\_CallHelper static method), 14  
 \_Context\_\_nodeshort() (annize.project.parser.\_XmlParser.\_Context static method), 93  
 \_DocumentGenerateAllLanguagesResult (class in annize.features.documentation.common), 60  
 \_DocumentGenerateResult (class in annize.features.documentation.common), 61  
 \_FEATURES\_NAMESPACE (annize.project.featureloader.DefaultFeatureLoader attribute), 93  
 \_METADATA\_IS\_TOPLEVEL\_OBJECT (annize.context.Context attribute), 15  
 \_S\_CHANGE (annize.features.changelog.common.ByVersionControlSystemCommitsMessagesChangelog attribute), 21  
 \_S\_LABEL (annize.features.changelog.common.ByVersionControlSystemCommitsMessagesChangelog attribute), 22  
 \_SimpleFsEntry (class in *annize.fs*), 81  
 \_TempDir (class in *annize.fs.temp*), 83  
 \_TempDir\_\_cleanup() (annize.fs.temp.\_TempDir method), 83  
 \_TransferHelper (class in *annize.fs*), 81  
 \_TransferHelper\_\_transfer\_dirmerge() (annize.fs.\_TransferHelper method), 81  
 \_TransferHelper\_\_transfer\_prepare() (annize.fs.\_TransferHelper method), 81  
 \_XmlParser (class in *annize.project.parser*), 93  
 \_XmlParser.\_Context (class in annize.project.parser), 93  
 \_XmlParser.\_TagParts (class in annize.project.parser), 93  
 \_XmlParser\_\_interpret\_attribute\_string() (annize.project.parser.\_XmlParser class method), 93  
 \_XmlParser\_\_parse\_attr() (annize.project.parser.\_XmlParser class method), 93  
 \_XmlParser\_\_parse\_child() (annize.project.parser.\_XmlParser method), 94  
 \_XmlParser\_\_parse\_children() (annize.project.parser.\_XmlParser method), 94  
 \_XmlParser\_\_parse\_tag() (annize.project.parser.\_XmlParser method), 94  
 \_\_action\_line() (annize.userfeedback.tty.TtyUserFeedbackController method), 96  
 \_\_answers\_from\_json\_files() (annize.annize\_cli.Cli class method), 97  
 \_\_answers\_from\_json\_strings() (annize.annize\_cli.Cli class method), 97  
 \_\_answers\_from\_single\_answers() (annize.annize\_cli.Cli class method), 97  
 \_\_calls\_git() (annize.features.versioncontrol.git.VersionControlSystemCommitsMessagesChangelog method), 73  
 \_\_context\_catch\_exceptions() (annize.project.materializer.behaviors.featureunavailable.FeatureUnavailable method), 90  
 \_\_context\_skipnode\_featureignorelist() (annize.project.materializer.behaviors.featureunavailable.FeatureUnavailable method), 90  
 \_\_dialog() (annize.userfeedback.tty.TtyUserFeedbackController method), 96  
 \_\_dialog\_frame\_actions() (annize.userfeedback.tty.TtyUserFeedbackController method), 97  
 \_\_dialog\_frame\_configkey() (annize.userfeedback.tty.TtyUserFeedbackController method), 97  
 \_\_dialog\_frame\_message() (annize.userfeedback.tty.TtyUserFeedbackController method), 97  
 \_\_does\_exclude() (annize.features.files.common.Exclude method), 64

<code>__effversion()</code> <i>annize.features.versioncontrol.common.BuildVersion</i> <i>method</i> ), 72	<code>(an- __patch_property_types_in_docstrings()</code> <i>annize.features.documentation.sphinx.python.Python3ApiReferen</i> <i>method</i> ), 58
<code>__generate_packagelist()</code> <i>annize.features.homepage.sections.download.Section</i> <i>method</i> ), 66	<code>(an- __put_object()</code> ( <i>annize.context.Context</i> <i>method</i> ), 15 <code>__reference_is_hopeless()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>class method</i> ), 91
<code>__generate_postpreproc()</code> <i>annize.features.homepage.common.Homepage</i> <i>method</i> ), 67	<code>__scanjsfile()</code> ( <i>an-</i> <i>annize.features.documentation.sphinx.javascript.JavaScriptApiRefer</i> <i>method</i> ), 58
<code>__generate_section()</code> <i>annize.features.homepage.common.Homepage</i> <i>method</i> ), 67	<code>__shorten()</code> ( <i>annize.project.ScalarValueNode</i> <i>method</i> ), 88
<code>__get_answer()</code> <i>annize.userfeedback.static.StaticUserFeedbackController</i> <i>method</i> ), 96	<code>__store_files_to_packagestore()</code> ( <i>an-</i> <i>annize.features.distributables.common.Group</i> <i>method</i> ), 25
<code>__get_description()</code> ( <i>annize.project.Node</i> <i>method</i> ), 86	<code>__transfer_predicate_and()</code> ( <i>an-</i> <i>annize.features.files.common.Dir</i> <i>class method</i> ), 63
<code>__get_files_from_packagestore()</code> ( <i>an-</i> <i>annize.features.distributables.common.Group</i> <i>method</i> ), 25	<code>__transfer_predicate_for_exclude()</code> ( <i>an-</i> <i>annize.features.files.common.Dir</i> <i>class method</i> ), 63
<code>__get_inner_generateinfo()</code> ( <i>an-</i> <i>annize.features.documentation.sphinx.common.CompositeDocument</i> <i>method</i> ), 53	<code>__transfer_predicate_for_excludes()</code> ( <i>an-</i> <i>annize.features.files.common.Dir</i> <i>class method</i> ), 63
<code>__get_refgeninfo()</code> ( <i>an-</i> <i>annize.features.documentation.sphinx.common.ApiReferenceDocument</i> <i>method</i> ), 51	<code>__translations_for_stringname()</code> ( <i>an-</i> <i>annize.features.i18n.common.ProjectDefinedTranslationProvider</i> <i>method</i> ), 70
<code>__get_variant()</code> ( <i>an-</i> <i>annize.features.documentation.sphinx.common.RstDocument</i> <i>method</i> ), 55	<code>__uniqueid_counter</code> ( <i>annize.data.unique.UniqueId</i> <i>attribute</i> ), 19
<code>__getinner()</code> ( <i>annize.fs.AdHocFsEntry</i> <i>method</i> ), 78	<code>__uniqueid_lock</code> ( <i>annize.data.unique.UniqueId</i> <i>at-</i> <i>tribute</i> ), 19
<code>__getname()</code> ( <i>annize.features.distributables.common.Group</i> <i>method</i> ), 25	<code>__unresolvable_references()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91
<code>__info()</code> ( <i>annize.features.documentation.sphinx.doxygencompat.DoxygenSupportedApiReferenceBehavior</i> <i>method</i> ), 58	<code>__unresolvable_references()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91
<code>__jsfiles()</code> ( <i>annize.features.documentation.sphinx.javascript.JavaScriptApiReferenceBehavior</i> <i>method</i> ), 58	<code>__unresolvable_references()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91
<code>__long_str()</code> ( <i>annize.data.unique.UniqueId</i> <i>method</i> ), 19	<code>__unresolvable_references()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91
<code>__materialization_for_node()</code> ( <i>an-</i> <i>annize.project.materializer.core.ProjectMaterializer</i> <i>method</i> ), 92	<code>__unresolvable_references()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91
<code>__materialize()</code> ( <i>an-</i> <i>annize.project.materializer.core.ProjectMaterializer</i> <i>method</i> ), 92	<code>__unresolvable_references()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91
<code>__node_is_hopeless()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>class method</i> ), 90	<code>__unresolvable_references()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91
<code>__nodes_by_name()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91	<code>__unresolvable_references()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91
<code>__object_metadata_dict()</code> ( <i>an-</i> <i>annize.context.Context</i> <i>method</i> ), 15	<code>__unresolvable_references()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91
<code>__object_raw_name()</code> ( <i>annize.context.Context</i> <i>method</i> ), 15	<code>__unresolvable_references()</code> ( <i>an-</i> <i>annize.project.materializer.behaviors.reference.SkipUnresolvableRe</i> <i>method</i> ), 91

`_abc_impl` (`annize.features.documentation.common.GeneratedDocPage` attribute), 60  
`_abc_impl` (`annize.features.documentation.common.HtmlOutputSpec` attribute), 60  
`_abc_impl` (`annize.features.documentation.common.OutputSpec` attribute), 60  
`_abc_impl` (`annize.features.documentation.common.PdfOutputSpec` attribute), 60  
`_abc_impl` (`annize.features.documentation.common.PlainTextOutputSpec` attribute), 60  
`_abc_impl` (`annize.features.documentation.sphinx.common.AboutProjectNode` attribute), 51  
`_abc_impl` (`annize.features.documentation.sphinx.common.ApiReferenceLanguage` attribute), 52  
`_abc_impl` (`annize.features.documentation.sphinx.common.ApiReferenceLanguage` attribute), 52  
`_abc_impl` (`annize.features.documentation.sphinx.common.ArgumentBehavior` attribute), 53  
`_abc_impl` (`annize.features.documentation.sphinx.common.CompositionDocument` attribute), 53  
`_abc_impl` (`annize.features.documentation.sphinx.common.Document` attribute), 54  
`_abc_impl` (`annize.features.documentation.sphinx.common.ReadmeDocument` attribute), 55  
`_abc_impl` (`annize.features.documentation.sphinx.common.RstDocument` attribute), 55  
`_abc_impl` (`annize.features.documentation.sphinx.cpp.CppApiReferenceLanguage` attribute), 56  
`_abc_impl` (`annize.features.documentation.sphinx.doxygen.DoxygenSupportedApiReferenceLanguage` attribute), 58  
`_abc_impl` (`annize.features.documentation.sphinx.javascript.JavaScriptApiReferenceLanguage` attribute), 58  
`_abc_impl` (`annize.features.documentation.sphinx.output.common.OutputGenerator` attribute), 48  
`_abc_impl` (`annize.features.documentation.sphinx.output.html.HtmlOutputGenerator` attribute), 48  
`_abc_impl` (`annize.features.documentation.sphinx.output.html.HtmlOutputSpec` attribute), 49  
`_abc_impl` (`annize.features.documentation.sphinx.output.pdf.PdfOutputGenerator` attribute), 50  
`_abc_impl` (`annize.features.documentation.sphinx.output.plaintext.PlainTextOutputGenerator` attribute), 50  
`_abc_impl` (`annize.features.documentation.sphinx.python.Python3ApiReferenceLanguage` attribute), 58  
`_abc_impl` (`annize.features.files.common.Dir` attribute), 63  
`_abc_impl` (`annize.features.files.common.File` attribute), 64  
`_abc_impl` (`annize.features.files.common.FsEntry` attribute), 64  
`_abc_impl` (`annize.features.files.common.MachineRootDirectory` attribute), 64  
`_abc_impl` (`annize.features.files.common.ProjectDir` attribute), 64  
`_abc_impl` (`annize.features.homepage.common.GeneratedHomepage` attribute), 67  
`_abc_impl` (`annize.fs.AdHocFsEntry` attribute), 78  
`_abc_impl` (`annize.fs.FsEntry` attribute), 79  
`_abc_impl` (`annize.fs.SimpleFsEntry` attribute), 81  
`_abc_impl` (`annize.fs.temp.TempFileByContent` attribute), 83  
`_abc_impl` (`annize.fs.temp.TempDir` attribute), 83  
`_abc_impl` (`annize.project.featureloader.DefaultFeatureLoader` attribute), 93  
`_abc_impl` (`annize.project.featureloader.FeatureLoader` attribute), 93  
`_abc_impl` (`annize.project.materializer.behaviors.Behavior` attribute), 89  
`_abc_impl` (`annize.project.materializer.behaviors.argument.ArgumentParser` attribute), 90  
`_abc_impl` (`annize.project.materializer.behaviors.basket.BasketBehavior` attribute), 90  
`_abc_impl` (`annize.project.materializer.behaviors.block.BlockBehavior` attribute), 90  
`_abc_impl` (`annize.project.materializer.behaviors.featureunavailable.FeatureUnavailableBehavior` attribute), 90  
`_abc_impl` (`annize.project.materializer.behaviors.reference.ReferenceBehavior` attribute), 90  
`_abc_impl` (`annize.project.materializer.behaviors.reference.SkipUnresolvedReferenceBehavior` attribute), 91  
`_context()` (in module `annize.userfeedback`), 95  
`_allowed_child_types()` (`annize.project.BlockNode` class method), 86  
`_allowed_child_types()` (`annize.project.CallableNode` class method), 86  
`_allowed_child_types()` (`annize.project.Node` class method), 87  
`_allowed_child_types()` (`annize.project.ReferenceNode` class method), 88  
`_authors()` (`annize.features.documentation.sphinx.common.Document` property), 54  
`_controllers_for_context()` (in module `annize.userfeedback`), 95  
`_controllers_tuples_for_context()` (in module `annize.userfeedback`), 95  
`_debian_category()` (in module `annize.features.distributables.debian`), 39  
`_debian_section()` (in module `annize.features.distributables.debian`), 39  
`_description_for_mediafile()` (in module `annize.features.distributables.debian`), 39

```

nize.features.mediagalleries.Gallery method), _initialize() (an-
77 nize.features.distributables.flatpak.GpgFile
_generate() (annize.features.homepage.common.HomepageSection method), 42
method), 69 _initialize() (an-
_generate() (annize.features.homepage.common.PinoHomepageApiReference method), 46
method), 70 nize.features.distributables.pythonwheel.Package
_generate() (annize.features.homepage.sections.about.Section initialize() (an-
method), 65 nize.features.distributables.tar.Package
_generate() (annize.features.homepage.sections.changelog.Section method), 47
method), 65 _initialize() (an-
_generate() (annize.features.homepage.sections.documentation.Section initialize() (an-
method), 65 nize.features.documentation.common.GeneratedDocument
_generate() (annize.features.homepage.sections.download.Section initialize() (annize.features.files.common.Dir
method), 66 method), 63
_generate() (annize.features.homepage.sections.gallery.Section initialize() (an-
method), 66 nize.features.files.common.FsEntry method),
_generate() (annize.features.homepage.sections.imprint.Section 64
method), 67 _initialize() (an-
_generate() (annize.features.homepage.sections.license.Section nize.features.files.common.MachineRootDirectory
method), 67 method), 64
_generate_master_doc() (an- _initialize() (an-
nize.features.documentation.sphinx.common.AboutProjectDir initialize() (an-
method), 51 64 nize.features.files.common.ProjectDir method),
_generate_master_doc() (an- _initialize() (an-
nize.features.documentation.sphinx.common.ApiReferenceDir initialize() (an-
method), 52 nize.features.homepage.common.GeneratedHomepage
method), 67
_generate_master_doc() (an- _initialize() (annize.fs.AdHocFsEntry method),
nize.features.documentation.sphinx.common.ArgparseCommand 78
method), 53 _initialize() (annize.fs.temp.TempFileByContent
_generate_master_doc() (an- method), 83
nize.features.documentation.sphinx.common.CompositeDocumentcher() (in module an-
method), 53 nize.features.changelog.common), 22
_generate_master_doc() (an- _license() (in module annize.features.licensing), 77
nize.features.documentation.sphinx.common.Document materialize_hlp_childobjs() (an-
method), 54 nize.project.materializer.core.ProjectMaterializer
_generate_master_doc() (an- method), 92
nize.features.documentation.sphinx.common.RstDocument package() (annize.features.distributables.debian.Package
method), 55 class method), 36
_generate_master_doc() (an- _mkpackage() (annize.features.distributables.flatpak.FlatpakImage
nize.features.documentation.sphinx.doxygencompat.DoxygenSupportApiReferenceLanguage
method), 58 _mkpackage() (annize.features.distributables.pythonwheel.Package
_get_data() (in module annize.features.base), 75 class method), 46
_initialize() (an- _mkpackage_applysource() (an-
nize.features.datainjections.common.Injection nize.features.distributables.flatpak.FlatpakImage
method), 22 class method), 41
_initialize() (an- _mkpackage_bdist_wheel() (an-
nize.features.distributables.debian.Package nize.features.distributables.pythonwheel.Package
method), 36 class method), 46
_initialize() (an- _mkpackage_correctbuildsourcepermissions()
nize.features.distributables.flatpak.FlatpakImage (annize.features.distributables.debian.Package
method), 41 class method), 36
_initialize() (an- _mkpackage_determinesize() (an-
nize.features.distributables.flatpak.FlatpakrefFile nize.features.distributables.debian.Package
method), 42 class method), 36

```

<code>_mkpackage_dpkg()</code>	( <i>annize.features.distributables.debian.Package</i> class method), 36	<code>_mkpackage_setuppyconf_classifiers()</code>	( <i>annize.features.distributables.pythonwheel.Package</i> class method), 46
<code>_mkpackage_flatpak_build_export()</code>	( <i>annize.features.distributables.flatpak.FlatpakImage</i> class method), 41	<code>_mkpackage_setuppyconf_install_requires()</code>	( <i>annize.features.distributables.pythonwheel.Package</i> class method), 47
<code>_mkpackage_flatpak_build_finish()</code>	( <i>annize.features.distributables.flatpak.FlatpakImage</i> class method), 41	<code>_mkpackage_setuppyconf_prepare()</code>	( <i>annize.features.distributables.pythonwheel.Package</i> class method), 47
<code>_mkpackage_flatpak_build_init()</code>	( <i>annize.features.distributables.flatpak.FlatpakImage</i> class method), 41	<code>_mkpackage_share()</code>	( <i>annize.features.distributables.flatpak.FlatpakImage</i> class method), 42
<code>_mkpackage_mkchangelog()</code>	( <i>annize.features.distributables.debian.Package</i> class method), 36	<code>_name_anon()</code>	( <i>annize.data.version.AbstractVersionPatternSegment</i> method), 19
<code>_mkpackage_mkcopyright()</code>	( <i>annize.features.distributables.debian.Package</i> class method), 36	<code>_name_anon()</code>	( <i>annize.data.version.ConcatenatedVersionPatternSegment</i> method), 20
<code>_mkpackage_mkdebianconffilesfile()</code>	( <i>annize.features.distributables.debian.Package</i> class method), 36	<code>_name_anon()</code>	( <i>annize.data.version.NumericVersionPatternSegment</i> method), 20
<code>_mkpackage_mkdebiancontrolfile()</code>	( <i>annize.features.distributables.debian.Package</i> class method), 37	<code>_name_anon()</code>	( <i>annize.data.version.OptionalVersionPatternSegment</i> method), 20
<code>_mkpackage_mkexeclinks()</code>	( <i>annize.features.distributables.debian.Package</i> class method), 37	<code>_prepare_context()</code>	(in module <i>annize.engine</i> ), 98
<code>_mkpackage_mkexeclinks()</code>	( <i>annize.features.distributables.pythonwheel.Package</i> class method), 46	<code>_prepare_generate()</code>	( <i>annize.features.documentation.sphinx.common.ApiReferenceDocument</i> method), 52
<code>_mkpackage_mkmanifestin()</code>	( <i>annize.features.distributables.pythonwheel.Package</i> class method), 46	<code>_prepare_generate()</code>	( <i>annize.features.documentation.sphinx.common.ArgparseCommandL</i> method), 53
<code>_mkpackage_mkmenuentries()</code>	( <i>annize.features.distributables.debian.Package</i> class method), 37	<code>_prepare_generate()</code>	( <i>annize.features.documentation.sphinx.common.CompositeDocument</i> method), 53
<code>_mkpackage_mkprepostcmds()</code>	( <i>annize.features.distributables.debian.Package</i> class method), 37	<code>_prepare_generate()</code>	( <i>annize.features.documentation.sphinx.common.Document</i> method), 54
<code>_mkpackage_mkservices()</code>	( <i>annize.features.distributables.debian.Package</i> class method), 37	<code>_projectname()</code>	( <i>annize.features.documentation.sphinx.common.Document</i> property), 54
<code>_mkpackage_mksetuppyconf()</code>	( <i>annize.features.distributables.pythonwheel.Package</i> class method), 46	<code>_release()</code>	( <i>annize.features.documentation.sphinx.common.Document</i> property), 54
<code>_mkpackage_prepareinfos()</code>	( <i>annize.features.distributables.debian.Package</i> class method), 37	<code>_set_entry_path()</code>	( <i>annize.features.documentation.common._DocumentGenerateResult</i> method), 61
<code>_mkpackage_prepareinfos()</code>	( <i>annize.features.distributables.flatpak.FlatpakImage</i> class method), 41	<code>_str_helper()</code>	( <i>annize.project.ArgumentNode</i> method), 86
<code>_mkpackage_prepareinfos()</code>	( <i>annize.features.distributables.pythonwheel.Package</i> class method), 46	<code>_str_helper()</code>	( <i>annize.project.CallableNode</i> method), 86
		<code>_str_helper()</code>	( <i>annize.project.FileNode</i> method), 86
		<code>_str_helper()</code>	( <i>annize.project.Node</i> method), 87
		<code>_str_helper()</code>	( <i>annize.project.OnFeatureUnavailableNode</i> method), 87
		<code>_str_helper()</code>	( <i>annize.project.ReferenceNode</i> method), 88
		<code>_str_helper()</code>	( <i>annize.project.ScalarValueNode</i> method), 88



`_str_helper()` (*annize.project.ScopableBlockNode* method), 89  
`_str_to_val()` (*annize.data.version.AbstractVersionPatternSegment* method), 19  
`_str_to_val()` (*annize.data.version.NumericVersionPatternSegment* method), 20  
`_to_dict()` (*annize.features.documentation.sphinx.common.DocumentGenerator* method), 54  
`_version()` (*annize.features.documentation.sphinx.common.Document* property), 54

## A

`AboutProjectDocument` (class in *annize.features.documentation.sphinx.common*), 51  
`AbstractVersionPatternSegment` (class in *annize.data.version*), 19  
`access_filesystem()` (*annize.features.files.transfer.common.Endpoint* method), 61  
`access_filesystem()` (*annize.features.files.transfer.common.FsEndpoint* method), 61  
`access_filesystem()` (*annize.features.files.transfer.ssh.Endpoint* method), 62  
`add_answer()` (*annize.userfeedback.static.StaticUserFeedbackController* method), 96  
`add_object()` (*annize.context.Context* method), 15  
`add_object()` (in module *annize.context*), 17  
`add_translation_provider()` (in module *annize.i18n*), 85  
`add_translations()` (*annize.features.i18n.common.ProjectDefinedTranslationProvider* method), 70  
`additional_info()` (*annize.features.licensing.License* method), 77  
`AdHocFsEntry` (class in *annize.fs*), 78  
`AdministrationUtilitiesSection` (in module *annize.features.distributables.debian*), 26  
`AFLv3` (in module *annize.features.licensing*), 76  
`AGPLv3` (in module *annize.features.licensing*), 76  
`always_true()` (in module *annize.callconv*), 14  
`annize.annize_cli` module, 97  
`annize.aux` module, 13  
`annize.aux._project_infos` module, 13  
`annize.callconv` module, 13  
`annize.callconv.typehints` module, 14  
`annize.context` module, 15  
`annize.data` module, 18  
`annize.data.color` module, 18  
`annize.data.unique` module, 19  
`annize.data.version` module, 19  
`annize.engine` module, 98  
`annize.features.authors` module, 74  
`annize.features.base` module, 74  
`annize.features.changelog.common` module, 21  
`annize.features.datainjections.common` module, 22  
`annize.features.datainjections.python` module, 23  
`annize.features.dependencies.common` module, 23  
`annize.features.dependencies.python` module, 24  
`annize.features.distributables.common` module, 25  
`annize.features.distributables.debian` module, 26  
`annize.features.distributables.flatpak` module, 39  
`annize.features.distributables.pythonwheel` module, 44  
`annize.features.distributables.store.pypi` module, 25  
`annize.features.distributables.tar` module, 47  
`annize.features.documentation.common` module, 59  
`annize.features.documentation.sphinx._utils` module, 50  
`annize.features.documentation.sphinx.common` module, 51  
`annize.features.documentation.sphinx.cpp` module, 56  
`annize.features.documentation.sphinx.doxygencompat` module, 57  
`annize.features.documentation.sphinx.javascript` module, 58

annize.features.documentation.sphinx.outmodule, 48  
 annize.features.documentation.sphinx.outmodule, 48  
 annize.features.documentation.sphinx.outmodule, 50  
 annize.features.documentation.sphinx.outmodule, 50  
 annize.features.documentation.sphinx.pythannize.i18n  
 module, 58 module, 83  
 annize.features.documentation.sphinx.rstannize.project  
 module, 59 module, 86  
 annize.features.files.common annize.project.featureloader  
 module, 62 module, 93  
 annize.features.files.transfer.common annize.project.materializer  
 module, 61 module, 89  
 annize.features.files.transfer.ssh annize.project.materializer.behaviors  
 module, 62 module, 89  
 annize.features.homepage.common annize.project.materializer.behaviors.argument  
 module, 67 module, 89  
 annize.features.homepage.sections.about annize.project.materializer.behaviors.basket  
 module, 65 module, 90  
 annize.features.homepage.sections.changelog annize.project.materializer.behaviors.block  
 module, 65 module, 90  
 annize.features.homepage.sections.documentannize.project.materializer.behaviors.featureunava  
 module, 65 module, 90  
 annize.features.homepage.sections.downloadannize.project.materializer.behaviors.reference  
 module, 66 module, 90  
 annize.features.homepage.sections.galleryannize.project.materializer.core  
 module, 66 module, 91  
 annize.features.homepage.sections.imprintannize.project.materializer.preprocessors  
 module, 67 module, 92  
 annize.features.homepage.sections.licenseannize.project.parser  
 module, 67 module, 93  
 annize.features.i18n.common annize.userfeedback  
 module, 70 module, 94  
 annize.features.i18n.gettext annize.userfeedback.static  
 module, 71 module, 96  
 annize.features.licensing annize.userfeedback.tty  
 module, 76 module, 96  
 annize.features.meddiagalleries annize\_user\_interaction\_culture() (in  
 module, 77 module annize.i18n), 85  
 annize.features.task Apache2 (in module annize.features.licensing), 76  
 module, 78 ApiReferenceDocument (class in an-  
 annize.features.testing.common nize.features.documentation.sphinx.common),  
 module, 71 51  
 annize.features.testing.pylint ApiReferenceLanguage (class in an-  
 module, 72 nize.features.documentation.sphinx.common),  
 52  
 annize.features.testing.pytest ApiReferenceLanguage.ApiReferenceGenerateInfo  
 module, 72 (class in an-  
 annize.features.testing.pyunit nize.features.documentation.sphinx.common),  
 module, 72 52  
 annize.features.version append\_child() (annize.project.Node method), 87  
 module, 78

[append\\_rst\(\)](#) (*annize.features.homepage.common.HomepageSectionNode* method), 68  
[append\\_to\(\)](#) (*annize.project.ArgumentNode* property), 86  
[ApplicationsAccessibilityCategory](#) (in module *annize.features.distributables.debian*), 26  
[ApplicationsAmateurradioCategory](#) (in module *annize.features.distributables.debian*), 26  
[ApplicationsDatamanagementCategory](#) (in module *annize.features.distributables.debian*), 26  
[ApplicationsEditorsCategory](#) (in module *annize.features.distributables.debian*), 26  
[ApplicationsEducationCategory](#) (in module *annize.features.distributables.debian*), 26  
[ApplicationsEmulatorsCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsFilemanagementCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsGraphicsCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsMobiledevicesCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsNetworkCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsNetworkCommunicationCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsNetworkFiletransferCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsNetworkMonitoringCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsNetworkWebbrowsingCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsNetworkWebnewsCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsOfficeCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsProgrammingCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsProjectmanagementCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsScienceAstronomyCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsScienceBiologyCategory](#) (in module *annize.features.distributables.debian*), 27  
[ApplicationsScienceChemistryCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsScienceDataanalysisCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsScienceElectronicsCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsScienceEngineeringCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsScienceGeoscienceCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsScienceMathematicsCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsScienceMedicineCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsSciencePhysicsCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsScienceSocialCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsShellsCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsSoundsCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsSystemAdministrationCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsSystemCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsSystemHardwareCategory](#) (in module *annize.features.distributables.debian*), 28  
[ApplicationsSystemLanguageenvironmentCategory](#) (in module *annize.features.distributables.debian*), 29  
[ApplicationsSystemMonitoringCategory](#) (in module *annize.features.distributables.debian*), 29  
[ApplicationsSystemPackagemanagementCategory](#) (in module *annize.features.distributables.debian*), 29  
[ApplicationsSystemSecurityCategory](#) (in module *annize.features.distributables.debian*), 29



29

ApplicationsTerminalemulatorsCategory (in module *annize.features.distributables.debian*), 29

ApplicationsTextCategory (in module *annize.features.distributables.debian*), 29

ApplicationsTvandradiocategory (in module *annize.features.distributables.debian*), 29

ApplicationsVideoCategory (in module *annize.features.distributables.debian*), 29

ApplicationsViewersCategory (in module *annize.features.distributables.debian*), 29

ApplicationsWebdevelopmentCategory (in module *annize.features.distributables.debian*), 29

architecture (*annize.features.distributables.debian.Package\_BuildInfo* attribute), 35

arg\_name () (*annize.project.ArgumentNode* property), 86

ArgparseCommandLineInterfaceDocument (class in *annize.features.documentation.sphinx.common*), 52

ArgumentBehavior (class in *annize.project.materializer.behaviors.argument*), 89

ArgumentNode (class in *annize.project*), 86

Artistic1 (in module *annize.features.licensing*), 76

Artwork (class in *annize.features.dependencies.common*), 23

attach\_media\_file () (*annize.features.homepage.common.HomepageSection\_Product* method), 68

ATTRIBUTE\_COMMAND\_END (*annize.project.parser.\_XmlParser* attribute), 93

ATTRIBUTE\_COMMAND\_START (*annize.project.parser.\_XmlParser* attribute), 93

author (*annize.features.distributables.debian.Package\_BuildInfo* attribute), 35

author (*annize.features.distributables.pythonwheel.Package\_BuildInfo* attribute), 45

Author (class in *annize.features.authors*), 74

authorstring (*annize.features.distributables.debian.Package\_BuildInfo* attribute), 35

available\_languages () (*annize.features.documentation.common.Document* method), 59

available\_languages () (*annize.features.documentation.sphinx.common.AboutProjectDocument* method), 51

available\_languages () (*annize.features.documentation.sphinx.common.ApiReferenceDocument* method), 52

available\_languages () (*annize.features.documentation.sphinx.common.ArgparseCommandL* method), 53

available\_languages () (*annize.features.documentation.sphinx.common.ReadmeDocument* method), 55

available\_languages () (*annize.features.documentation.sphinx.common.RstDocument* method), 55

## B

background\_image () (*annize.features.documentation.sphinx.output.html.HtmlOutputSpec* property), 49

BaseBuildingError, 78

BadStructureError, 86

basename () (*annize.fs.FsEntry* method), 79

Basket (class in *annize.data.container*), 19

Basket (class in *annize.features.base*), 74

BasketBehavior (class in *annize.project.materializer.behaviors.basket*), 90

Behavior (class in *annize.project.materializer.behaviors*), 89

BLOCK (*annize.project.ScopableBlockNode.Scope* attribute), 88

BlockBehavior (class in *annize.project.materializer.behaviors.block*), 90

BlockNode (class in *annize.project*), 86

BrandColor (*annize.data.color.Color* property), 18

brand\_color () (in module *annize.features.base*), 75

BrandColor (class in *annize.features.base*), 74

BSD2clause (in module *annize.features.licensing*), 76

BSD3clause (in module *annize.features.licensing*), 76

BuildVersion (class in *annize.features.versioncontrol.common*), 72

ByVersionControlSystemCommitMessagesChangelog (class in *annize.features.changelog.common*), 21

## C

call () (*annize.callconv.\_CallHelper* class method), 14

call () (in module *annize.callconv*), 14

CallableNode (class in *annize.project*), 86

callee () (*annize.project.CallableNode* property), 86

Category (class in *annize.features.distributables.debian*), 29

category () (*annize.features.distributables.debian.MenuEntry* property), 33

Cc0v1 (in module *annize.features.licensing*), 76

CcBy3 (in module *annize.features.licensing*), 76

CcByNcs (in module *annize.features.licensing*), 76

CcByNcNd3 (in module *annize.features.licensing*), 76  
 CcByNcSa3 (in module *annize.features.licensing*), 76  
 CcByNd3 (in module *annize.features.licensing*), 76  
 CcBySa3 (in module *annize.features.licensing*), 76  
 Changelog (class in *annize.features.changelog.common*), 22  
 ChDir (class in *annize.fs*), 78  
 child\_by\_relative\_path() (*annize.fs.FsEntry* method), 79  
 children() (*annize.fs.FsEntry* method), 79  
 children() (*annize.project.Node* property), 87  
 choice\_dialog() (*annize.userfeedback.NullUserFeedbackController* method), 94  
 choice\_dialog() (*annize.userfeedback.static.StaticUserFeedbackController* method), 96  
 choice\_dialog() (*annize.userfeedback.tty.TtyUserFeedbackController* method), 97  
 choice\_dialog() (*annize.userfeedback.UserFeedbackController* method), 94  
 choice\_dialog() (in module *annize.userfeedback*), 95  
 Cli (class in *annize.annize\_cli*), 97  
 clone() (*annize.project.Node* method), 87  
 Color (class in *annize.data.color*), 18  
 command (*annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo* attribute), 40  
 command() (*annize.features.distributables.debian.MenuEntry* property), 33  
 comment() (*annize.features.dependencies.common.Dependency* property), 23  
 CommonVersionPattern (class in *annize.features.version*), 78  
 CommunicationProgramsSection (in module *annize.features.distributables.debian*), 29  
 CompositeDocument (class in *annize.features.documentation.sphinx.common*), 53  
 ConcatenatedVersionPatternSegment (class in *annize.data.version*), 19  
 confdir() (*annize.features.documentation.sphinx.common.DocumentGeneratorInfo* property), 54  
 config\_files (*annize.features.distributables.debian.Package.\_BuildInfo* attribute), 35  
 configlines() (*annize.features.documentation.sphinx.common.DocumentGeneratorInfo* property), 54  
 configvalues() (*annize.features.documentation.sphinx.common.DocumentGeneratorInfo* property), 54  
 Connection (class in *annize.features.distributables.store.pypi*), 25  
 construct\_from\_string() (*annize.callconv.typehints.TypeInfo* method), 15  
 Context (class in *annize.context*), 15  
 context() (*annize.engine.LoadedProject* property), 98  
 copy\_to() (*annize.fs.FsEntry* method), 79  
 CppApiReferenceLanguage (class in *annize.features.documentation.sphinx.cpp*), 56  
 created\_time() (*annize.fs.FsEntry* method), 79  
 culture (*annize.features.homepage.common.HomepageSection.\_Generate* attribute), 69  
 Culture (class in *annize.i18n*), 83  
 current() (in module *annize.context*), 17  
 current\_culture() (in module *annize.i18n*), 85  
 customarg (*annize.features.homepage.common.HomepageSection.\_Generate* attribute), 69  
 customarg (*annize.features.homepage.common.HomepageSection.\_Prepare* attribute), 69

## D

Data (class in *annize.features.base*), 74  
 DatabasesSection (in module *annize.features.distributables.debian*), 29  
 DateTime (class in *annize.features.base*), 75  
 debian\_name() (*annize.features.distributables.debian.Category* property), 29  
 DebianInstallerUdebPackagesSection (in module *annize.features.distributables.debian*), 30  
 DebugPackagesSection (in module *annize.features.distributables.debian*), 30  
 default\_changelog() (in module *annize.features.changelog.common*), 22  
 default\_project\_authors() (in module *annize.features.authors*), 74  
 default\_version\_control\_system() (in module *annize.features.versioncontrol.common*), 73  
 default\_version\_pattern() (in module *annize.features.version*), 78  
 DefaultFeatureLoader (class in *annize.features.project.featureloader*), 93  
 dependencies\_to\_rst\_text() (in module *annize.features.dependencies.common*), 24  
 Dependency (class in *annize.features.dependencies.common*), 23  
 description (*annize.features.distributables.debian.Package.\_BuildInfo* attribute), 35  
 description (*annize.features.distributables.pythonwheel.Package.\_BuildInfo* attribute), 45  
 description() (*annize.features.distributables.common.Group* method), 45

property), 26  
description() (annize.features.distributables.flatpak.Group property), 43  
description() (annize.features.mediagalleries.Gallery.Item property), 77  
destination() (annize.fs.Mount property), 81  
destination\_is\_parent() (annize.features.files.common.DirPart property), 63  
destination\_path() (annize.features.files.common.DirPart property), 63  
detect\_retry\_useful\_node\_context() (annize.project.materializer.behaviors.reference.ReferenceBehavior method), 90  
DevelopmentSection (in module annize.features.distributables.debian), 30  
Dir (class in annize.features.files.common), 62  
DirPart (class in annize.features.files.common), 63  
disowned() (annize.fs.FsEntry method), 79  
do() (annize.annize\_cli.Cli class method), 97  
do() (annize.project.OnFeatureUnavailableNode property), 87  
Document (class in annize.features.documentation.common), 59  
Document (class in annize.features.documentation.sphinx.common), 53  
Document.GenerateInfo (class in annize.features.documentation.sphinx.common), 54  
document\_root\_directory (annize.features.homepage.common.HomepageSection attribute), 69  
document\_root\_url (annize.features.homepage.common.HomepageSection attribute), 69  
document\_root\_url (annize.features.homepage.common.HomepageSection attribute), 69  
document\_variant\_directory (annize.features.homepage.common.HomepageSection attribute), 69  
document\_variant\_url (annize.features.homepage.common.HomepageSection attribute), 69  
documentation\_source (annize.features.distributables.debian.Package.\_BuildInfo attribute), 35  
DocumentationSection (in module annize.features.distributables.debian), 30  
does\_exclude() (annize.features.files.common.Exclude method), 64  
does\_exclude() (annize.features.versioncontrol.git.ExcludeByGitIgnores method), 73  
DoxygenSupportedApiReferenceLanguage (class in annize.features.documentation.sphinx.doxygencompat), 57  
E  
EditorsSection (in module annize.features.distributables.debian), 30  
EducationSection (in module annize.features.distributables.debian), 30  
EmailsSection (in module annize.features.distributables.debian), 30  
email() (annize.features.authors.Author property), 74  
EmbeddedSoftwareSection (in module annize.features.distributables.debian), 30  
Endpoint (class in annize.features.files.transfer.common), 61  
Endpoint (class in annize.features.files.transfer.ssh), 62  
english\_lang\_name() (annize.i18n.Culture property), 83  
entries() (annize.features.changelog.common.ByVersionControlSystem property), 22  
entries() (annize.features.changelog.common.Changelog property), 22  
Entry (class in annize.features.changelog.common), 22  
entry\_path() (annize.features.documentation.common.\_DocumentGenerateAllLanguage (annize.features.documentation.common.\_DocumentGenerateAllLanguage method), 60  
entry\_path() (annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo attribute), 40  
EnvironmentVariable (class in annize.features.distributables.flatpak), 39  
escape\_attribute\_string() (annize.project.parser.XmlParser class method), 94  
Exclude (class in annize.features.files.common), 63  
ExcludeByGitIgnores (class in annize.features.versioncontrol.git), 73  
excludes() (annize.features.files.common.DirPart property), 63  
exec() (annize.features.files.transfer.ssh.Endpoint method), 62  
executable\_links (annize.features.distributables.debian.Package.\_BuildInfo attribute), 35

executable\_links (an- FlatpakImage.\_BuildInfo (class in an-  
nize.features.distributables.pythonwheel.Package.  
attribute), 45 \_BuildInfo (class in an-  
nize.features.distributables.flatpak), 40  
ExecutableLink (class in an- FlatpakrefFile (class in an-  
nize.features.distributables.flatpak), 42  
ExecutableLink (class in an- FontsSection (in module an-  
nize.features.distributables.debian), 30  
nize.features.distributables.pythonwheel),  
format () (annize.i18n.TrStr method), 84  
44 formatname () (annize.features.documentation.sphinx.output.common.O  
method), 48  
exists () (annize.fs.FsEntry method), 79  
formatname () (annize.features.documentation.sphinx.output.html.HtmlC  
method), 48  
formatname () (annize.features.documentation.sphinx.output.pdf.PdfOut  
method), 50  
formatname () (annize.features.documentation.sphinx.output.plaintext.P  
method), 50  
freedesktop\_name () (an-  
nize.features.distributables.debian.Category  
property), 29  
fresh\_temp\_dir () (in module annize.fs.temp), 83  
friendly\_filesize () (in module an-  
nize.features.homepage.sections.download),  
66  
friendly\_name\_suggestion () (an-  
nize.features.distributables.flatpak.Repository  
property), 43  
from\_files () (annize.project.ProjectDefinition static  
method), 88  
FsEndpoint (class in an-  
nize.features.files.transfer.common), 61  
FsEntry (class in annize.features.files.common), 64  
FsEntry (class in annize.fs), 79  
fsentry\_by\_path () (in module annize.fs), 82  
fullname () (annize.features.authors.Author prop-  
erty), 74  
G  
Gallery (class in annize.features.mediagalleries), 77  
Gallery.Item (class in an-  
nize.features.mediagalleries), 77  
GamesActionCategory (in module an-  
nize.features.distributables.debian), 30  
GamesAdventureCategory (in module an-  
nize.features.distributables.debian), 30  
GamesBlocksCategory (in module an-  
nize.features.distributables.debian), 30  
GamesBoardCategory (in module an-  
nize.features.distributables.debian), 31  
GamesCardCategory (in module an-  
nize.features.distributables.debian), 31  
GamesPuzzlesCategory (in module an-  
nize.features.distributables.debian), 31  
GamesSection (in module an-  
nize.features.distributables.debian), 31

annize.features.distributables.pythonwheel.Package.  
attribute), 45  
nize.features.distributables.flatpak), 40  
nize.features.distributables.flatpak), 42  
nize.features.distributables.debian), 30  
nize.features.distributables.debian), 30  
nize.features.distributables.pythonwheel),  
format () (annize.i18n.TrStr method), 84  
formatname () (annize.features.documentation.sphinx.output.common.O  
method), 48  
formatname () (annize.features.documentation.sphinx.output.html.HtmlC  
method), 48  
formatname () (annize.features.documentation.sphinx.output.pdf.PdfOut  
method), 50  
formatname () (annize.features.documentation.sphinx.output.plaintext.P  
method), 50  
freedesktop\_name () (an-  
nize.features.distributables.debian.Category  
property), 29  
fresh\_temp\_dir () (in module annize.fs.temp), 83  
friendly\_filesize () (in module an-  
nize.features.homepage.sections.download),  
66  
friendly\_name\_suggestion () (an-  
nize.features.distributables.flatpak.Repository  
property), 43  
from\_files () (annize.project.ProjectDefinition static  
method), 88  
FsEndpoint (class in an-  
nize.features.files.transfer.common), 61  
FsEntry (class in annize.features.files.common), 64  
FsEntry (class in annize.fs), 79  
fsentry\_by\_path () (in module annize.fs), 82  
fullname () (annize.features.authors.Author prop-  
erty), 74  
G  
Gallery (class in annize.features.mediagalleries), 77  
Gallery.Item (class in an-  
nize.features.mediagalleries), 77  
GamesActionCategory (in module an-  
nize.features.distributables.debian), 30  
GamesAdventureCategory (in module an-  
nize.features.distributables.debian), 30  
GamesBlocksCategory (in module an-  
nize.features.distributables.debian), 30  
GamesBoardCategory (in module an-  
nize.features.distributables.debian), 31  
GamesCardCategory (in module an-  
nize.features.distributables.debian), 31  
GamesPuzzlesCategory (in module an-  
nize.features.distributables.debian), 31  
GamesSection (in module an-  
nize.features.distributables.debian), 31



GamesSimulationCategory (in module *annize.features.distributables.debian*), 31

GamesStrategyCategory (in module *annize.features.distributables.debian*), 31

GamesToolsCategory (in module *annize.features.distributables.debian*), 31

GamesToysCategory (in module *annize.features.distributables.debian*), 31

generate() (*annize.features.documentation.common.Document* method), 59

generate() (*annize.features.documentation.sphinx.common.Document* method), 54

generate() (*annize.features.homepage.common.Homepage* method), 68

generate() (*annize.features.homepage.common.HomepageSection* method), 92

generate\_all\_languages() (*annize.features.documentation.common.Document* method), 59

generate\_all\_languages() (*annize.features.documentation.sphinx.common.Document* method), 54

generate\_master\_doc() (*annize.features.documentation.sphinx.common.ApiReferenceLanguage* method), 52

generate\_master\_doc() (*annize.features.documentation.sphinx.javascript.JavaScriptApiReferenceLanguage* method), 58

generate\_master\_doc() (*annize.features.documentation.sphinx.python.Python3ApiReferenceLanguage* method), 59

GeneratedDocument (class in *annize.features.documentation.common*), 59

GeneratedHomepage (class in *annize.features.homepage.common*), 67

GenerateMOs (class in *annize.features.i18n.gettext*), 71

get\_arguments\_infos() (in module *annize.callconv.typehints*), 15

get\_commit\_message() (*annize.features.versioncontrol.common.VersionControlSystem* method), 72

get\_commit\_message() (*annize.features.versioncontrol.git.VersionControlSystem* method), 73

get\_commit\_time() (*annize.features.versioncontrol.common.VersionControlSystem* method), 73

get\_commit\_time() (*annize.features.versioncontrol.git.VersionControlSystem* method), 73

get\_culture() (in module *annize.i18n*), 85

get\_current\_revision() (*annize.features.versioncontrol.common.VersionControlSystem* method), 73

get\_current\_revision() (*annize.features.versioncontrol.git.VersionControlSystem* method), 73

get\_description() (*annize.project.Node* method), 87

get\_files() (*annize.features.distributables.common.Group* method), 26

get\_files() (*annize.features.distributables.flatpak.Group* method), 43

get\_iso\_639\_1\_lang\_code() (*annize.i18n.Culture* static method), 83

get\_materialized() (*annize.project.materializer.core.ProjectMaterializer* method), 92

get\_materialized\_children() (*annize.project.materializer.core.NodeMaterialization* method), 91

get\_materialized\_children\_tuples() (*annize.project.materializer.core.NodeMaterialization* method), 91

get\_object\_metadata() (*annize.context.Context* method), 15

get\_package() (*annize.features.distributables.common.PackageStore* method), 26

get\_package\_history() (*annize.features.distributables.common.PackageStore* method), 26

get\_parameters\_from\_childnodes\_infos() (in module *annize.callconv*), 14

get\_parser() (in module *annize.annize\_cli*), 98

get\_revision\_list() (*annize.features.versioncontrol.common.VersionControlSystem* method), 73

get\_revision\_list() (*annize.features.versioncontrol.git.VersionControlSystem* method), 73

get\_revision\_number() (*annize.features.versioncontrol.common.VersionControlSystem* method), 73

get\_revision\_number() (*annize.features.versioncontrol.git.VersionControlSystem* method), 73

get\_typeinfo() (in module *annize.callconv.typehints*), 15

GettextTranslationProvider (class in *annize.i18n*), 83

GnomeSection (in module *annize.features.distributables.debian*), 31

GnuLinux (class in *annize.features.dependencies.common*), 24

GnuRSection (in module *nize.features.distributables.debian*), 31

GnustepSection (in module *nize.features.distributables.debian*), 31

GpgFile (class in *nize.features.distributables.flatpak*), 42

GPLv3 (in module *annize.features.licensing*), 76

GraphicsSection (in module *nize.features.distributables.debian*), 31

green() (*annize.data.color.Color* property), 18

Group (class in *annize.features.distributables.common*), 25

Group (class in *annize.features.distributables.flatpak*), 42

guess\_project\_root\_directory() (in module *annize.engine*), 98

## H

HamRadioSection (in module *nize.features.distributables.debian*), 31

has\_result() (*annize.project.materializer.core.NodeMaterialization* property), 91

has\_shell\_access() (*annize.features.files.transfer.ssh.Endpoint* property), 62

HaskellSection (in module *nize.features.distributables.debian*), 31

head() (*annize.features.homepage.common.HomepageSection* property), 69

heading() (*annize.features.documentation.sphinx.common.ApiReference* property), 52

heading() (*annize.features.documentation.sphinx.rst.RstGenerator* static method), 59

HelpCategory (in module *nize.features.distributables.debian*), 31

homepage (*annize.features.distributables.debian.Package* attribute), 35

homepage (*annize.features.distributables.pythonwheel.Package* attribute), 45

Homepage (class in *annize.features.homepage.common*), 67

homepage\_url() (*annize.features.base.Data* property), 74

homepage\_url() (in module *annize.features.base*), 75

HomepageSection (class in *annize.features.homepage.common*), 68

HomepageSection.\_GenerateInfo (class in *annize.features.homepage.common*), 68

HomepageSection.\_PrePostprocGenerateInfo (class in *annize.features.homepage.common*), 69

HomepageSection.Product (class in *annize.features.homepage.common*), 68

an-host() (*annize.features.files.transfer.ssh.Endpoint* property), 62

an-html\_color\_spec() (*annize.data.color.Color* property), 18

an-HtmlOutputGenerator (class in *annize.features.documentation.sphinx.output.html*), 48

an-HtmlOutputSpec (class in *annize.features.documentation.common*), 60

an-HtmlOutputSpec (class in *annize.features.documentation.sphinx.output.html*), 48

an-icon() (*annize.features.dependencies.common.Dependency* property), 23

an-icon() (*annize.features.distributables.debian.MenuEntry* property), 33

an-identity\_file() (*annize.features.files.transfer.ssh.Endpoint* property), 62

an-if\_passing(*annize.callconv.ParametersFromChildNodesInfos* attribute), 13

an-if\_typehint\_matches(*annize.callconv.ParametersFromChildNodesInfos* attribute), 13

an-importance() (*annize.features.dependencies.common.Dependency* property), 23

an-importance() (*annize.features.dependencies.common.Kind* property), 24

an-imprint() (*annize.features.base.Data* property), 74

an-in\_file() (in module *annize.features.base*), 75

an-in\_file() (*annize.project.parser.XmlParser.Context* method), 93

an-in\_node() (*annize.project.parser.XmlParser.Context* method), 93

an-Included (class in *annize.features.dependencies.common*), 24

an-IndependentTrStr (class in *annize.i18n*), 84

an-Inject (class in *annize.features.datainjections.common*), 22

an-inject() (*annize.features.datainjections.common.Injection* method), 22

an-inject() (*annize.features.datainjections.python.Injection* method), 23

an-Injection (class in *annize.features.datainjections.common*), 22

an-Injection (class in *annize.features.datainjections.python*), 23

an-inner() (*annize.callconv.typehints.ListTypeInfo* property), 14

an-input\_dialog() (*annize.userfeedback.NullUserFeedbackController* method), 94

input\_dialog() (annize.userfeedback.static.StaticUserFeedbackController method), 96

input\_dialog() (annize.userfeedback.tty.TtyUserFeedbackController method), 97

input\_dialog() (annize.userfeedback.UserFeedbackController method), 95

input\_dialog() (in module annize.userfeedback), 95

insert\_child() (annize.project.Node method), 87

InterpretersSection (in module annize.features.distributables.debian), 32

IntrospectionSection (in module annize.features.distributables.debian), 32

intstruct() (annize.features.documentation.sphinx.common.Document.GenerateInfo property), 54

is\_compatible\_for() (annize.features.documentation.sphinx.output.common.OutputGenerator class method), 48

is\_compatible\_for() (annize.features.documentation.sphinx.output.html.HtmlOutputGenerator class method), 48

is\_compatible\_for() (annize.features.documentation.sphinx.output.pdf.PdfOutputGenerator class method), 50

is\_compatible\_for() (annize.features.documentation.sphinx.output.plaintext.PlaintextOutputGenerator class method), 50

is\_constructable\_from\_string() (annize.callconv.typehints.TypeInfo property), 15

is\_dir() (annize.fs.FsEntry method), 79

is\_file() (annize.fs.FsEntry method), 79

is\_friendly\_name() (annize.context.Context method), 16

is\_friendly\_name() (in module annize.context), 17

is\_gui() (annize.features.distributables.debian.MenuEntry property), 33

is\_gui() (annize.features.distributables.pythonwheel.ExecutableLink property), 44

is\_homepage() (annize.features.documentation.common.HtmlOutputSpec property), 60

is\_link() (annize.fs.FsEntry method), 79

is\_ownable() (annize.fs.FsEntry property), 80

is\_owned() (annize.fs.FsEntry property), 80

is\_toplevel\_object() (annize.context.Context method), 16

is\_toplevel\_object() (in module annize.context), 17

iso\_639\_1\_lang\_code() (annize.i18n.Culture property), 83

Item (class in annize.features.changelog.common), 22

items() (annize.features.changelog.common.Entry property), 22

items() (annize.features.mediagalleries.Gallery property), 77

**J**

JavaScriptApiReferenceLanguage (class in annize.features.documentation.sphinx.javascript), 58

JavascriptSection (in module annize.features.distributables.debian), 32

JavaSection (in module annize.features.distributables.debian), 32

K

KdeSection (in module annize.features.distributables.debian), 32

kernelSection (in module annize.features.distributables.debian), 32

Keyword (class in annize.features.base), 75

Keywords (annize.features.distributables.pythonwheel.Package.\_BuildInfo attribute), 45

Keywords (class in annize.features.base), 75

Keywords() (annize.features.base.Keywords property), 75

Kind (class in annize.features.dependencies.common), 24

kind() (annize.features.dependencies.common.Dependency property), 24

kitversion (annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo attribute), 41

**L**

label() (annize.features.dependencies.common.Dependency property), 24

label() (annize.features.dependencies.common.Kind property), 24

language() (annize.features.documentation.sphinx.common.Document.C property), 54

language() (annize.features.documentation.sphinx.common.RstDocument property), 56

LanguagePacksSection (in module annize.features.distributables.debian), 32

languages() (annize.features.documentation.common.\_DocumentGenerator property), 60

languages() (annize.features.homepage.common.Homepage property), 68

LGPLv3 (in module annize.features.licensing), 76

LibrariesSection (in module annize.features.distributables.debian), 32

LibraryDevelopmentSection (in module annize.features.distributables.debian), 32

license (*annize.features.distributables.pythonwheel.Package.\_BuildInfo* attribute), 45  
License (class in *annize.features.licensing*), 76  
licensename (*annize.features.distributables.debian.Package.\_BuildInfo* attribute), 35  
Line (class in *annize.features.version*), 78  
linkname () (*annize.features.distributables.pythonwheel.ExecutableLink* property), 44  
LispSection (in module *annize.features.distributables.debian*), 32  
ListTypeInfo (class in *annize.callconv.typehints*), 14  
load\_feature () (*annize.project.featureloader.DefaultFeatureLoader* method), 93  
load\_feature () (*annize.project.featureloader.FeatureLoader* method), 93  
load\_project () (in module *annize.engine*), 98  
load\_project\_definition () (in module *annize.engine*), 98  
LoadedProject (class in *annize.engine*), 98  
LocalRepository (class in *annize.features.distributables.flatpak*), 43  
locationstring () (*annize.features.files.transfer.ssh.Endpoint* method), 62  
logo\_image () (*annize.features.documentation.sphinx.output.html.HtmlOutputSpec* property), 49  
long\_description (in *annize.features.distributables.pythonwheel.Package.\_BuildInfo* attribute), 46  
long\_description () (*annize.features.base.Data* property), 74  
long\_description () (in module *annize.features.base*), 75  
long\_str () (*annize.data.unique.UniqueId* property), 19

## M

MachineRootDirectory (class in *annize.features.files.common*), 64  
MailSection (in module *annize.features.distributables.debian*), 32  
main () (in module *annize.annize\_cli*), 98  
mark\_object\_as\_toplevel () (*annize.context.Context* method), 16  
masterlink () (*annize.features.documentation.sphinx.output.html.HtmlOutputSpec* property), 49  
materialize () (in module *annize.project.materializer*), 89  
MaterializerError, 86  
MathematicsSection (in module *annize.features.distributables.debian*), 32  
MenuEntries (class in *annize.features.distributables.debian.Package.\_BuildInfo* attribute), 35  
MenuEntry (class in *annize.features.distributables.debian*), 32  
MenuEntry (class in *annize.features.distributables.flatpak*), 43  
message\_dialog () (*annize.userfeedback.NullUserFeedbackController* method), 94  
message\_dialog () (*annize.userfeedback.static.StaticUserFeedbackController* method), 96  
message\_dialog () (*annize.userfeedback.tty.TtyUserFeedbackController* method), 97  
message\_dialog () (*annize.userfeedback.UserFeedbackController* method), 95  
message\_dialog () (in module *annize.userfeedback*), 96  
MetaPackagesSection (in module *annize.features.distributables.debian*), 33  
methodname () (*annize.features.distributables.pythonwheel.ExecutableLink* property), 44  
MiscellaneousSection (in module *annize.features.distributables.debian*), 33  
MissingOwnershipError, 81  
MIT (in module *annize.features.licensing*), 77  
mkdir () (*annize.fs.FsEntry* method), 80  
modified\_time () (*annize.fs.FsEntry* method), 80  
module  
annize.annize\_cli, 97  
annize.aux, 13  
annize.aux.\_project\_infos, 13  
annize.callconv, 13  
annize.callconv.typehints, 14  
annize.context, 15  
annize.data, 18  
annize.data.color, 18  
annize.data.container, 19  
annize.data.unique, 19  
annize.data.version, 19  
annize.engine, 98  
annize.features.authors, 74  
annize.features.base, 74  
annize.features.changelog.common, 21  
annize.features.datainjections.common, 22  
annize.features.datainjections.python,



23  
 annize.features.dependencies.common,  
 23  
 annize.features.dependencies.python,  
 24  
 annize.features.distributables.common,  
 25  
 annize.features.distributables.debian,  
 26  
 annize.features.distributables.flatpak,  
 39  
 annize.features.distributables.pythonwheel,  
 44  
 annize.features.distributables.store.py,  
 25  
 annize.features.distributables.tar,  
 47  
 annize.features.documentation.common,  
 59  
 annize.features.documentation.sphinx.\_util,  
 50  
 annize.features.documentation.sphinx.common,  
 51  
 annize.features.documentation.sphinx.cpp,  
 56  
 annize.features.documentation.sphinx.doxy,  
 57  
 annize.features.documentation.sphinx.java,  
 58  
 annize.features.documentation.sphinx.out,  
 48  
 annize.features.documentation.sphinx.out,  
 48  
 annize.features.documentation.sphinx.out,  
 50  
 annize.features.documentation.sphinx.out,  
 50  
 annize.features.documentation.sphinx.py,  
 58  
 annize.features.documentation.sphinx.rst,  
 59  
 annize.features.files.common, 62  
 annize.features.files.transfer.common,  
 61  
 annize.features.files.transfer.ssh,  
 62  
 annize.features.homepage.common, 67  
 annize.features.homepage.sections.about,  
 65  
 annize.features.homepage.sections.changelog,  
 65  
 annize.features.homepage.sections.documentation,  
 65  
 annize.features.homepage.sections.download,  
 66  
 annize.features.homepage.sections.gallery,  
 66  
 annize.features.homepage.sections.imprint,  
 67  
 annize.features.homepage.sections.license,  
 67  
 annize.features.il8n.common, 70  
 annize.features.il8n.gettext, 71  
 annize.features.licensing, 76  
 annize.features.mediagalleries, 77  
 annize.features.task, 78  
 annize.features.testing.common, 71  
 annize.features.testing.pylint, 72  
 annize.features.testing.pytest, 72  
 annize.features.testing.pyunit, 72  
 annize.features.version, 78  
 annize.features.versioncontrol.common,  
 72  
 annize.features.versioncontrol.git,  
 73  
 annize.fs, 78  
 annize.fs.temp, 83  
 annize.il8n, 83  
 annize.project, 86  
 annize.project.featureloader, 93  
 annize.project.materializer, 89  
 annize.project.materializer.behaviors,  
 89  
 annize.project.materializer.behaviors.argument,  
 89  
 annize.project.materializer.behaviors.basket,  
 90  
 annize.project.materializer.behaviors.block,  
 90  
 annize.project.materializer.behaviors.featureun,  
 90  
 annize.project.materializer.behaviors.reference,  
 90  
 annize.project.materializer.core, 91  
 annize.project.materializer.preprocessors,  
 92  
 annize.project.parser, 93  
 annize.userfeedback, 94  
 annize.userfeedback.static, 96  
 annize.userfeedback.tty, 96  
 module`name()` (*annize.features.distributables.pythonwheel.ExecutableL*  
*property*), 44  
 MonoCliSection (in module *an-*  
*nize.features.distributables.debian*), 33  
 Mount (class in *annize.fs*), 81  
`mount()` (*annize.fs.FsEntry* method), 80  
 MPLv11 (in module *annize.features.licensing*), 77  
 MPLv21 (in module *annize.features.licensing*), 77

multilanguage\_frame() (an- nize.features.documentation.sphinx.output.common.OutputGenerator.project.materializer.behaviors.reference.SkipUnresolvableRe-  
 method), 48 method), 91  
 multilanguage\_frame() (an- NodeMaterialization (class in an- nize.features.documentation.sphinx.output.html.HtmlOutputGenerator.project.materializer.core), 91  
 method), 48 nodes() (annize.project.ProjectDefinition property), 88  
 multiple(annize.callconv.ParametersFromChildNodesInfo, formalize\_blockscopes() (in module an-  
 attribute), 13 nize.project.materializer.preprocessors), 92  
 MultipleValuesForSingleArgumentError, NullUserFeedbackController (class in an-  
 13 nize.userfeedback), 94  
 NumericVersionPatternSegment (class in an-  
 nize.data.version), 20  
**N**  
 name(annize.features.distributables.debian.Package.\_BuildInfo  
 attribute), 35 **O**  
 name(annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo\_by\_name() (annize.context.Context method),  
 attribute), 41 16  
 name(annize.features.distributables.pythonwheel.Package.\_BuildInfo\_by\_name() (in module annize.context), 17  
 attribute), 46 object\_name() (annize.context.Context method), 16  
 name() (annize.callconv.typehints.TypeInfo property), object\_name() (in module annize.context), 17  
 15 object\_names() (annize.context.Context method), 16  
 name() (annize.features.distributables.debian.ExecutableLink, object\_names() (in module annize.context), 17  
 property), 30 objects\_by\_type() (annize.context.Context  
 name() (annize.features.distributables.debian.MenuEntry method), 16  
 property), 33 objects\_by\_type() (in module annize.context), 17  
 name() (annize.features.distributables.debian.Section OcamlSection (in module an-  
 property), 38 nize.features.distributables.debian), 33  
 name() (annize.features.licensing.License property), 77 OldLibrariesSection (in module an-  
 name() (annize.project.ArgumentNode property), 86 nize.features.distributables.debian), 33  
 NetworkSection (in module an- on\_unresolvable() (annize.project.ReferenceNode  
 nize.features.distributables.debian), 33 property), 88  
 NewsgroupsSection (in module an- OnFeatureUnavailableNode (class in an-  
 nize.features.distributables.debian), 33 nize.project), 87  
 NoCurrentCultureError, 84 OnFeatureUnavailableNode.Action (class in  
 Node (class in annize.project), 86 annize.project), 87  
 node() (annize.project.materializer.core.NodeMaterialization, OptionalVersionPatternSegment (class in an-  
 property), 91 nize.data.version), 20  
 node\_context() (an- OriginalOwnershipImpossibleError, 81  
 nize.project.materializer.behaviors.argument.ArgumentBehavior, andFSsSection (in module an-  
 method), 90 nize.features.distributables.debian), 33  
 node\_context() (an- outdir() (annize.features.documentation.sphinx.common.Document.Gen-  
 nize.project.materializer.behaviors.basket.BasketBehavior property), 54  
 method), 90 OutOfContextError, 17  
 node\_context() (an- OutputGenerator (class in an-  
 nize.project.materializer.behaviors.Behavior nize.features.documentation.sphinx.output.common),  
 method), 89 48  
 node\_context() (an- OutputSpec (class in an-  
 nize.project.materializer.behaviors.block.BlockBehavior nize.features.documentation.common), 60  
 method), 90 outputspec() (annize.features.documentation.sphinx.output.common.O  
 node\_context() (an- property), 48  
 nize.project.materializer.behaviors.featureunavailable.FeatureUnavailableBehavior, 80  
 method), 90 **P**  
 node\_context() (an- Package (class in an-  
 nize.project.materializer.behaviors.reference.ReferenceBehavior nize.features.distributables.debian), 33  
 method), 90

Package (class in annize.features.distributables.pythonwheel), 44  
 Package (class in annize.features.distributables.tar), 47  
 Package.\_BuildInfo (class in annize.features.distributables.debian), 34  
 Package.\_BuildInfo (class in annize.features.distributables.pythonwheel), 45  
 PackageStore (class in annize.features.distributables.common), 26  
 packagestore() (annize.features.distributables.common.Group property), 26  
 parameter\_from\_childnodes() (in module annize.callconv), 14  
 parameter\_name (annize.callconv.ParametersFromChildNodesInfos attribute), 13  
 ParametersFromChildNodesInfos (class in annize.callconv), 13  
 parent() (annize.fs.FsEntry method), 80  
 parent() (annize.project.Node property), 87  
 parse() (in module annize.project.parser), 94  
 parse\_file() (annize.project.parser.XmlParser method), 94  
 ParserError, 87  
 partname() (annize.data.version.NumericVersionPatternSegment property), 20  
 path() (annize.features.distributables.debian.ExecutableLink property), 30  
 path() (annize.features.files.common.DirPart property), 63  
 path() (annize.features.versioncontrol.git.VersionControlSystem property), 73  
 path() (annize.fs.FsEntry method), 80  
 path() (annize.project.FileNode property), 86  
 pattern() (annize.data.version.Version property), 21  
 PdfOutputGenerator (class in annize.features.documentation.sphinx.output.pdf), 50  
 PdfOutputSpec (class in annize.features.documentation.common), 60  
 PerlSection (in module annize.features.distributables.debian), 37  
 PhpSection (in module annize.features.distributables.debian), 37  
 PinoHomepageAuxInfo (class in annize.features.homepage.common), 70  
 pkgpath\_debian (annize.features.distributables.debian.Package.\_BuildInfo attribute), 35  
 pkgpath\_documentation (annize.features.distributables.debian.Package.\_BuildInfo attribute), 35  
 pkgpath\_pixmaps (annize.features.distributables.debian.Package.\_BuildInfo attribute), 35  
 pkgpath\_setuppy (annize.features.distributables.pythonwheel.Package.\_BuildInfo attribute), 46  
 pkgpath\_share (annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo attribute), 41  
 pkgpath\_share\_applications (annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo attribute), 41  
 pkgpath\_share\_icons (annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo attribute), 41  
 pkgpath\_usrbin (annize.features.distributables.debian.Package.\_BuildInfo attribute), 36  
 pkgrootpath (annize.features.distributables.debian.Package.\_BuildInfo attribute), 36  
 pkgrootpath (annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo attribute), 41  
 pkgrootpath (annize.features.distributables.pythonwheel.Package.\_BuildInfo attribute), 46  
 pkgsize (annize.features.distributables.debian.Package.\_BuildInfo attribute), 36  
 PlainTextOutputGenerator (class in annize.features.documentation.sphinx.output.plaintext), 50  
 PlaintextOutputSpec (class in annize.features.documentation.common), 60  
 platform (annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo attribute), 41  
 port() (annize.features.files.transfer.ssh.Endpoint property), 62  
 postinst (annize.features.distributables.debian.Package.\_BuildInfo attribute), 36  
 postproc() (annize.features.documentation.sphinx.output.common.Output method), 48  
 postproc() (annize.features.documentation.sphinx.output.pdf.PdfOutput method), 50  
 postproc\_generate() (annize.features.homepage.common.HomepageSection method), 69  
 postproc\_generate() (annize.features.homepage.sections.download.Section method), 66  
 prepare\_generate() (annize.features.documentation.sphinx.common.ApiReferenceLanguage method), 52  
 prepare\_generate() (annize.features.documentation.sphinx.doxygencompat.DoxygenSupport method), 58

prepare\_generate() (an- ProjectDefinedTranslationProvider (class  
     nize.features.documentation.sphinx.javascript.JavaScriptApiReferenceLanguage (class in annize.features.common), 70  
     method), 58      ProjectDefinition (class in annize.project), 88  
 prepare\_generate() (an- ProjectDir (class in annize.features.files.common),  
     nize.features.documentation.sphinx.output.common.OutputGenerator  
     method), 48      ProjectMaterializer (class in an-  
 prepare\_generate() (an- nize.project.materializer.core), 91  
     nize.features.documentation.sphinx.output.html.HtmlOutputGenerator (class in annize.i18n), 84  
     method), 48      public\_url() (annize.features.distributables.flatpak.Repository  
 prepare\_generate() (an- property), 44  
     nize.features.documentation.sphinx.python.Python3ApiReferenceLanguage (class in module annize.features.licensing),  
     method), 59      77  
 preproc\_generate() (an- Python (class in annize.features.dependencies.python),  
     nize.features.homepage.common.HomepageSection 24  
     method), 69      Python3ApiReferenceLanguage (class in an-  
 preproc\_generate() (an- nize.features.documentation.sphinx.python), 58  
     nize.features.homepage.common.PinoHomepageAbstractSection (in module an-  
     method), 70      nize.features.distributables.debian), 37  
 preproc\_generate() (an-  
     nize.features.homepage.sections.documentation.Section  
     method), 65      read\_file() (annize.fs.FsEntry method), 80  
 preproc\_generate() (an- readable\_path() (annize.fs.\_SimpleFsEntry  
     nize.features.homepage.sections.download.Section method), 81  
     method), 66      readable\_path() (annize.fs.AdHocFsEntry  
 preproc\_generate() (an- method), 78  
     nize.features.homepage.sections.gallery.Section readable\_path() (annize.fs.FsEntry method), 80  
     method), 66      readable\_path() (annize.fs.temp.\_TempDir  
 prerm(annize.features.distributables.debian.Package.\_BuildInfo method), 83  
     attribute), 36      ReadmeDocument (class in an-  
 pretty\_project\_name() (an- nize.features.documentation.sphinx.common),  
     nize.features.base.Data property), 74      54  
 pretty\_project\_name() (in module an- Recommended (class in an-  
     nize.features.base), 75      nize.features.dependencies.common), 24  
 processonly\_long\_str() (an- red() (annize.data.color.Color property), 18  
     nize.data.unique.UniqueId property), 19      reference\_key() (annize.project.ReferenceNode  
 processonly\_short\_str() (an- property), 88  
     nize.data.unique.UniqueId property), 19      ReferenceBehavior (class in an-  
 PROJECT (annize.project.ScopableBlockNode.Scope at- nize.project.materializer.behaviors.reference),  
     tribute), 88      90  
 project\_default (annize.annize\_cli.Cli attribute),  
     98      ReferenceNode (class in annize.project), 88  
 project\_directory() (annize.features.base.Data      ReferenceNode.OnUnresolvableAction (class  
     property), 75      in annize.project), 88  
 project\_directory() (in module an-      regexp\_string() (an-  
     nize.features.base), 75      nize.data.version.AbstractVersionPatternSegment  
 project\_keywords() (in module an-      method), 19  
     nize.features.base), 76      regexp\_string() (an-  
 project\_licenses() (in module an-      nize.data.version.ConcatenatedVersionPatternSegment  
     nize.features.licensing), 77      method), 20  
 project\_name() (annize.features.base.Data prop-      regexp\_string() (an-  
     erty), 75      nize.data.version.NumericVersionPatternSegment  
 project\_name() (in module annize.features.base), 76      method), 20  
 project\_versions() (in module an-      regexp\_string() (an-  
     nize.features.version), 78      nize.data.version.OptionalVersionPatternSegment  
     method), 20

regexp\_string() (annize.data.version.SeparatorVersionPatternSegment method), 21  
 register\_output\_generator() (in module annize.features.documentation.sphinx.output.common), 48  
 relative\_path() (annize.features.files.common.FsEntry property), 64  
 remove() (annize.fs.FsEntry method), 80  
 remove\_child() (annize.project.Node method), 87  
 Repository (class in annize.features.distributables.flatpak), 43  
 Required (class in annize.features.dependencies.common), 24  
 resolve\_appendtonodes() (in module annize.project.materializer.preprocessors), 92  
 resolved\_type() (annize.callconv.typehints.TypeInfo property), 15  
 resolver\_node\_context() (annize.project.materializer.behaviors.reference.ReferenceBehavior method), 90  
 restrict\_path() (annize.fs.temp.\_TempDir method), 83  
 result (annize.features.distributables.debian.Package.\_BuildInfo attribute), 36  
 result (annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo attribute), 41  
 result (annize.features.distributables.pythonwheel.Package.\_BuildInfo attribute), 46  
 result() (annize.project.materializer.core.NodeMaterialization property), 91  
 resultpath() (annize.features.documentation.sphinx.output.pdf.PdfOutputGenerator method), 50  
 retry\_node\_context() (annize.project.materializer.behaviors.reference.ReferenceBehavior method), 90  
 root() (annize.features.files.common.DirPart property), 63  
 root() (annize.features.files.common.FsEntry property), 64  
 rst\_text() (annize.features.homepage.common.HomepageSection.Product property), 68  
 RstDocument (class in annize.features.documentation.sphinx.common), 55  
 RstDocumentVariant (class in annize.features.documentation.sphinx.common), 55  
 RstGenerator (class in annize.features.documentation.sphinx.rst), 59  
 RubySection (in module annize.features.distributables.debian), 37  
 run() (annize.features.testing.common.Test method), 71  
 run() (annize.features.testing.common.TestGroup method), 71  
 run() (annize.features.testing.pylint.Test method), 72  
 run() (annize.features.testing.pytest.Test method), 72  
 run() (annize.features.testing.pyunit.Test method), 72  
 RunTests (class in annize.features.testing.common), 71  
 RustSection (in module annize.features.distributables.debian), 38

## S

ScalarValueNode (class in annize.project), 88  
 scalehue() (annize.data.color.Color method), 18  
 ScienceSection (in module annize.features.distributables.debian), 38  
 ScopableBlockNode (class in annize.project), 88  
 ScopableBlockNode.Scope (class in annize.project), 88  
 scope() (annize.project.ScopableBlockNode property), 89  
 ScreenLockingCategory (in module annize.features.distributables.debian), 38  
 ScreenSavingCategory (in module annize.features.distributables.debian), 38  
 sdk (annize.features.distributables.flatpak.FlatpakImage.\_BuildInfo attribute), 41  
 section (annize.features.distributables.debian.Package.\_BuildInfo attribute), 36  
 Section (class in annize.features.distributables.debian), 38  
 Section (class in annize.features.homepage.sections.about), 65  
 Section (class in annize.features.homepage.sections.changelog), 65  
 Section (class in annize.features.homepage.sections.documentation), 65  
 Section (class in annize.features.homepage.sections.download), 66  
 Section (class in annize.features.homepage.sections.gallery), 66  
 Section (class in annize.features.homepage.sections.imprint), 67  
 Section (class in annize.features.homepage.sections.license), 67  
 sections() (annize.features.homepage.common.Homepage property), 68  
 segment\_names() (annize.data.version.AbstractVersionPatternSegment method), 71



[property](#)), 19  
[segment\\_names\(\)](#) ([annize.data.version.ConcatenatedVersionPatternSegment](#) [property](#)), 20  
[segment\\_names\(\)](#) ([annize.data.version.NumericVersionPatternSegment](#) [property](#)), 20  
[segment\\_names\(\)](#) ([annize.data.version.OptionalVersionPatternSegment](#) [property](#)), 20  
[segment\\_names\(\)](#) ([annize.data.version.VersionPattern](#) [property](#)), 21  
[segments\(\)](#) ([annize.data.version.VersionPattern](#) [property](#)), 21  
[segments\\_tuples\(\)](#) ([annize.data.version.Version](#) [property](#)), 21  
[segments\\_tuples\(\)](#) ([annize.features.versioncontrol.common.BuildVersion](#) [property](#)), 72  
[segments\\_tuples\\_to\\_text\(\)](#) ([annize.data.version.AbstractVersionPatternSegment](#) [method](#)), 19  
[segments\\_tuples\\_to\\_text\(\)](#) ([annize.data.version.ConcatenatedVersionPatternSegment](#) [method](#)), 20  
[segments\\_tuples\\_to\\_text\(\)](#) ([annize.data.version.NumericVersionPatternSegment](#) [method](#)), 20  
[segments\\_tuples\\_to\\_text\(\)](#) ([annize.data.version.OptionalVersionPatternSegment](#) [method](#)), 20  
[segments\\_tuples\\_to\\_text\(\)](#) ([annize.data.version.SeparatorVersionPatternSegment](#) [method](#)), 21  
[segments\\_tuples\\_to\\_text\(\)](#) ([annize.data.version.VersionPattern](#) [method](#)), 21  
[segments\\_values\(\)](#) ([annize.data.version.Version](#) [property](#)), 21  
[SeparatorVersionPatternSegment](#) (class in [annize.data.version](#)), 20  
[serialize\\_for\\_confpy\(\)](#) (in module [annize.features.documentation.sphinx.\\_utils](#)), 50  
[ServiceDescription](#) (class in [annize.features.distributables.debian](#)), 38  
[services](#) ([annize.features.distributables.debian.Package](#) [attribute](#)), 36  
[set\\_materialized\\_result\(\)](#) ([annize.project.materializer.core.NodeMaterialization](#) [method](#)), 91  
[set\\_object\\_metadata\(\)](#) ([annize.context.Context](#) [method](#)), 16  
[set\\_object\\_metadata\(\)](#) (in module [annize.context](#)), 18  
[set\\_object\\_name\(\)](#) ([annize.context.Context](#) [method](#)), 16  
[set\\_object\\_name\(\)](#) (in module [annize.context](#)), 18  
[setupy\\_conf\(\)](#) ([annize.features.distributables.pythonwheel.Package](#) [attribute](#)), 46  
[Share](#) (class in [annize.features.distributables.flatpak](#)), 44  
[shares](#) ([annize.features.distributables.flatpak.FlatpakImage](#) [attribute](#)), 41  
[ShellsSection](#) (in module [annize.features.distributables.debian](#)), 38  
[short\\_desc\(\)](#) ([annize.features.documentation.sphinx.output.html.HtmlOutputSpec](#) [property](#)), 49  
[short\\_desc\(\)](#) ([annize.features.homepage.common.Homepage](#) [property](#)), 68  
[short\\_str\(\)](#) ([annize.data.unique.UniqueId](#) [property](#)), 19  
[short\\_title\(\)](#) ([annize.features.documentation.sphinx.output.html.HtmlOutputSpec](#) [property](#)), 49  
[SimpleProjectHomepage](#) (class in [annize.features.homepage.common](#)), 70  
[SKIP](#) ([annize.project.ReferenceNode.OnUnresolvableAction](#) [attribute](#)), 88  
[SKIP\\_BLOCK](#) ([annize.project.OnFeatureUnavailableNode.Action](#) [attribute](#)), 87  
[SKIP\\_NODE](#) ([annize.project.OnFeatureUnavailableNode.Action](#) [attribute](#)), 87  
[SkipUnresolvableReferenceBehavior](#) (class in [annize.project.materializer.behaviors.reference](#)), 90  
[sockets](#) ([annize.features.distributables.flatpak.FlatpakImage](#) [attribute](#)), 41  
[sortindex\(\)](#) ([annize.features.homepage.common.HomepageSection](#) [property](#)), 70  
[SoundSection](#) (in module [annize.features.distributables.debian](#)), 38  
[source](#) ([annize.features.distributables.debian.Package](#) [attribute](#)), 36  
[source](#) ([annize.features.distributables.flatpak.FlatpakImage](#) [attribute](#)), 41  
[source](#) ([annize.features.distributables.pythonwheel.Package](#) [attribute](#)), 46  
[source\(\)](#) ([annize.features.documentation.sphinx.common.ApiReferenceList](#) [property](#)), 52  
[source\(\)](#) ([annize.features.documentation.sphinx.common.RstDocumentView](#) [property](#)), 56  
[StaticUserFeedbackController](#) (class in [annize.userfeedback.static](#)), 96  
[stop\\_materialization\\_here\\_and\\_try\\_again\(\)](#) ([annize.project.materializer.behaviors.Behavior](#) [method](#)), 96

method), 89  
store\_package() (annize.features.distributables.common.PackageStore method), 26  
String (class in annize.features.i18n.common), 71  
summary (annize.features.distributables.debian.Package.\_BuildInfo attribute), 36  
summary() (annize.features.base.Data property), 75  
summary() (in module annize.features.base), 76

## T

Task (class in annize.features.task), 78  
TasksSection (in module annize.features.distributables.debian), 38  
temp\_clone() (annize.fs.FsEntry method), 80  
TempFileByContent (class in annize.fs.temp), 83  
Test (class in annize.features.testing.common), 71  
Test (class in annize.features.testing.pylint), 72  
Test (class in annize.features.testing.pytest), 72  
Test (class in annize.features.testing.pyunit), 72  
TestGroup (class in annize.features.testing.common), 71  
TexSection (in module annize.features.distributables.debian), 38  
text() (annize.data.version.Version property), 21  
text() (annize.features.changelog.common.Item property), 22  
text() (annize.features.licensing.License property), 77  
text() (annize.features.versioncontrol.common.BuildVersion property), 72  
text\_to\_segments\_tuples() (annize.data.version.VersionPattern method), 21  
TextProcessingSection (in module annize.features.distributables.debian), 38  
theme() (annize.features.documentation.sphinx.output.html.HtmlOutputSpec property), 49  
time() (annize.features.changelog.common.Entry property), 22  
title() (annize.features.distributables.common.Group property), 26  
title() (annize.features.distributables.debian.MenuEntry property), 33  
title() (annize.features.documentation.common.HtmlOutputSpec property), 60  
title() (annize.features.documentation.sphinx.common.ReadmeDoc property), 55  
title() (annize.features.homepage.common.Homepage property), 68  
title() (annize.features.mediagalleries.Gallery property), 77  
TODO (class in annize.i18n), 84  
token() (annize.features.distributables.store.pypi.Connection property), 25  
tr() (annize.i18n.TrStr static method), 84  
tr() (in module annize.i18n), 85  
tr\_if\_trstr() (in module annize.i18n), 85  
transfer\_action\_copy() (annize.fs.\_TransferHelper method), 82  
transfer\_action\_move() (annize.fs.\_TransferHelper method), 82  
transfer\_to() (annize.fs.\_TransferHelper method), 82  
transformed() (annize.data.color.Color method), 18  
translate() (annize.features.i18n.common.ProjectDefinedTranslationProvider method), 71  
translate() (annize.i18n.GettextTranslationProvider method), 83  
translate() (annize.i18n.IndependentTrStr method), 84  
translate() (annize.i18n.ProvidedTrStr method), 84  
translate() (annize.i18n.TODO method), 84  
translate() (annize.i18n.TranslationProvider method), 84  
translate() (annize.i18n.TrStr method), 84  
translation\_providers() (in module annize.i18n), 85  
TranslationProvider (class in annize.i18n), 84  
TranslationUnavailableError, 85  
TrStr (class in annize.i18n), 84  
TrStrOrStr (in module annize.i18n), 84  
try\_get\_materialization\_for\_node() (annize.project.materializer.core.NodeMaterialization method), 91  
TryAgain, 92  
TtyUserFeedbackController (class in annize.userfeedback.tty), 96  
TypeInfo (class in annize.callconv.typehints), 14

## U

UniqueId (class in annize.data.unique), 19  
UnresolvableReferenceError, 89  
UnsatisfiableUserFeedbackAttemptError, 94  
UpdatePOs (class in annize.features.i18n.gettext), 71  
Upload (class in annize.features.distributables.store.pypi), 25  
UploadSpec (class in annize.features.files.transfer.common), 61  
update\_repo() (annize.features.distributables.flatpak.LocalRepository method), 43  
upload() (annize.features.distributables.flatpak.Repository method), 44  
UserFeedbackController (class in annize.userfeedback), 94  
username() (annize.features.files.transfer.ssh.Endpoint property), 62

UtilitiesSection (in module *annize.features.distributables.debian*), 38

## V

value() (*annize.project.ScalarValueNode* property), 88  
 version(*annize.features.distributables.debian.Package.\_BuildInfo* attribute), 36  
 version(*annize.features.distributables.pythonwheel.Package.\_BuildInfo* attribute), 46  
 Version (class in *annize.data.version*), 21  
 Version (class in *annize.features.version*), 78  
 version() (*annize.features.changelog.common.Entry* property), 22  
 version() (*annize.features.version.Line* property), 78  
 VersionControlSystem (class in *annize.features.versioncontrol.common*), 72  
 VersionControlSystem (class in *annize.features.versioncontrol.git*), 73  
 VersionControlSystemsSection (in module *annize.features.distributables.debian*), 38  
 VersionPattern (class in *annize.data.version*), 21  
 VideoSection (in module *annize.features.distributables.debian*), 39  
 VirtualPackagesSection (in module *annize.features.distributables.debian*), 39

## W

WebServersSection (in module *annize.features.distributables.debian*), 39  
 WebSoftwareSection (in module *annize.features.distributables.debian*), 39  
 write\_file() (*annize.fs.FsEntry* method), 80

## X

XfceSection (in module *annize.features.distributables.debian*), 39  
 XWindowSystemSoftwareSection (in module *annize.features.distributables.debian*), 39

## Z

ZopePloneFrameworkSection (in module *annize.features.distributables.debian*), 39